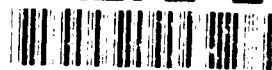


MASSACHUSETTS INSTITUTE OF TECHNOLOGY

# LINCOLN LABORATORY

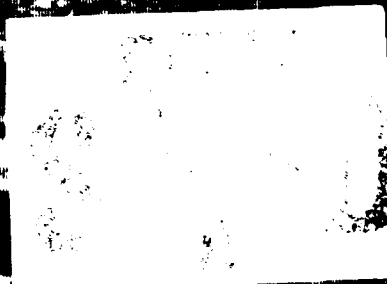
AD-A272 273



*Journal*

Spring 1958 Volume 6, Number 1

\*Original contains color  
plates. All LLL reproductions  
will be in black and  
white.\*



## On the Cover

■ A SYNTHETIC-APERTURE RADAR (SAR) image of a golf course in Stockbridge, New York was collected by the Lincoln Laboratory K<sub>a</sub>-band airborne radar, which transmits and receives horizontally and vertically polarized signals to produce three unique polarimetric images (HH, HV, and VV) of a scene. The polarimetric images are combined by using the polarimetric whitening filter, which reduces the speckle inherent in high-frequency SAR imagery. Two-by-two noncoherent averaging of the data reduces the speckle further. The combination of polarimetric whitening and noncoherent averaging produces a SAR image with near-optical quality (see the optical photograph below for comparison). Unlike optical sensors, however, the SAR can produce high-quality imagery day or night, under all conditions, including through dense clouds or smoke.

The SAR data were collected from a range of 7 km at a depression angle of 20°. Range in the SAR image increases from top to bottom; therefore, the radar shadows of trees and other objects extend toward the bottom of the image. Some features visible in this image are a street, a parking lot, the golf-course club house, a putting green adjacent to a water hazard, and a fairway to the right of the water hazard.

A large database of tactical target imagery and Stockbridge clutter imagery has been used to develop and test automatic target recognition algorithms. The article entitled "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System," by Leslie M. Novak, Gregory J. Owirka, and Christine M. Netishen, presents an overview of an end-to-end automatic target recognition system and a summary of the performance results of the recognition algorithms.



## EDITORIAL BOARD

Roger W. Sudbury, CHAIR  
Peter E. Blankenship  
Charles F. Bruce  
Vincent W.S. Chan  
John C. Fielding  
Herbert Kleiman  
John A. McCook  
Ivars Melngailis  
Antonio F. Pensa  
John A. Tabaczynski  
Lee O. Upton

## EDITORIAL STAFF

Jack Nolan, EDITOR-IN-CHIEF  
Dan E. Dudgeon, GUEST EDITOR  
Richard T. Lacoss, GUEST EDITOR  
Alden M. Hayashi, TECHNICAL EDITOR  
Randall Warniers, TECHNICAL EDITOR  
Leslie Safford Spiro, COPY EDITOR  
Patricia L. MacDonald, SUBSCRIPTION COORDINATOR

## PRODUCTION STAFF

Claude A. French, Jr., PUBLICATIONS MANAGER  
Patricia Kennedy Graham, ASSOCIATE PUBLICATIONS MANAGER  
Jonathan C. Barron, PHOTOGRAPHER  
Mary E. Bourquin, DOCUMENT COORDINATOR  
Kathleen L. Coy, LEAD TECHNICAL ARTIST  
Richard B. Doubleday, TECHNICAL ARTIST  
Paula M. Gentile, TECHNICAL ARTIST  
Allene T. Shimomura, PROOFREADER

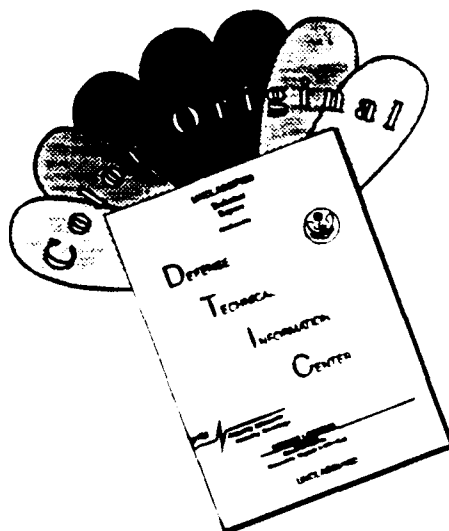
THE LINCOLN LABORATORY JOURNAL (ISSN 0896-4130) is published by Lincoln Laboratory, Massachusetts Institute of Technology, 244 Wood St., Lexington, MA 02173-9108. Subscriptions are free of charge, but provided only to qualified recipients (government employees and contractors, libraries, university faculty, and R & D laboratories). Requests for individual copies, subscriptions, or permission to reprint articles should be submitted to the Subscription Coordinator, Room A-222, at the above address. Phone (617) 981-7026, or [journal@ll.mit.edu](mailto:journal@ll.mit.edu) on Internet e-mail.

The work reported in this journal was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. The views expressed in this journal are those of the authors and do not reflect the official policy or position of the United States Government.

Postmaster: Please send address changes to the above address.

©1993 by Lincoln Laboratory, Massachusetts Institute of Technology. All rights reserved.

# DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

**3 An Overview of Automatic Target Recognition***Dan E. Dudgeon and Richard T. Lacoss*

In this article we introduce the subject of automatic target recognition (ATR). Interest in ATR is increasing in the defense community as the need for precision strikes in limited warfare situations becomes an increasingly important part of our defense posture. We discuss the difficulty of the ATR problem and we survey the variety of approaches that try to solve the problem. We conclude by introducing the other articles in this issue of the *Lincoln Laboratory Journal*.

**11 Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System***Leslie M. Novak, Gregory J. Owirka, and Christine M. Netishen*

Lincoln Laboratory is investigating the detection, discrimination, and classification of ground targets in high-resolution, fully polarimetric, synthetic-aperture radar (SAR) imagery. This article summarizes our work in SAR ATR by discussing the prescreeener, discrimination, and classification algorithms we have developed. The prescreeener required a low threshold to detect most of the targets in the data, which resulted in a high density of false alarms. The discriminator and classifier stages then reduced this false-alarm density by a factor of 100. We improved target-detection performance by using fully polarimetric imagery processed by the polarimetric whitening filter (PWF), rather than by using single-channel imagery. The PWF-processed imagery improved the probability of correct classification in a four-class classifier.

**25 Discriminating Targets from Clutter***Daniel E. Kreithen, Shawn D. Halversen, and Gregory J. Owirka*

The Lincoln Laboratory multistage target-detection algorithm for SAR imagery can be separated into three stages: the prescreeener, the discriminator, and the classifier. In this article, we examine fifteen features that are used in the discrimination algorithm. The set of best features from this pool of fifteen was determined by a theoretical analysis, and was then verified by using real SAR data. Performance was evaluated for a number of different cases; in all cases the theoretical performance analysis closely matched the real data performance. In addition, we formulate a set of criteria for best feature choice that apply to quadratic discrimination algorithms in general.

**53 Improving a Template-Based Classifier in a SAR Automatic Target Recognition System by Using 3-D Target Information***Shawn M. Verbout, William W. Irving, and Amanda S. Hanes*

We propose an improved version of a conventional template-matching classifier currently used in an operational ATR system for SAR imagery. This classifier was originally designed to maintain a library of 2-D reference images formed at a variety of radar viewing directions. The classifier accepts an input image of a target of unknown type, correlates this image with a reference template selected from each target library, and then classifies this image to the target category with the highest correlation score. The algorithm produces surprisingly poor classification results for some target types, however, because of differences in SAR geometry between the input image and the best-matching reference image. We correct this deficiency by incorporating a model-based reference generation procedure into the original classifier.

**77 Neural Systems for Automatic Target Learning and Recognition***Allen M. Waxman, Michael Seibert, Ann Marie Bernardon, and David A. Fay*

We have designed and implemented several computational neural systems for the automatic learning and recognition of targets in both passive visible and synthetic-aperture radar (SAR) imagery. Motivated by biological vision systems (in particular, that of the macaque monkey), our computational neural systems employ a variety of neural networks. In this article we present an overview of our research for the past several years, highlighting our earlier work on the unsupervised learning of three-dimensional objects as applied to aircraft recognition in the passive visible domain, the recent modification of this system with application to the learning and recognition of tactical targets from SAR imagery, the further application of this system to reentry-vehicle recognition from inverse SAR, or ISAR, imagery, and the incorporation of this recognition system on a mobile robot called the Mobile Adaptive Visual Navigator (MAVIN) at Lincoln Laboratory.

DTIC QUALITY INSPECTED 2

Dist	Approved for Special
A1	



---

117 **Multidimensional Automatic Target Recognition System Evaluation**

*Paul J. Kolodzy*

We are developing an evaluation facility that includes an electronic terrain board (ETB) to provide an effective test environment for ATR systems. The input to the ETB is very high-resolution data taken in the modalities of interest (laser radar, passive IR, and visible). The ETB contains sensor and target models so that measured imagery can be modified for sensitivity analyses. The evaluation facility also contains a reconfigurable suite of ATR algorithms that can be interfaced to real and synthetic data for developing and testing ATR modules. This article presents a description of the infrared airborne radar used to gather sensor data, a discussion of sensor fusion and the hybrid ATR measurement system, and a review of the ATR evaluation facility. We give results of processing real and synthetic imagery with the ATR system, with an emphasis on interpreting results with respect to sensor design.

---

147 **An Efficient MRF Image-Restoration Technique Using Deterministic Scale-Based Optimization**

*Murali M. Menon*

A method for performing piecewise smooth restorations on images corrupted with high levels of noise has been developed. Based on a Markov Random Field (MRF) model, the method uses a neural network sigmoid nonlinearity between pixels in the image to produce a restoration with sharp boundaries while providing noise reduction. The model equations are solved with the Gradient Descent Gain Annealing (GDGA) method—an efficient deterministic search algorithm that typically requires fewer than 200 iterations for image restoration when implemented as a digital computer simulation. A novel feature of the GDGA method is that it *automatically* develops an annealing schedule by adaptively selecting the scale step size during iteration.

---

161 **Machine Intelligent Automatic Recognition of Critical Mobile Targets in Laser Radar Imagery**

*Richard L. Delaney, Jacques G. Verly, and Dan E. Dudgeon*

A variety of machine intelligence (MI) techniques have been developed at Lincoln Laboratory to increase the performance reliability of automatic target recognition (ATR) systems. Useful for recognizing targets that are only marginally visible (due to sensor limitations or to the intentional concealment of the targets), these MI techniques have become integral parts of the Experimental Target Recognition System (XTRS)—a general-purpose system for model-based ATR. Using laser radar images collected by an airborne sensor, the prototype system recognized a variety of semi-trailer trucks with high reliability, even though the trucks were deployed in high-clutter environments.

---

187 **Machine Intelligent Gust Front Detection**

*Richard L. Delaney and Seth W. Troxel*

Techniques of low-level machine intelligence, originally developed at Lincoln Laboratory to recognize military ground vehicles obscured by camouflage and foliage, are being used to detect gust fronts in Doppler weather radar imagery. This Machine Intelligent Gust Front Algorithm (MIGFA) is part of a suite of hazardous-weather-detection functions being developed under contract with the Federal Aviation Administration. MIGFA has demonstrated levels of detection performance that have not only markedly exceeded the capabilities of existing gust front algorithms, but that are competitive with human interpreters.

---

213 **Extracting Target Features from Angle-Angle and Range-Doppler Images**

*Su May Hsu*

For diffuse targets, features such as shape, size, and motion can be determined from a time series of images from either angle-angle passive telescopes or range-Doppler radars. The extracted target features can then be used for automated target recognition and identification. An algorithm that uses scene-analysis techniques has been developed to perform the feature extraction.

# An Overview of Automatic Target Recognition

Dan E. Dudgeon and Richard T. Lacoss

■ In this article we introduce the subject of automatic target recognition (ATR). Interest in ATR is increasing in the defense community as the need for precision strikes in limited warfare situations becomes an increasingly important part of our defense posture. We discuss reasons for the difficulty of the ATR problem and we survey the variety of approaches that try to solve the problem. We conclude by introducing the remaining articles in this special issue of the *Lincoln Laboratory Journal*.

AUTOMATIC TARGET RECOGNITION (ATR) generally refers to the use of computer processing to detect and recognize target signatures in sensor data. The sensor data are usually an image from a forward-looking infrared (FLIR) camera, a synthetic-aperture radar (SAR), a television camera, or a laser radar, although ATR techniques can be applied to non-imaging sensors as well. ATR has become increasingly important in modern defense strategy because it permits precision strikes against certain tactical targets with reduced risk and increased efficiency, while minimizing collateral damage to other objects. If computers can be made to detect and recognize targets automatically, the workload of a pilot can be reduced and the accuracy and efficiency of the pilot's weapons can be improved.

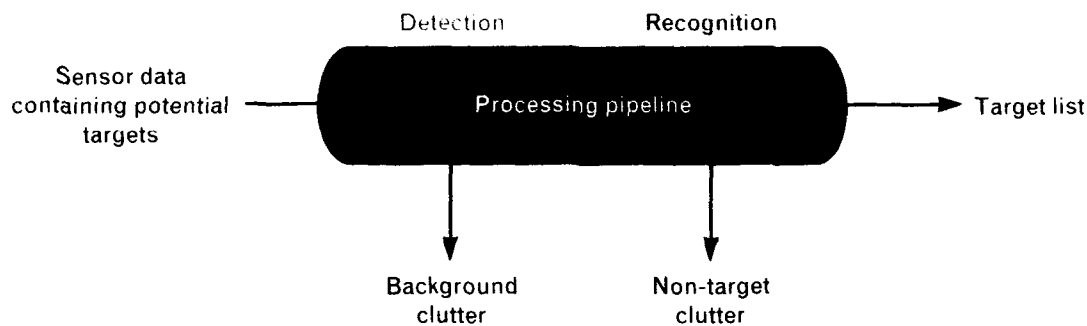
ATR technology can also be applied to non-military problems as well. For example, the problem of recognizing landmarks seen by a visual navigation system or a robotic system is related to the ATR problem. The recognition of particular objects or faces in photographs or video sequences is also related to ATR. We can think of the ATR problem as one part of the general problem of machine vision; namely, how can computers be made to do what we humans do so easily and naturally?

The fundamental problem of ATR is to detect and recognize objects of interest (targets) in an environment of clutter imaged by an imperfect sensor that introduces noise into the resulting signal. The defini-

tions of target, noise, and clutter depend upon the application. Target categories can be coarse (e.g., a treaded ground vehicle) or fine (e.g., a specific type of tank or even a specific tank). Often the term *classification* is used for coarse categorization and the term *identification* is used for fine categorization, although they are also used synonymously with the term *recognition*. Unfortunately, usage is not consistent.

Clutter refers to real things that are imaged (buildings, cars, trucks, grass, trees, and other objects) but are not targets of interest. Sometimes a distinction is made between naturally occurring clutter (grass, trees, topographical features) and man-made cultural clutter (buildings, vehicles, and other works). Clutter tends to dominate the imagery simply because targets are generally sparse compared to the environment in which they operate. Noise refers to electronic noise in the sensor as well as inaccuracies introduced in the computations by a signal processor. Depending on the ATR application, the problem may be one of extracting a signal from noise or it may be one of separating a target from its surrounding clutter.

The distinction between detection and recognition is ill defined. We could argue that recognition is just the detection of a specific target type. But algorithms developed from this viewpoint tend to require prohibitive amounts of processing. For this reason, ATR systems generally include a front-end detection stage. The goal of the detection stage is to eliminate most of the sensor data from further consideration without



**FIGURE 1.** Conceptual data flow in automatic target recognition (ATR) systems. Simple detection algorithms are applied to all the sensor data to isolate small portions that might contain targets. More complex recognition algorithms then process the selected portions of the data to reject non-target clutter and classify targets. Ideally, all targets of interest pass through the pipeline and are included in the output target list.

eliminating any of the targets of interest. In this context, the term detection means that something interesting has been discovered, and this discovery requires further analysis. For example, a small cluster of bright pixels in an image could indicate the presence of an object. Computationally simple detection algorithms are required at this stage because all the sensor data in the input image must be examined.

Practical implementations of ATR systems can be viewed as pipeline processing systems, as illustrated in Figure 1. Ideally, all targets of interest pass through the pipeline and are included in the output target list. As data move through the pipeline, the processing algorithms become more target specific and computationally intensive for each data item, while the number of data items processed and the number of clutter false alarms each decreases. Even with this structure, the front-end detection stages of the processing pipeline often require the most computational power because the ATR system must search large amounts of imagery to find a few instances of the target.

For a specific ATR problem, both the target signatures and the clutter background can vary. In thermal images, for example, a tank can be hotter or colder than its background (causing positive or negative thermal contrast). It can also exhibit one shape when viewed from the side and another shape when viewed from the front; its turret can be rotated to any position and its gun barrel can take on a range of elevation angles. The background clutter can be a benign

meadow, a treeline, a forest road, a featureless desert, or an urban area. Such complicated variations in both target signature and background clutter contribute significantly to the difficulty of the ATR problem.

### ATR Technology

Many technologies and techniques are utilized in attempting to solve the problems of ATR, as illustrated in Figure 2. Sensor technology is critical because performance is ultimately limited by the quality of the information provided by the sensors. Processing hardware is also critical because ATR algorithms must process large quantities of data, often in real time, and because system development can be significantly hindered by the lack of processing capacity. Software, simulation, and evaluation methodologies and tools are also important elements. In addition, important but indirect contributions are made by neural and cognitive sciences, statistics, and sensor physics.

Figure 2 also identifies several algorithmic approaches to the problem of ATR; the multiplicity of these approaches indicates that no satisfactory single approach has yet been found. The most successful ATR systems will probably blend several algorithmic techniques to get satisfactory performance.

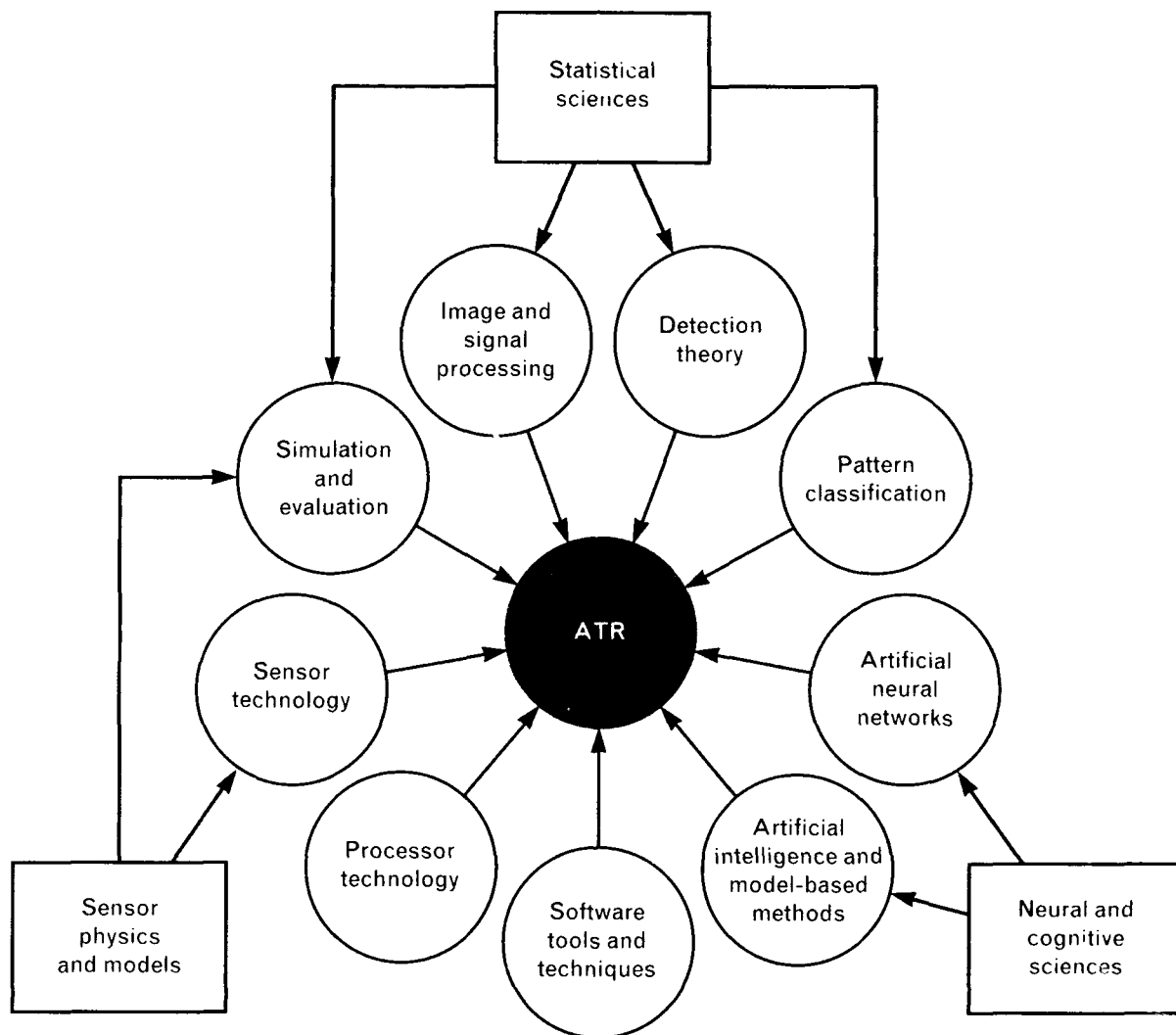
### Detection Theory

ATR is based in part on detection theory and related statistical ideas that date back to the early days of radar processing. If a target signature is deterministic

or if it is stochastic with well-defined stationary statistics, and if the clutter is well characterized as a stationary random process, then optimal detection techniques such as the matched filter can be derived. In classical detection-theory problems, a trade-off exists between the probability of detecting a target signature when the target is there, and the probability of declaring that a target is present when in fact it is not there (i.e., a false alarm). The objective of optimal detection processing is to separate the distributions of target

signatures and clutter signatures so that they can be distinguished by a simple statistical test.

A seductive aspect of the detection-theory approach is that it provides a firm theoretical foundation for both the development of algorithms and the understanding of their performance. This approach to ATR, however, requires valid and analytically (or computationally) tractable statistical models for both targets and clutter. Such models are difficult to develop, and this approach is greatly complicated by signature and



**FIGURE 2.** ATR technologies and processing methods. The circles identify some of the technologies and methods that are required to solve difficult ATR problems, although not all technologies and processing methods are used in all cases. The three boxes show some of the basic supporting research activities.

clutter variability. Detection theory is conceptually appealing, but it has had only limited success for ATR problems.

### *Pattern Recognition*

Pattern recognition is the most mature approach used for ATR applications. Target-signature representation options range from two-dimensional image templates to lower-dimensional vectors of features that are designed to be differentially sensitive to targets and non-targets. Recognition depends on feature vectors (or templates thought of as vectors) for targets clustering together and being distinct from non-target clusters.

Potential targets (detections) are confirmed by comparing target images or feature vectors with a database of target and non-target exemplars. Recognition consists of selecting the best match between the target data and the exemplar database. The matching criteria may be ad hoc (e.g., mean-square differences between data and exemplar vectors), or they may be based on statistical assumptions that give the appearance of a more rational basis.

Pattern recognition is not disjoint from the detection-theory approaches discussed above or from the artificial neural network approaches discussed below. A close functional similarity exists between template (or feature vector) matching and matched filtering; differences between the two are at a more detailed level. Detection theory emphasizes optimal (or near-optimal) algorithms that are derived by using statistical models of raw data. Pattern recognition, which also relies heavily on statistics, includes more ad hoc approaches (e.g., spectral coefficients, fractal dimensions, and blob aspect ratios), especially in the definition and extraction of the features used to characterize targets.

### *Artificial Neural Networks*

The neural network approach places even more emphasis upon experiential learning-by-example than does the pattern recognition approach. Neural networks represent a processing paradigm motivated by the human visual system, which remains the most flexible and robust target-recognition system for imagery that we know. The goal of neural network approaches is to develop an architecture that reproduces

the flexibility and robustness of the human visual system in a piece of equipment.

In the nearer term, neural networks can have implementation advantages because they are highly parallel, and their ability to learn by example can make them capable of discovering and using signature differences that distinguish different types of targets. Neural networks are a type of nonlinear processing that could have advantages over classical detection-theoretic techniques or pattern-recognition techniques. One challenge for neural network approaches is to achieve good performance over the entire range of target signatures and background-clutter conditions, given limited training data.

Neural network learning can be categorized as supervised or unsupervised. Supervised networks are trained by using independently classified images or feature vectors. Training typically consists of computerized adjustment of parameters to optimize performance on the training set. Performance on new data depends on how well the training data are classified and how representative they are. Unsupervised networks define their own internal classifications of data, independent of the external source. Data are clustered together if they look similar to other members of the cluster. When new data are not similar enough to any existing cluster, a new cluster is formed.

Unsupervised networks, however, ultimately require supervised training to perform useful recognition. A target class must be associated with each internal cluster learned by the network. Typically, a cluster is assigned the target name corresponding to the majority (or even plurality) of training examples in the cluster. This training is often an iterative process in which parameters of the unsupervised network are adjusted by the ATR algorithm developer until each of its clusters tends to be dominated by only one target type. Multiple clusters of data can be assigned the same target type.

### *Model-Based Target Recognition*

The genesis of model-based target-recognition methods is in artificial intelligence as it is applied to image understanding. Two characteristics typify the model-based approach: (1) matching of processed sensor data to predictions based on hypotheses concerning

the target type, pose, and range, and (2) matching of processed sensor data on the basis of multiple localized features.

The difference between model-based target features and the feature vectors of other approaches is important. The elements of feature vectors are not features in the model-based sense. Corners, bright points, line segments, and small regions corresponding to identifiable parts of a target (e.g., the gun, turret, or body of a tank) might be features for a model-based system. Recognition depends on matching parts and interrelationships between the parts. Flexible and sophisticated matching techniques can be used, in principle, to design systems that are more robust with regard to obscuration and target variability. In contrast, the elements of feature vectors are usually global target measures such as the coefficients of a transform that represents the target image or a texture measure such as fractal dimension.

In some sense, every ATR algorithm is model based because every algorithm makes and uses *a priori* assumptions about target and clutter characteristics. The representation of the *a priori* information can vary widely, however, and it is conceptually different for model-based and non-model-based approaches. Generally, in model-based target recognition the appearance of the target in an image is modeled.

The search-and-match approaches of model-based methods also tends to distinguish them from other algorithmic approaches. The general paradigm is as follows: first form an initial set of hypotheses based on the sensor data (the *indexing* problem), then use the hypotheses to predict features and their relationships, and finally compare the predictions to features extracted from the data. This approach is quite different from conducting a computationally intensive exhaustive search for a best matching pattern.

Continuing areas of research in model-based target recognition include matching and evidence accumulation (i.e., what is the most robust way to match hypothesized signatures to the sensor data), indexing (how to generate a small number of hypotheses that contain the correct target), and modeling. In some systems the models are built by hand; that is, analysts examine sensor data, observe target signature characteristics, and encode them into some form of data

representation that is convenient for matching. This approach is time consuming, and consequently there is a need to build models automatically from data. For example, the radar image-understanding system discussed in an earlier article in the *Lincoln Laboratory Journal* can build target models of reentry vehicles from radar data [1]. (Note that a system does not have to be a neural network to be capable of learning from data.)

#### *Data Requirements for ATR System Development*

ATR system development and evaluation requires an enormous quantity of data because of the variability in target signatures and background clutter. The assembly of large, realistic, experimental databases, however, is time consuming and expensive. As a result, we need to develop techniques that minimize data requirements or, to put it differently, utilize experimental data more effectively. Simulation is one such approach; limited experimental data can be used to develop and validate simulation models that can then be used to generate data for system development and evaluation.

If the statistics of a target signature can be modeled parametrically, then a small amount of experimental data might be used to determine the model parameters. Another approach is to develop physical models of image formation and use them to create synthetic target signatures in both real-clutter and synthetic-clutter backgrounds. This approach doesn't reduce the need for data; it simply allows us to create the data in a computer rather than collect real data with a real sensor observing real targets in real backgrounds in all their various states. Of course, the synthetic imagery must be realistic to be useful, and realism is determined by comparing the synthetic data to a real database. Fortunately, both model development and synthetic scene generation are done off line, so that real-time performance is not required. But the issue of how to develop and evaluate ATR systems in a reasonable amount of time with a reasonable amount of data is still open.

#### **In This Issue**

We are fortunate at Lincoln Laboratory to have several state-of-the-art sensors for collecting data that

can be used to develop target detection and recognition systems. The in-house availability of these sensors is reflected in the articles included in this issue. The articles emphasize research that utilizes data from synthetic-aperture radars, laser radars, range-Doppler imaging radars, and air surveillance radars. Even when the work has not involved such sophisticated sensors, it is nevertheless experimentally oriented and it emphasizes the use of real sensor data. Realistic experimental data stimulate the work and provide a rich source of information for understanding and exploiting how targets and the clutter backgrounds in which they are embedded can be modeled and separated.

In this issue, we have collected nine other articles detailing various aspects of ATR. We begin by focusing on polarimetric synthetic-aperture radar imagery and the techniques used to detect, discriminate, and classify targets found in such imagery. The article by Leslie M. Novak et al. gives an overview of a classical pattern-recognition approach. The article by Daniel E. Kreithen et al. discusses the problem of discriminating targets from the natural-clutter objects (e.g., trees and bushes) that reflect enough radar energy to pass the detection stage. The article by Shawn M. Verbout et al. investigates some of the problems in recognizing three-dimensional targets from their two-dimensional SAR signatures, and develops a technique for incorporating three-dimensional information in the classification process.

The article by Allen M. Waxman et al. introduces the topic of neural network recognition systems. It discusses some of the motivation for neural network architectures that are derived from biological vision systems. It also presents the idea that information can be extracted from the changes in target signature from one viewpoint to the next as a target and a sensor move past each other. These neural-processing concepts are demonstrated on several different target-recognition problems.

The theme of neural networks continues in articles by Paul J. Kolodzy and by Murali M. Menon. The article by Kolodzy describes laser radar imagery, which simultaneously captures reflectivity information and range information. The article also addresses the problem of evaluating ATR algorithms. Because of the cost of collecting accurate training and test imagery

spanning the breadth of possible ATR scenarios, the development of an electronic terrain board (which is a system that can generate synthetic imagery accurately and quickly) for testing and evaluating ATR algorithms may be more cost effective.

The article by Menon discusses the use of neural networks to implement an image-enhancement algorithm, based on Markov random fields, for laser radar imagery. The effect of this preprocessing step on the performance of the target classifier is demonstrated.

The article by Richard L. Delanoy et al. also demonstrates the use of laser radar measurements for target recognition, but here the authors use a model-based approach. Detection is done by several algorithms that operate in parallel and are keyed to look for various image features by using a process called *functional template correlation*. Multiple functional-template outputs are combined to form a simple interest image to focus subsequent processing. Target information is extracted and decomposed into features that are matched against stored appearance models. The weights used to govern the matching process can be adaptively learned on training data. The appearance models themselves can also be built semi-automatically from training data.

The article by Delanoy and Seth W. Troxel demonstrates the flexibility and power of functional template correlation. This article discusses how functional templates are used to detect and recognize gust fronts in weather radar data. The algorithm was recently implemented in real time and demonstrated on the ASR-9 surveillance radar system at Orlando International Airport in Florida.

The issue concludes with an article by Su May Hsu on research in ballistic missile defense. This article discusses the use of image processing techniques to extract target features for classification of reentering objects.

### **The Future of ATR**

ATR continues to be an important defense technology. While great progress has been made in sensor technology and processing-hardware technology, the development of recognition algorithms is far from complete. Efforts are continuing to accelerate the de-

velopment of these algorithms. These efforts are sponsored by ARPA, the Army Night Vision Electronic Sensors Directorate (formerly the Night Vision Lab), the Air Force Wright Laboratory, the Naval Air Warfare Center (China Lake), and other government laboratories. Common databases and evaluation techniques are being developed so that ATR approaches can be objectively compared.

Because of the amount of data required to develop and test ATR systems for robustness, and the cost of mounting realistic data collection efforts, the development of accurate synthetic data-generation tools is also being addressed. Organizations such as the ATR Working Group (a joint industry-government working group) and the Department of Defense Working Group on ATR bring together government, industry, and academic laboratories to exchange views on the problems and potential solutions in ATR technology.

Just as the speech-recognition problem required decades of serious research before simple applications could be brought to market, the ATR problem needs a wide range of research before we can expect practical systems. Most people believe the ATR problem will not be solved by a single brilliant idea. The solution will probably require a combination of improved sensors, faster computers, and better algorithms. From our point of view, the area is extremely interesting because it involves so many technical disciplines (material science, sensor technology, signal and image processing, detection theory, algorithm development, computer technology, data representation, pattern classification, and neural networks) and because it has so many potential applications.

---

## REFERENCES

1. A.M. Aull, R.A. Gabel, and T.J. Goblick, "Real-Time Radar Image Understanding: A Machine Intelligent Approach," *Time Lab. J.* 5, 195 (1992).





**DAN E. DUDGEON**

is a senior staff member in the Machine Intelligence Technology group. His research specialties are in image processing, computer vision, and computer architectures for algorithm development and real-time implementations. Before joining Lincoln Laboratory in 1979, he worked at Bolt, Beranek, and Newman, Inc. in Cambridge, Massachusetts. Dan received S.B., S.M., and E.E. degrees in electrical engineering, and a Sc.D. degree in signal processing, all from MIT. In 1976 he was corecipient of IEEE's Browder J. Thompson Memorial Prize for best paper by an author under the age of 30. He is the coauthor of *Multidimensional Digital Signal Processing* and *Array Signal Processing*. Dan was elected an IEEE Fellow in 1987, and he was named a Distinguished Lecturer of the ASSP Society in 1988. He is currently taking ice skating lessons in a futile attempt to keep pace with his daughter's athletic endeavors.



**RICHARD T. LACOSS**

is the leader of the Machine Intelligence Technology group. He received a B.A. degree and a B.S. degree in electrical engineering from Columbia University, and a Ph.D. degree in electrical engineering from the University of California at Berkeley. From 1965 to 1978 he worked in seismic discrimination and signal processing research at Lincoln Laboratory. His research now includes image understanding, expert systems, seismo-acoustic surveillance, neural networks, and parallel processing. He has taught in both the Earth and Planetary Sciences Department and the Electrical Engineering Department at MIT, and he is the author of many articles and reports in seismology and signal processing. His two-year old son and his three-year-old daughter also keep him very busy these days.

# Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System

Leslie M. Novak, Gregory J. Owirka, and Christine M. Netishen

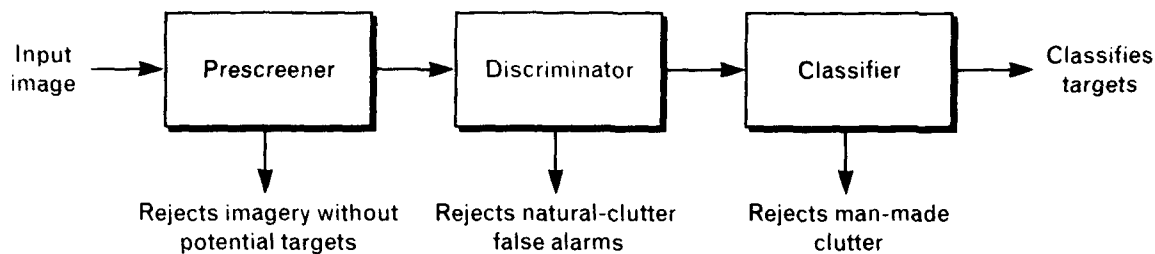
■ Lincoln Laboratory is investigating the detection, discrimination, and classification of ground targets in high-resolution, fully polarimetric, synthetic-aperture radar (SAR) imagery. This paper summarizes our work in SAR automatic target recognition by discussing the prescreening, discrimination, and classification algorithms we have developed; data from 5 km<sup>2</sup> of clutter and 339 targets were used to study the performance of these algorithms. The prescreener required a low threshold to detect most of the targets in the data, which resulted in a high density of false alarms. The discriminator and classifier stages then reduced this false-alarm density by a factor of 100. We improved target-detection performance by using fully polarimetric imagery processed by the polarimetric whitening filter (PWF), rather than by using single-channel imagery. In addition, the PWF-processed imagery improved the probability of correct classification in a four-class (tank, armored personnel carrier, howitzer, or clutter) classifier.

THE ARPA-SPONSORED WARBREAKER program is a broad-based advanced technology program to develop new weapons technology that can locate and destroy critical mobile targets such as SCUD launch systems and other highly mobile platforms. Automatic target recognition (ATR) is an important candidate technology for this effort. To address the surveillance and targeting aspects of the Warbreaker program, Lincoln Laboratory has developed a complete, end-to-end, 2-D synthetic-aperture radar (SAR) ATR system. This system requires a sensor that can search large areas and also provide fine enough resolution to detect and identify mobile targets in a variety of landscapes and deployments.

The Lincoln Laboratory ATR system has three basic stages: detection (or *prescreening*), discrimina-

tion, and classification (see Figure 1). In the prescreening stage, a two-parameter constant-false-alarm-rate (CFAR) detector selects candidate targets in a SAR image by examining the amplitude of the radar signal in each pixel of the image. In the discrimination stage, a target-sized 2-D matched filter accurately locates the candidate targets and determines their orientation. Then texture-discrimination features (standard deviation, fractal dimension, and weighted-rank fill ratio) are used to reject natural-clutter false alarms [1]. In the classification stage, a 2-D pattern-matching algorithm rejects cultural-clutter false alarms (i.e., man-made objects that are not targets) and classifies the remaining detections by target type (tank, armored personnel carrier, or howitzer).

To evaluate the performance of the ATR system,



**FIGURE 1.** Block diagram of the SAR automatic target recognition system. The prescreener locates candidate targets based on the brightness of pixels in the input image, the discriminator rejects natural-clutter false alarms, and the classifier rejects non-target cultural clutter and classifies targets by vehicle type.

we used high-resolution (1 ft by 1 ft), fully polarimetric target data and clutter data gathered by the Lincoln Laboratory millimeter-wave SAR [2] at a depression angle of  $22.5^\circ$  and a slant range of 7 km. We demonstrated the robustness of the ATR system by testing it against targets both with and without radar camouflage.

Figure 2 is an example of the quality of the imagery gathered by the Lincoln Laboratory SAR. In this image of a golf course in Stockbridge, New York, the high resolution of the SAR resolves individual trees and bushes as well as small objects such as the flagpole located in the center of the putting green. This particular SAR image was obtained under clear weather conditions; the quality and resolution of the image would not have been degraded, however, by dense fog or thick cloud cover. Thus a SAR sensor has a significant advantage over optical sensors. SAR image quality is not dependent on weather conditions, and the sensor can be used at any time of day or night. In addition, SAR sensors can perform other tasks, such as searching large areas from a long distance.

The image in Figure 2 was constructed from fully polarimetric SAR data that were processed with a technique known as the polarimetric whitening filter (PWF) [3]. PWF processing optimally combines the HH (horizontal transmit, horizontal receive), HV (horizontal transmit, vertical receive), and VV (vertical transmit, vertical receive) polarization components of the radar return. This polarimetric combination enhances the quality of the imagery in two ways: (1) the amount of speckle in the imagery is minimized, and (2) the edges of objects in the image (such

as the pond) are sharper. As a result, PWF-processed imagery is visually clearer than single-polarimetric-channel imagery. In addition, PWF-processed imagery improves the performance of all three stages of the ATR system (compared with the performance achieved by using single-polarimetric-channel imagery) because PWF processing reduces clutter variance and enhances target signatures relative to the clutter background.

This article begins with an overview of the three stages of the baseline ATR system. Next we describe the performance of the ATR system with both camouflaged and uncamouflaged targets, and then we compare performance using PWF data with performance using single-channel (HH) data. We also present details of the three discrimination features used in our studies, with particular emphasis on the fractal-dimension feature. Finally, we discuss future improvements to the discrimination and classification stages.

### Overview of the Baseline ATR System

This section describes our three-stage baseline SAR ATR system, which is illustrated in Figure 1 by a simplified block diagram. The three stages—prescreener, discriminator, and classifier—are described below.

#### *Stage 1: Prescreener*

In the first stage of processing, a two-parameter CFAR detector [4] is used as a prescreener; this stage of processing identifies potential targets in the image on the basis of radar amplitude (i.e., by searching for bright returns). Computation time for this stage of processing is significantly reduced by operating the

detector at a reduced resolution (1 m by 1 m) rather than at the full resolution (1 ft by 1 ft).

Figure 3 is a sketch of the two-parameter CFAR detector used by the prescreener; the detector is defined by the rule

$$\begin{aligned} \frac{X_t - \hat{\mu}_c}{\hat{\sigma}_c} &> K_{\text{CFAR}} \Rightarrow \text{target}, \\ \frac{X_t - \hat{\mu}_c}{\hat{\sigma}_c} &\leq K_{\text{CFAR}} \Rightarrow \text{clutter}, \end{aligned} \quad (1)$$

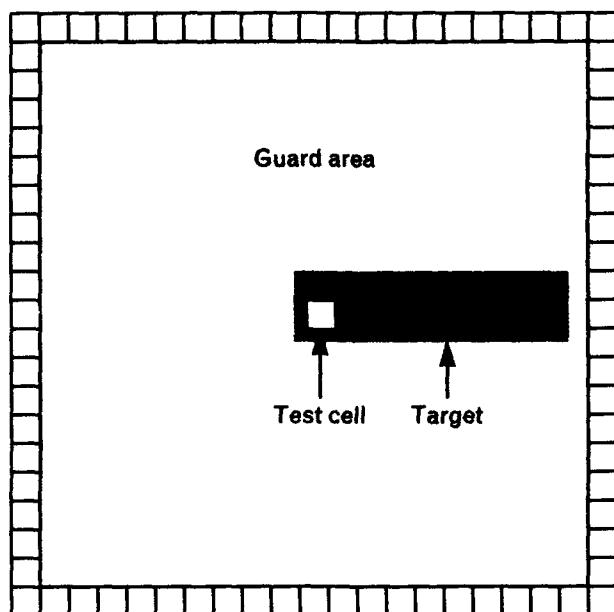
where  $X_t$  is the amplitude of the test cell,  $\hat{\mu}_c$  is the estimated mean of the clutter amplitude,  $\hat{\sigma}_c$  is the estimated standard deviation of the clutter amplitude,

and  $K_{\text{CFAR}}$  is a constant threshold value that defines the false-alarm rate. As shown in the figure, the test cell is at the center of a defined local region, and the 80 cells in the boundary stencil are used to estimate the mean and standard deviation of the local clutter. The guard area ensures that no target cells are included in the estimation of the clutter statistics. If the detection statistic calculated in Equation 1 exceeds  $K_{\text{CFAR}}$ , the test cell is declared to be a target pixel; if not, it is declared to be a clutter pixel.

When the amplitude distribution of the clutter is Gaussian, the CFAR detector provides a constant false-alarm rate for any given  $K_{\text{CFAR}}$  [5]. Because the clutter distributions of high-resolution data are typi-



**FIGURE 2.** High resolution (1 ft by 1 ft) synthetic-aperture radar (SAR) image of a golf course near Stockbridge, New York. Polarimetric whitening filter (PWF) processing was used to produce this minimum-speckle image. The radar is located at the top of the image; therefore, the radar shadows go toward the bottom of the page. Notice that the SAR can resolve details as small as the flagpole in the putting green near the center of the image.



**FIGURE 3.** The prescreener CFAR detector. The amplitude of the test cell is compared with the mean and standard deviation of the clutter. The boundary consists of 80 cells that are used for clutter statistics estimation. Each cell in the boundary consists of 16 raw pixels that are noncoherently averaged. The guard area ensures that no target cells are included in the clutter statistics estimation.

cally not Gaussian [6], however, the detector does not always yield a constant false-alarm rate. In spite of this fact, the detector given by Equation 1 still proves to be an effective algorithm for detecting targets in clutter.

Only those test cells whose amplitudes stand out from the surrounding cells are declared to be targets. The higher we set the threshold value of  $K_{CFAR}$ , the more a test cell must stand out from its background for the cell to be declared a target. Because a single target can produce multiple CFAR detections, the detected pixels are clustered (grouped together) by the detector if they are within a target-sized neighborhood. Then a 120-ft-by-120-ft region of interest of full resolution (1 ft by 1 ft) data around each cluster centroid is extracted and passed to the discrimination stage of the algorithm for further processing.

#### Stage 2: Discriminator

The discrimination stage takes as its input the 120-ft-by-120-ft regions of interest passed to it by the prescreener, and it analyzes each region of interest at

full resolution (1 ft by 1 ft). The goal of discrimination processing is to reject the regions containing natural-clutter false alarms while accepting the regions containing real targets. This stage consists of three steps: (1) determining the position and orientation of a detected object, (2) computing simple textural features, and (3) combining the features into a discrimination statistic that measures how targetlike the detected object is.

In the first step of the discrimination stage the algorithm determines the position and orientation of the target by placing a target-sized rectangular template on the image. The algorithm then slides and rotates the template until the energy within the template is maximized. The position estimate produced in the discrimination stage is more accurate than the position estimate produced in the prescreening stage. This operation is computationally feasible because it is performed only on the selected high-resolution regions of interest passed by the prescreener, and not on the entire image. Mathematically, this operation is equivalent to processing the data in the region of interest with a 2-D matched filter for the case when the orientation of the target is unknown.

In the second step of the discrimination stage the algorithm calculates three textural features: (1) the standard deviation of the data within the target-sized template, (2) the fractal dimension of the pixels in the region of interest, and (3) the weighted-rank fill ratio of the data within the template. The standard deviation of the data within the template is a statistical measure of the fluctuation of the pixel intensities; targets typically exhibit significantly larger standard deviations than natural clutter. The fractal dimension of the pixels in the region of interest provides information about the spatial distribution of the brightest scatterers of the detected object. It complements the standard-deviation feature, which depends only on the intensities of the scatterers and not on their spatial locations. The weighted-rank fill ratio of the data within the template measures the fraction of the total power contained in the brightest 5% of the detected object's scatterers. For targets, a significant portion of the total power comes from a small number of very bright scatterers; for natural clutter, the total power is distributed more evenly among the scatterers.

In the third step of the discrimination stage the algorithm combines the three textural features into a single discrimination statistic; this discrimination statistic is calculated as a quadratic distance measurement (see the accompanying article entitled "Discriminating Targets from Clutter" by Daniel E. Kreithen et al.). Most natural-clutter false alarms have a large quadratic distance and are rejected at this stage. Most man-made clutter discretely (such as buildings and bridges) pass the discrimination stage; therefore, the next stage—the classifier stage—must have the ability to reject them.

### *Stage 3: Classifier*

A 2-D pattern-matching classifier rejects cultural false alarms caused by man-made clutter discretely and then classifies target detections by vehicle type. In our studies we implemented a four-class classifier (tank, armored personnel carrier, howitzer, and clutter) using high-resolution (1 ft by 1 ft) PWF imagery. Detected objects that pass the discrimination stage are matched against stored references of the tank, armored personnel carrier, and howitzer. If none of the matches exceeds a minimum required score, the detected object is classified as clutter; otherwise, the detected object is assigned to the class (tank, armored personnel carrier, or howitzer) with the highest match score.

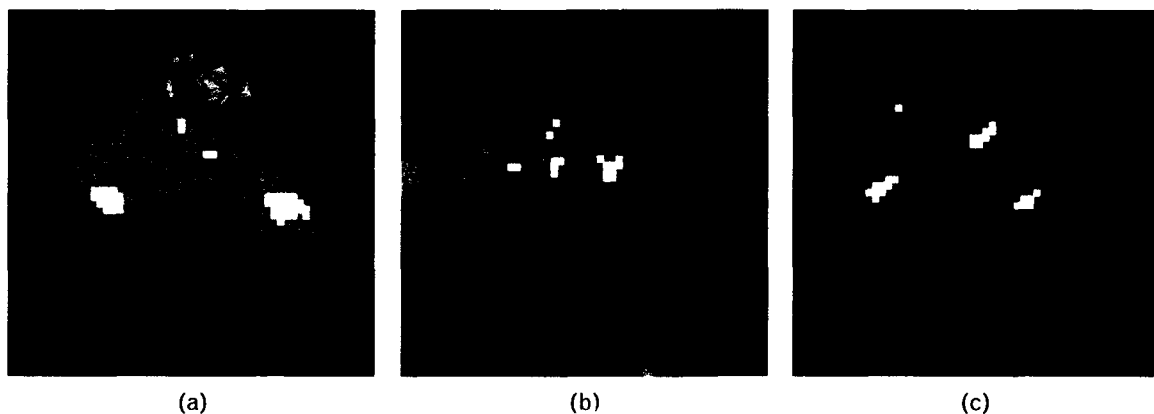
The pattern-matching references used in the classifier were constructed by averaging five consecutive

spotlight-mode images of a target collected at  $1^\circ$  increments of azimuth, yielding 72 smoothed images of each of the targets. Figure 4 shows typical pattern-matching references for the three targets at a particular aspect angle.

### **Performance of the Baseline ATR System**

This section describes the performance of the prescreening, discrimination, and classification stages of the baseline SAR ATR system. Clutter data from  $5 \text{ km}^2$  of ground area were processed through the ATR-system algorithms, along with data for 162 camouflaged targets and 177 uncamouflaged targets. The camouflaged target data used in this study represent a difficult scenario in which the test targets were realistically deployed and covered with radar camouflage. The training data used to design the ATR system were taken from the uncamouflaged targets. The clutter data contained a moderate number of man-made clutter discretely.

The CFAR detection threshold in the prescreener was set relatively low to obtain a high initial probability of detection ( $P_D$ ) for the target data. At the output of the prescreener,  $P_D = 1.00$  was obtained for the uncamouflaged targets, while  $P_D = 0.82$  was obtained for the camouflaged targets. At this CFAR threshold, a false-alarm density of approximately 30 false alarms per  $\text{km}^2$  ( $\text{FA}/\text{km}^2$ ) was obtained. The initial detection processing was carried out at reduced resolution (1 m by 1 m).



**FIGURE 4.** Typical pattern-matching reference templates for (a) a tank, (b) an armored personnel carrier, and (c) a howitzer. These pattern-matching references are used to classify detected objects by vehicle type. They are constructed by averaging five consecutive spotlight-mode images of a target collected at  $1^\circ$  increments of azimuth.

**Table 1. Overview of ATR System Performance.**

	$FA/km^2$	$P_D$ Uncamouflaged Targets*	$P_D$ Camouflaged Targets
After prescreening	30	1.00	0.82
After discrimination	3.0	1.00	0.75
After classification	0.3	1.00	0.70

\* The uncamouflaged target test data was used for algorithm training.

Each detected region of interest (containing a potential target) was passed to the discrimination stage for further processing at full resolution (1 ft by 1 ft). The discrimination stage determined the location and orientation of each detected object, and then calculated the textural features (standard deviation, fractal dimension, and weighted-rank fill ratio) that were used to reject natural-clutter discrettes. Discrimination processing reduced the false-alarm density by a factor of 10, to 3  $FA/km^2$ . No uncamouflaged targets were rejected by the textural-feature tests; thus the initial  $P_D$  of 1.00 was maintained. A few of the camouflaged targets were rejected by the textural-feature tests, which resulted in  $P_D = 0.75$  for these targets.

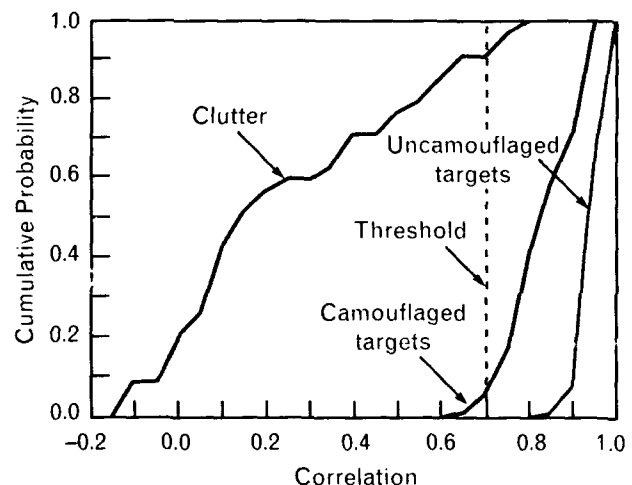
In the classification stage of processing, the 2-D pattern-matcher was applied to those detections which had passed the discrimination stage. Classification processing reduced the false-alarm rate by another factor of 10, to approximately 0.3  $FA/km^2$ . No uncamouflaged targets were rejected by the pattern matcher (resulting in  $P_D = 1.00$  for these targets), but some camouflaged targets were incorrectly classified as clutter (resulting in  $P_D = 0.70$  for these targets).

Table 1 summarizes the performance of all three stages of the ATR system. The uncamouflaged target data were used for training the discrimination and classification stages. The thresholds of the algorithms were set so that perfect performance was achieved with the uncamouflaged data. Once the thresholds had been set in this way, the clutter and camouflaged targets were processed.

Figure 5 illustrates how clutter rejection was imple-

mented by the pattern-matching classifier. As the figure shows, most of the clutter discrettes had correlation scores below the threshold value of 0.7, and thus were rejected (i.e., classified as clutter). Detected objects with correlation scores equal to or greater than 0.7 were considered to be sufficiently targetlike, and were classified according to their highest correlation scores (as tank, armored personnel carrier, or howitzer). Figure 5 also indicates that only a small fraction of the camouflaged targets were declared to be clutter because of low correlation scores.

The second function of the classifier is to assign



**FIGURE 5.** Overview of classifier performance. The clutter and target data were processed separately. Most of the clutter had correlation scores below the threshold value of 0.7, and was rejected. Detected objects above the threshold were classified according to their highest correlation scores.

**Table 2. Correlation Pattern-Matching Classifier Performance**

(training data: uncamouflaged targets;  
test data: camouflaged targets and clutter discretes)

	Tank	Percent Classified as		
		APC	Howitzer	Clutter
Tank	89%	11%	0%	0%
APC	0%	96%	0%	4%
Howitzer	0%	13%	71%	16%
Clutter	0%	14%	0%	86%

objects accepted as targets to target classes (tank, armored personnel carrier, howitzer). Table 2 shows the classification performance of the baseline classifier as a confusion matrix that tabulates the correct and incorrect classifications. Recall that the classifier used templates constructed from uncamouflaged targets; the classification results shown in Table 2 are for clutter discretes and camouflaged test targets that passed the detection and discrimination stages. At the output of the classification stage, 70% of the camou-

flagged targets were classified as targets, and 85% of those targets were correctly classified by vehicle type.

The four-class, 2-D pattern-matching algorithm used in this study was implemented with normalized dB references, which provided the best overall performance among five different reference schemes that were tested.

#### *ATR Performance Using Single-Channel Data versus Fully Polarimetric Data*

We compared the performance of the ATR system using single-channel (HH) data with the performance of the system using fully polarimetric PWF data. Figure 6(a) shows an HH-polarization SAR image of a scene processed to reduced resolution (1 m by 1 m). In this image two regions of trees are separated by a narrow strip of coarse scrub. Also visible in the image, although somewhat faint, are four power-line towers located in the scrub. Figure 6(b) shows the corresponding PWF-processed image of the scene. The power-line towers have greater intensity in the PWF image than in the HH image because the PWF image includes contributions from HH, HV, and VV polarizations.

Table 3 compares ATR system performance using HH versus PWF data. The comparison was performed by using the same target and clutter data used to



(a)



(b)

**FIGURE 6.** Comparison of (a) HH and (b) PWF imagery. The power-line towers are more clearly visible in the PWF image because PWF processing combines data from all three polarization channels (HH, HV, and VV).



**Table 3. Comparison of ATR System Performance Using HH Imagery versus PWF Imagery.**

	FA/km <sup>2</sup>	$P_D$ HH Data	$P_D$ PWF Data
After prescreening	30	0.65	0.82
After discrimination	3.0	0.57	0.75
After classification	0.3	0.24	0.70

generate the results in Table 1. At the output of each stage (prescreening, discrimination, and classification) the false-alarm densities were set equal for HH and PWF clutter imagery. This normalization permits us to compare the HH and PWF detection performance at each stage.

The detection performance was better with PWF data than with HH data. At the output of the detection stage  $P_D = 0.82$  for PWF data and  $P_D = 0.65$  for HH data. At the output of the discrimination stage  $P_D = 0.75$  for PWF data and  $P_D = 0.57$  for HH data. At the output of the classification stage  $P_D = 0.70$  for PWF data and  $P_D = 0.24$  for HH data; the  $P_D$  at the end of the classification stage represents the overall end-to-end performance of the ATR system.

#### Details of the Baseline Discrimination Features

This section presents details of the three baseline discrimination features: standard deviation, fractal dimension, and weighted-rank fill ratio. The equations for calculating each feature are also discussed. Because the concept of the fractal-dimension feature is fairly involved, this feature is discussed at greater length than the other two features.

##### Standard-Deviation Feature

The standard-deviation feature is a measure of the fluctuation in intensity, or radar cross section, in an image. The log standard deviation for a particular region is defined as the standard deviation of the radar returns (in dB) from the region. If the radar intensity in power from range  $r$  and azimuth  $a$  is denoted by  $P(r, a)$ , then the log standard deviation  $\sigma$

can be estimated as follows:

$$\hat{\sigma} = \sqrt{\frac{S_2 - S_1^2 / N}{N - 1}},$$

where

$$S_1 = \sum_{r,a \in \text{Region}} 10 \log_{10} P(r, a) \quad (2)$$

and

$$S_2 = \sum_{r,a \in \text{Region}} [10 \log_{10} P(r, a)]^2 \quad (3)$$

and  $N$  is the number of points in the region.

##### Fractal-Dimension Feature

The fractal-dimension feature provides a measure of the spatial distribution of the brightest scatterers in a region. In the following paragraphs we present the formal definition of fractal dimension, along with several simple examples to illustrate the definition. We also show how to calculate the fractal dimension of detected objects in a SAR image. By using high-resolution SAR imagery gathered at Stockbridge, New York, we demonstrate how the spatial texture differences measured by the fractal-dimension feature can be used to discriminate between natural and cultural objects.

The fractal dimension of a set  $S$  in a two-dimensional space can be defined as follows:

$$\dim(S) = \lim_{\epsilon \rightarrow 0} \frac{\log M_\epsilon}{\log \left( \frac{1}{\epsilon} \right)}, \quad (4)$$

where  $M_\epsilon$  = the minimum number of  $\epsilon$ -by- $\epsilon$  boxes needed to cover  $S$ . (By covering  $S$ , we mean finding a set of square boxes  $B_i$  such that  $\cup B_i \supseteq S$ .) For small values of  $\epsilon$ , the definition in Equation 4 is equivalent to writing

$$M_\epsilon \approx K \epsilon^{-\dim(S)}, \quad (5)$$

where  $K$  is a constant. This equation expresses one of the important ideas behind fractal analysis: fractal

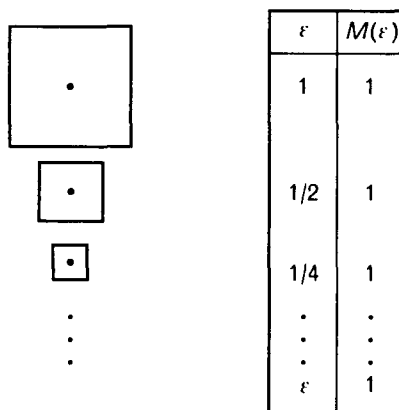
dimension measures how certain properties of a set change with the scale of observation  $\epsilon$ . In the following paragraphs, three specific examples clarify this idea.

*Example 1.* Let  $S$  be a single point. A point can be covered by one box regardless of the box size  $\epsilon$ ; hence

$$\dim(\text{point}) = \lim_{\epsilon \rightarrow 0} \frac{\log M_\epsilon}{\log \left( \frac{1}{\epsilon} \right)} = \lim_{\epsilon \rightarrow 0} \frac{\log 1}{\log \left( \frac{1}{\epsilon} \right)} = 0.$$

We could use Equation 5 to derive this same result by noting that the number of square boxes needed to cover  $S$  is independent of the box size; thus  $\dim(\text{point})$  equals zero. Figure 7 summarizes this example. In addition, as long as  $\epsilon$  is below a certain critical value, a finite set of isolated points can be covered by a fixed number of boxes (independent of  $\epsilon$ ). Therefore, a finite set of isolated points also has a dimension of zero.

*Example 2.* Let  $S$  be a line segment. For simplicity, we assume the line is 1 unit long. A single 1-unit-by-1-unit box can cover the line. If we reduce the box size to 1/2 unit by 1/2 unit, then two boxes are



$$\text{Dim}(\text{point}) = \lim_{\epsilon \rightarrow 0} \frac{\log(1)}{\log(1/\epsilon)} = 0.$$

**FIGURE 7.** Fractal-dimension calculation for a point. As the size of the square box that covers the point decreases, the number of boxes required to cover the point remains the same (i.e., one). As a result, the fractal dimension of a point is zero.

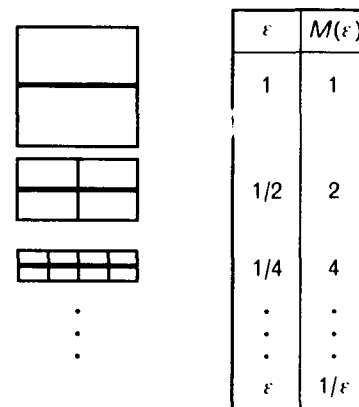
needed to cover the line. If we reduce the box size again, to 1/4 unit by 1/4 unit, four boxes are needed to cover the line. Each time the box size is halved, the number of boxes needed to cover a line doubles; thus

$$\dim(\text{line}) = \lim_{\epsilon \rightarrow 0} \frac{\log 2^n}{\log \left( \frac{1}{\epsilon} \right)} = 1.$$

Figure 8 summarizes this example. It can also be shown that a finite set of isolated line segments has a fractal dimension of one.

*Example 3.* Let  $S$  be a square area. Again, for simplicity, we assume the square is 1 unit by 1 unit in size. A single 1-unit-by-1-unit box can cover the square. If we reduce the box size to 1/2 unit by 1/2 unit, four boxes are required. If we reduce the box size again, to 1/4 unit by 1/4 unit, 16 boxes are required. As the box size is halved, the number of boxes needed to cover the square area quadruples; thus

$$\dim(\text{square}) = \lim_{\epsilon \rightarrow 0} \frac{\log 2^{2n}}{\log \left( \frac{1}{\epsilon} \right)} = 2.$$



$$\text{Dim}(\text{line}) = \lim_{\epsilon \rightarrow 0} \frac{\log(1/\epsilon)}{\log(1/\epsilon)} = 1.$$

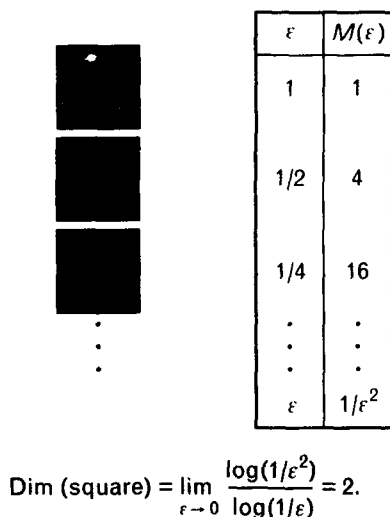
**FIGURE 8.** Fractal-dimension calculation for a line segment. As the size of each square box that covers the line segment is halved, the number of boxes required to cover the line segment doubles. As a result, the fractal dimension of a line segment is one.

Figure 9 summarizes this example. We used a square area in this example for convenience. Any area that can be synthesized from a finite number of square areas, however, will have a fractal dimension of two.

From these simple examples, we see that fractal dimension clearly has the potential to discriminate between certain types of objects in 2-D space. The question is, how can this feature be applied to SAR data?

The first step in applying the fractal-dimension concept to a radar image is to select an appropriately sized region of interest, and then convert the pixel values in the region of interest to binary (i.e., each pixel value equals 0 or 1). One method of performing this conversion is *amplitude thresholding*, in which all pixel values exceeding a specified threshold are converted to 1, and the remaining pixel values are converted to 0. Another method is to select the  $N$  brightest pixels in the region of interest and convert their values to 1, while converting the rest of the pixel values to 0; this second method is the approach we used (because it worked better).

After converting the radar image to a binary image,



**FIGURE 9.** Fractal-dimension calculation for a square area. Each time the size of each square box that covers the square area is halved, the number of boxes required to cover the square quadruples. As a result, the fractal dimension of a square is two.

we let the pixels with the value 1 constitute the set  $S$  in Equation 4. A problem arises, however, when we attempt to apply the definition of Equation 4 directly to our binary image. According to the definition of fractal dimension, we need to take a limit as the box size  $\epsilon$  goes to zero. The smallest meaningful value of the box size  $\epsilon$ , however, is the size of one pixel. Therefore, we must develop an approximation to the formula of Equation 4.

From Equation 5 we observe that

$$\log M_\epsilon \approx -\text{dim} \log \epsilon + \log K$$

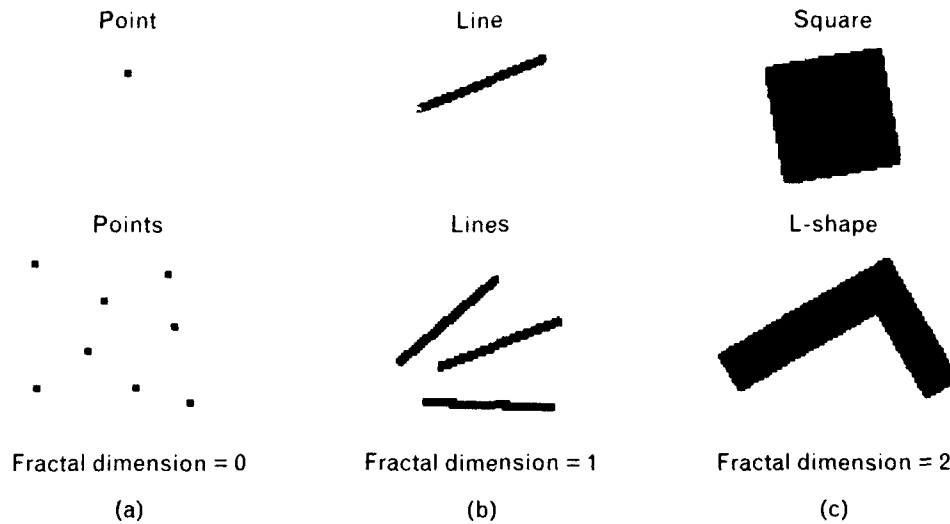
for small  $\epsilon$ . Because the relation between  $\log M_\epsilon$  and  $\log \epsilon$  is linear for small  $\epsilon$ , with the slope equal to the negative of the dimension, the fractal dimension can be approximated by using only the box counts for  $\epsilon = 1$  and  $\epsilon = 2$  in the following way:

$$\text{dim} = -\frac{\log M_1 - \log M_2}{\log 1 - \log 2} = \frac{\log M_1 - \log M_2}{\log 2}, \quad (6)$$

where  $M_1$  is the number of 1-pixel-by-1-pixel boxes needed to cover the image and  $M_2$  is the number of 2-pixel-by-2-pixel boxes needed to cover the image. Figure 10 summarizes the fractal dimensions of simple objects as they are observed in SAR imagery.

The following paragraphs provide two examples of calculating the fractal dimension of regions of interest in radar imagery. The examples use data extracted from the SAR image shown in Figure 11. The figure shows a Stockbridge, New York, clutter scene that includes trees, a street with houses on both sides, a swimming pool, and a meadow. The examples demonstrate the fractal-dimension calculation for a typical tree (natural clutter) and the rooftop of a house (cultural clutter).

Figure 12 illustrates the fractal-dimension calculation for a binary image of a tree; the binary image was formed by selecting the 50 brightest pixels from a 120-ft-by-120-ft region of interest in the image of Figure 11. The number of 1-pixel-by-1-pixel boxes needed to cover this image is identical to the number of pixels with the value 1 (i.e.,  $M_1$  equals 50). The



**FIGURE 10.** Fractal dimensions of simple objects in SAR imagery. (a) Points have a fractal dimension of zero, (b) lines have a fractal dimension of one, and (c) squares and L-shaped objects have a fractal dimension of two. In the text we show how to calculate the fractal dimensions of these objects by using the approximation derived in Equation 6.

minimum number of 2-pixel-by-2-pixel boxes needed to cover the image is 41; Figure 12 shows this minimal covering. By applying Equation 6, we find that the fractal dimension of the tree is 0.29. This relatively low value reflects the fact that the binary image of the tree consists primarily of isolated pixels.

Figure 13 illustrates the fractal-dimension calculation for a binary image of a house rooftop (this image was formed in the same way as the image of the tree in Figure 12). Notice that the pixels in this image are clustered into lines and areas. The number of 1-pixel-by-1-pixel boxes needed to cover the image is 50, but the minimum number of 2-pixel-by-2-pixel boxes needed to cover the image is only 21. By using Equation 6, we find that the fractal dimension of the house rooftop is 1.25. This relatively high value is caused by the clustering of the pixels. The different fractal-dimension values for the tree and the rooftop illustrate that this feature can be used to discriminate between natural clutter and cultural clutter.

#### *Weighted-Rank Fill Ratio Feature*

The third textural feature, the weighted-rank fill ratio, measures the percentage of the total energy con-

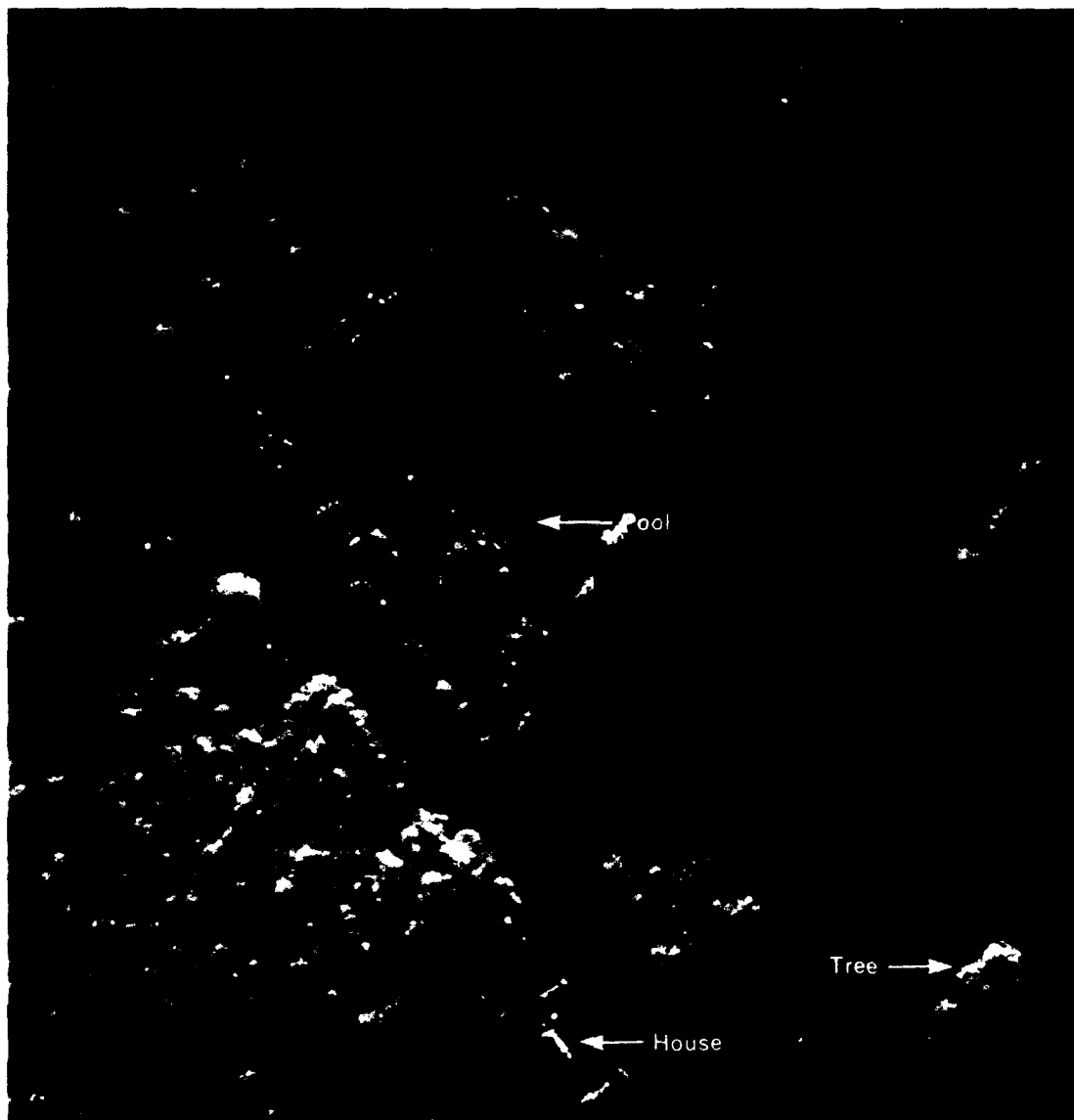
tained in the brightest scatterers of a detected object. Using the notation of Equations 2 and 3, we define the weighted-rank fill ratio  $\eta$  as follows:

$$\eta = \frac{\sum_{k \text{ brightest pixels}} P(r, a)}{\sum_{\text{all pixels}} P(r, a)},$$

where  $k$  is selected to correspond approximately to the brightest 5% of the detected object's pixels. For man-made objects a significant portion of the total energy comes from a small number of bright scatterers; for natural clutter the total energy is distributed more evenly among the pixels.

#### **Future ATR System Improvements**

The baseline ATR system currently uses only three features in the discrimination stage (standard deviation, fractal dimension, and weighted-rank fill ratio); we have found that these features reliably reject natural-clutter false alarms. Other discrimination features could be added that would also reject some cultural-clutter false alarms. For example, a size feature, such as length and width of the detected object, could

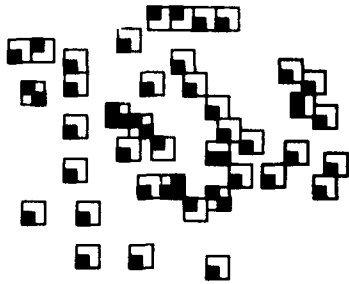


**FIGURE 11.** High-resolution (1 ft by 1 ft) PWF-processed SAR image of mixed natural and cultural clutter in Stockbridge, New York. A tree and a house (bottom from this image are used in the feature extraction calculations illustrated in Figures 12 and 13).

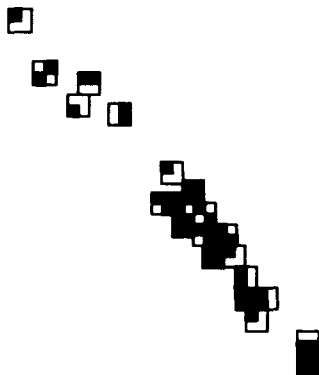
reject objects too large or too small to be targets. Numerous discrimination features have been investigated, and an improved set of five features has been found to provide better discrimination performance than the baseline feature set used in this article. A detailed description of all the discrimination features investigated so far is given in the accompanying article entitled "Discriminating Targets from Clutter" by Daniel E. Kriethen et al.

Improvements in the baseline pattern matching classifier will be necessary before an operational, end-

to-end ATR system can be implemented. Target signatures vary with aspect angle, depression angle, and SAR squint angle; constructing templates incorporating these variables would produce a large and computationally unwieldy set of templates. In an operational scenario, the target aspect angle is not known. Therefore, each detected object would have to be correlated with a large subset of the pattern matching templates, which would increase computation time significantly. The accompanying article entitled "Improving a Template-Based Classifier in a SAR Automatic Target Rec-



**FIGURE 12.** Fractal-dimension calculation for the binary image of a tree. The 50 brightest pixels (indicated by the small black boxes) are relatively isolated, and 41 two-pixel-by-two-pixel boxes are needed to cover them, which results in a low fractal dimension of 0.29.



**FIGURE 13.** Fractal-dimension calculation for the binary image of a house rooftop. The 50 brightest pixels (indicated by the small black boxes) are more tightly clustered than they are for the tree in Figure 12, and only 21 two-pixel-by-two-pixel boxes are needed to cover them, which results in a higher fractal dimension of 1.25.

ognition System by Using 3-D Target Information" by Shawn M. Verbout et al. presents an approach to this classification task based on the generation of 2-D templates from 3-D models of targets.

Because the pattern-matching approach to classification requires a large number of templates, we are investigating an alternative approach to classification—

using spatial matched filters. The initial results of these studies indicate the possibility of a significant reduction in computation time and storage requirements with no reduction in performance [7].

### Acknowledgments

The authors wish to acknowledge the significant contributions made by Michael C. Burl, particularly to the development of the fractal-dimension feature. The authors also wish to thank Steven M. Auerbach for his help in technical editing and Robin Fedorchuk for her outstanding typing efforts. This work was sponsored by the Advanced Research Projects Agency.

## REFERENCES

1. M.C. Burl, G.J. Owirka, and L.M. Novak, "Texture Discrimination in Synthetic Aperture Radar," *23rd Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, 30 Oct. - 1 Nov. 1989*, p. 399.
2. J.C. Henry, "The Lincoln Laboratory 35 GHz Airborne Polarimetric SAR Imaging System," *IEEE Natl. Teleystems Conf., Atlanta, GA, 26-27 Mar. 1991*, p. 353.
3. L.M. Novak, M.C. Burl, R. Chaney, and G.J. Owirka, "Optimal Processing of Polarimetric Synthetic Aperture Radar Imagery," *Proc. Lab. J.*, **3**, 273 (1990).
4. L.M. Novak, M.C. Burl, and W.W. Irving, "Optimal Polarimetric Processing for Enhanced Target Detection," *IEEE Trans. Aerosp. Electron. Syst.*, **29**, 234 (1993).
5. G.B. Goldstein, "False-Alarm Regulation in Log Normal and Weibull Clutter," *IEEE Trans. Aerosp. Electron. Syst.*, **8**, 84 (1973).
6. W.W. Irving, G.J. Owirka, and L.M. Novak, "A New Model for High-Resolution Polarimetric SAR Clutter Data," *SPIE*, **1630**, 208 (1992).
7. L.M. Novak, G.J. Owirka, and C.M. Netishen, "Radar Target Identification Using Spatial Matched Filters," *3rd ATR Systems and Technology Conf.*, June 1993.



**LESLIE M. NOVAK**

is a senior staff member in the Surveillance Systems group. He received a B.S.E.E. degree from Fairleigh Dickinson University in 1961, an M.S.E.E. degree from the University of Southern California in 1963, and a Ph.D. degree in electrical engineering from the University of California, Los Angeles, in 1971. Since 1977 Les has been a member of the technical staff at Lincoln Laboratory, where he has studied the detection, discrimination, and classification of radar targets. He has contributed chapters on stochastic observer theory to the series *Advances in Control Theory*, edited by C.T. Leondes (Academic Press, New York), volumes 9 and 12.



**GREGORY J. OWIRKA**

is an assistant staff member in the Surveillance Systems group. He received a B.S. degree (cum laude) in applied mathematics from Southeastern Massachusetts University, and he is currently working on an M.S. degree in electrical engineering at Northeastern University. Greg's current research interests are in automatic target recognition. He has been at Lincoln Laboratory since 1987.



**CHRISTINE M. NETISHEN**

is an assistant staff member in the Surveillance Systems group. Her research speciality is in the detection, discrimination, and classification of stationary ground vehicles in SAR imagery. She received a B.S. degree in mathematics (cum laude) from Providence College; she also spent a year at Cambridge University in England studying mathematics and physics. Christine has been at Lincoln Laboratory since 1991.

# Discriminating Targets from Clutter

Daniel E. Kreithen, Shawn D. Halversen, and Gregory J. Owirka

■ The Lincoln Laboratory multistage target-detection algorithm for synthetic-aperture radar (SAR) imagery can be separated into three stages: the prescreener, the discriminator, and the classifier. In this article, we focus on the discrimination algorithm, which is a one-class, feature-based quadratic discriminator. An important element of the algorithm design is the choice of features. We examine fifteen features that are used in the discrimination algorithm—three features developed by Lincoln Laboratory, nine developed by the Environmental Research Institute of Michigan, two developed by Rockwell International Corporation, and one developed by Loral Defense Systems. The set of best features from this pool of fifteen was determined by a theoretical analysis, and was then verified by using real SAR data. Performance was evaluated for a number of different cases: for fully polarimetric data and HH polarization data and for 1-ft resolution data and 1-m resolution data. In all cases the theoretical performance analysis closely matched the real data performance. This closeness demonstrates a good understanding of the discrimination algorithm. In addition, we formulate a set of criteria for best feature choice that apply to quadratic discrimination algorithms in general.

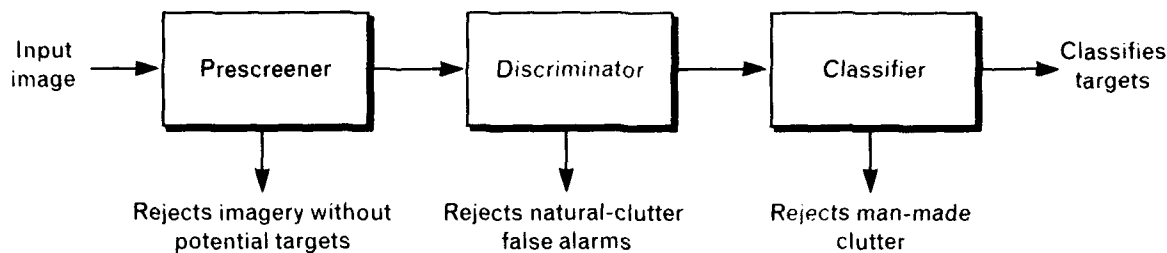
LINCOLN LABORATORY HAS PROCURED a fully polarimetric, instrumentation-quality, high-resolution (1 ft by 1 ft), 35-GHz, millimeter-wave (MMW) synthetic-aperture radar (SAR), which has been used to gather imagery of targets of interest and clutter in a number of different locations and deployments. The radar, which is mounted in a Gulfstream G-1 aircraft, records data in-flight onto 24-track magnetic tapes. The tapes are then processed on the ground to form the SAR imagery. A recent *Lincoln Laboratory Journal* article by Leslie M. Novak et al. describes this radar system [1].

The Surveillance Systems group at Lincoln Laboratory has been developing algorithms to detect targets of interest in this SAR imagery. A block diagram of the algorithm suite is shown in Figure 1. The target-detection algorithm suite takes the form of a multistage algorithm. In theory, it is possible to construct a single algorithm that performs target detection in an optimal manner, and which exploits all of

the information present in a high-resolution SAR image. Unfortunately, it is often difficult to design algorithms using the single-algorithm approach, because high-resolution SAR imagery is difficult to model accurately and hence is poorly understood. The multistage approach becomes an attractive alternative, because of the reduction in required computational capability and the simplification in algorithm design.

The Lincoln Laboratory multistage algorithm has three separate stages, each of which performs easily identifiable functions. The first stage, which is called the *prescreener*, is a computationally simple algorithm whose function is to pass all targets and eliminate only obviously non-targetlike naturally occurring clutter. The second stage, called the *discriminator*, ideally eliminates all naturally occurring clutter that has been passed by the prescreener, and passes only man-made objects to the third stage, which is called the *classifier*. The classifier receives all man-made objects that have been passed by the discriminator and categorizes each





**FIGURE 1.** Block diagram of the multistage target-detection algorithm. This article concentrates on the discriminator stage.

one either as a target of interest (of which there can be a number of classes) or as an uninteresting man-made object.

In this article, we concentrate our attention on the second stage, the discriminator. The prescreener stage is covered elsewhere [2]; the classification stage is still under development.

### Algorithm Description

The discrimination algorithm used in the Lincoln Laboratory automatic target-detection algorithm suite is centered around a one-class quadratic discriminator [3–5]. A one-class discriminator is trained only on a target-training set, and it assumes that the clutter false-alarm dataset (i.e., the set of false alarms passed by the prescreener stage) has unknown attributes in a feature space. Figure 2 illustrates the concept of this discrimination algorithm. For each region of interest, the algorithm produces a score that measures the distance from the candidate to the center of the target-training set (in a feature space). When the algorithm is properly trained, a lower value of this distance metric indicates a more targetlike candidate.

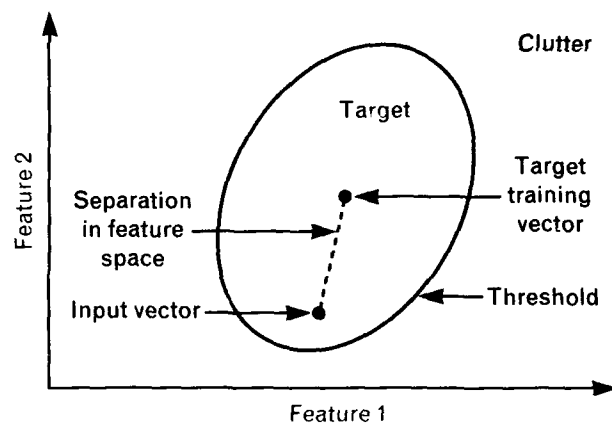
Key elements of the discrimination algorithm are the features used to compute the distance metric. We cover the discrimination features used in the Lincoln Laboratory target-detection algorithm suite in the next five sections of this article. Subsequent sections cover the discrimination algorithm itself in great detail.

### Discrimination Features

A number of attributes that are present in the fully polarimetric, high-resolution SAR imagery can be exploited to discriminate between targets and clutter

false alarms. These attributes include size, shape, power, polarimetric properties, spatial distribution of reflected power, and dimensionality. Unfortunately, at present *no method exists for developing discrimination features to exploit these attributes in any optimal fashion*. The best that can be done is to design a feature that seems to exploit a specific attribute, and then test the feature on a variety of data to see if it separates targets from natural-clutter false alarms. If it does not separate targets from false alarms, the feature design is obviously poor; if it does separate them, then the feature *may* be a good one.

The other major criterion that a feature must satisfy is *orthogonality*. In simple terms, features used



**FIGURE 2.** Conceptual diagram of a one-class discrimination algorithm. This diagram represents a two-dimensional feature space. If the separation in feature space is less than the threshold, then the region of interest from which the input vector is extracted is declared to be a target. Conversely, if the separation in feature space is greater than the threshold, the region of interest is declared to be clutter.

together in a discrimination algorithm must measure different attributes of the region of interest. For example, five different features that measure similar polarimetric properties of the candidate region of interest should not be used together in the same discrimination algorithm. In fact, using similar features is likely to make discrimination performance worse. This fact is a consequence of the likely occurrence that the target-training dataset will differ somewhat from the target-testing dataset. This phenomenon is covered more fully in the sidebar entitled "Adding Features Can Degrade Performance."

Another desirable property of the features in a discrimination algorithm is that they should be robust in a number of ways. Many feature algorithms require thresholds to be set to isolate the brightest scatterers, for example, or to isolate the scatterers with the most contrast. The feature values should not be too sensitive to the settings of these thresholds, because they may then work in one deployment situation but not in another similar situation. The features should also be somewhat robust to countermeasures; radar signatures of military vehicles are frequently altered by any number of methods. Some common methods include placing foliage and mud on the vehicle, adding metal parts to the vehicle, deploying camouflage netting around and on top of the vehicle, coating the target with radar-absorbing material, or simply opening the hatches of an armored target. An effective discrimination feature would ideally be insensitive to these methods and to other types of countermeasures.

We examined fifteen features for use in the Lincoln Laboratory target discrimination algorithm; three of these features were developed at Lincoln Laboratory [6], nine were developed at the Environmental Research Institute of Michigan (ERIM) of Ann Arbor, Michigan, two were developed by Rockwell International Corporation of El Segundo, California, and one was developed by Loral Defense Systems of Goodyear, Arizona. The non-Lincoln Laboratory features were developed under the Strategic Target Algorithm Research (STAR) contract, a yearlong research contract funded jointly by the Advanced Research Projects Agency (ARPA) and the United States Air Force. This contract was administered by Lincoln

Laboratory; the goal was to develop and test target detection algorithms by using a common dataset provided by Lincoln Laboratory. Each contractor's approach to the target-detection problem was somewhat different, but all used a number of features. We chose the most promising features for evaluation in this study.

#### *Lincoln Laboratory Discrimination Features*

The three Lincoln Laboratory discrimination features are standard deviation, fractal dimension, and weighted-rank fill ratio. They were developed by Leslie Novak, Michael Burl, and Gregory Owirka, all of Lincoln Laboratory. The features are computed from target-sized areas that are centered over the pixels identified by the prescanner for further processing. The extent of the target-sized area is determined by the *a priori* knowledge of what type of target is being sought, and a box-spinning algorithm is used to determine target orientation [7].

The standard-deviation feature is computed from the typical estimator for the standard deviation. It uses the power (expressed in dB) of all the pixels in a target-sized box.

The fractal-dimension feature, which is illustrated in Figure 3, provides a measure of the spatial dimensionality of the potential target [6]. This feature estimates the Hausdorff dimension of the spatial distribution of the top  $N$  scatterers in the region of interest. For example, a straight line has a Hausdorff dimension of one, and a solid rectangle has a Hausdorff dimension of two. Various other space-filling objects with holes have a Hausdorff dimension that falls between one and two. An isolated point has a Hausdorff dimension of zero.

To compute the fractal-dimension feature, we threshold the region of interest by taking only the top  $N$  scatterers in terms of power. A binary image is created from these scatterers, and the minimum number  $n_1$  of 1-pixel-by-1-pixel boxes ( $d_1 = 1$ ) that cover all  $N$  scatterers is determined. This number, of course, is equal to the value  $N$ . Then the minimum number  $n_2$  of 2-pixel-by-2-pixel boxes ( $d_2 = 2$ ) that cover all  $N$  scatterers is determined. This number is less than or equal to  $N$ . If the spatial distribution of the scatterers is highly diffuse, the value of  $n_2$  will be close to  $N$ ; if

## ADDING FEATURES CAN DEGRADE PERFORMANCE

IN THE MAIN ARTICLE, a theoretical analysis of the one-class quadratic discriminator shows theoretical expressions for the probability of detection ( $P_d$ ) and the probability of false alarm ( $P_{fa}$ ) of the algorithm. We make the claim in the section on discrimination features that adding features does not necessarily improve discrimination performance. We show here that this is indeed the case by giving two examples; the first example shows discrimination performance with the set of five best features, and the second example shows discrimination performance with the same set of features in addition to seven other features.

The idea that adding features can degrade performance is, perhaps, counterintuitive. In fact, we cannot degrade performance by adding features if a few key conditions are met. These conditions are (1) the real data obey perfectly the multivariate Gaussian assumption made in the section entitled "Theoretical Analysis of the One-Class Quadratic Discrimination Algorithm," and (2) the target-training data and the target-testing data have exactly the same statistical distribution.

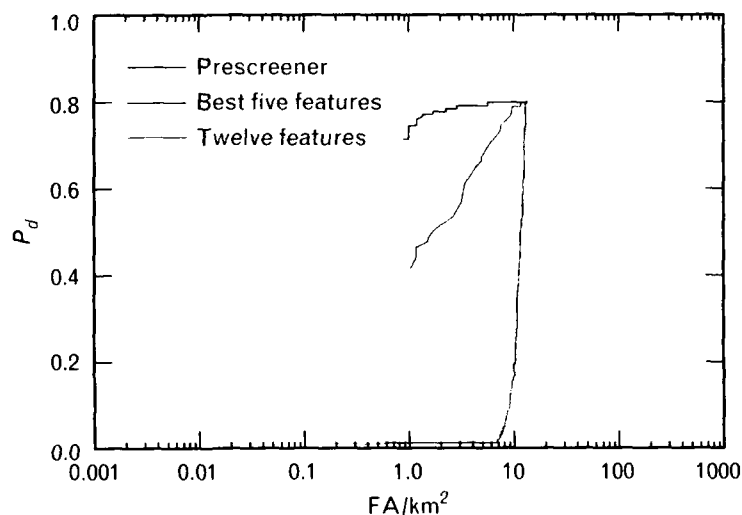
The multivariate Gaussian condition is difficult to verify for the real data, especially when a number of features are being consid-

ered (see the section entitled "Confirming the Gaussian Assumption"). Small departures from Gaussianity, however, are probably not the major cause of the phenomenon we are addressing here. Instead, the major cause of degraded performance is that the second condition is not being met.

Figure A shows performance curves for the 1-ft, polarimetric whitening filter (PWF) data using the features described in the section entitled "Best Features for Discrimination." This figure, which is the same diagram as Figure 17, also shows performance curves for the same dataset using

all twelve Lincoln Laboratory and ERIM STAR discrimination features. Notice the significant performance degradation that occurs when the seven extra features are added. Notes on interpreting this type of graph can be found in the sidebar entitled "Interpreting Plots of  $P_d$  versus  $FA/km^2$ ."

Figure B is a notional diagram that illustrates the degradation phenomenon. The diagram is complicated but the explanation of it is relatively easy. There are three distinct sets of data displayed in the diagram: target training, target testing, and clutter false alarm. Each dataset is displayed for two features, which we call



**FIGURE A.** Performance curves comparing discrimination performance for the five best features and for all twelve features. Performance degrades when more features are added.

Feature 1 and Feature 2. Imagine that the target-training data and the target-testing data are the same (training and testing is done on the same data, so ignore the red points on the diagram for the time being).

First, assume that the discrimination algorithm uses only Feature 1 (refer to only the green and black points plotted along the abscissa for this case). We see that the discriminator does a good job of separating targets from clutter by using Threshold A (three false alarms are called targets). In this case the two key conditions are satisfied: the target-training data and target-testing data have ex-

actly the same statistical distribution, and the data obey (more or less) the Gaussian assumption (within each data type).

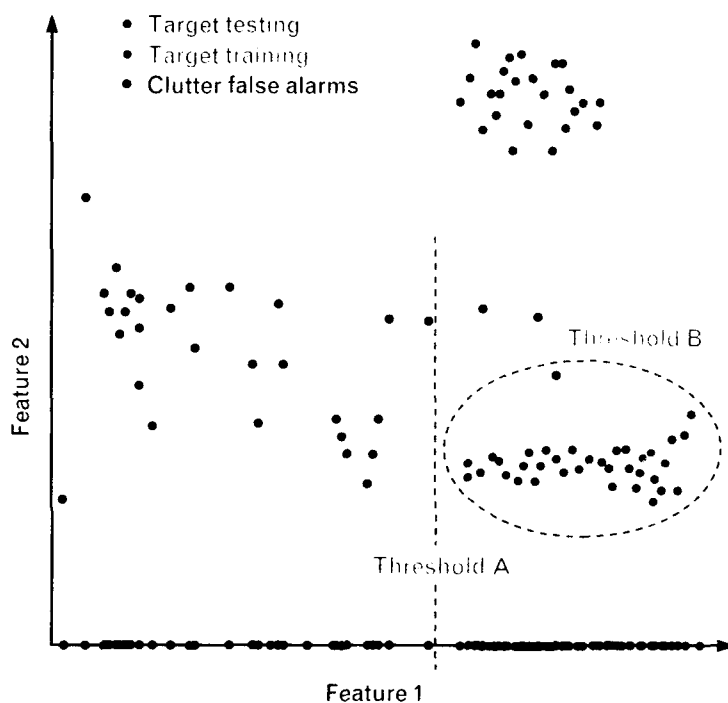
Now, while still assuming the target-training data and target-testing data are the same, we add Feature 2 (refer to only the green and black points in the middle of the graph). We can then draw Threshold B as an ellipse around the data, and do a still better job at separating targets from clutter (in this case one false alarm is called a target). The two key criteria are still obeyed.

If we assume now that the target-training data and the target-testing data are different datasets,

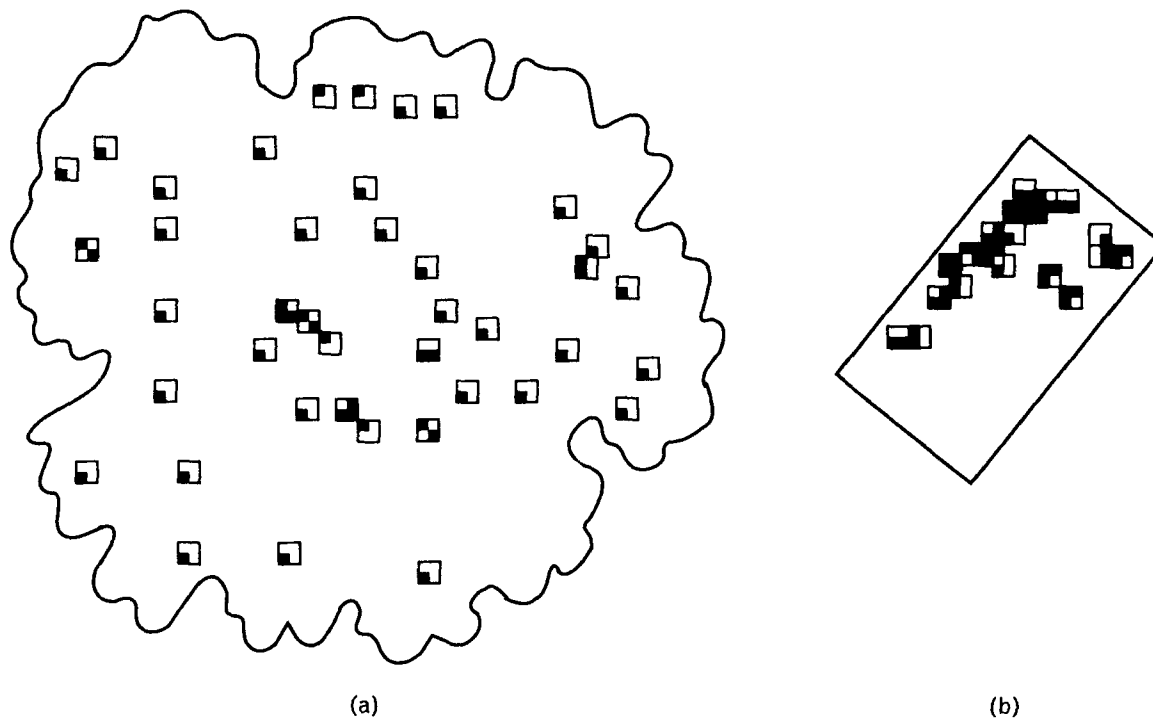
we remove the second key condition mentioned above. First, we assume that the discrimination algorithm uses only Feature 1 (refer to the green, red, and black points plotted along the abscissa). By drawing a threshold at the point labeled Threshold A, we see that the discrimination performance is equivalent to the previous case because the target-training data and the target-testing data for Feature 1 have similar characteristics.

Now, we add Feature 2. Performance is severely degraded because we cannot draw an ellipse (with the same orientation as Threshold B) around the center of mass of the green points that does not engulf large numbers of clutter false alarms (black points) while still engulfing the target-testing data (red points).

The important point is that any threshold ellipse must have the same orientation as the ellipse shown as Threshold B, because the orientation of the ellipse is determined by the statistical characteristics of the target-training data. This example is a particularly egregious illustration of the failure of the second key condition, because the target-training data and the target-testing data now have different statistical characteristics. The addition of Feature 2 obviously degrades performance severely. More subtle cases that significantly affect performance occur more frequently.



**FIGURE B.** Notional diagram of reason for performance degradation when features are added. This graph shows two features in a feature space.



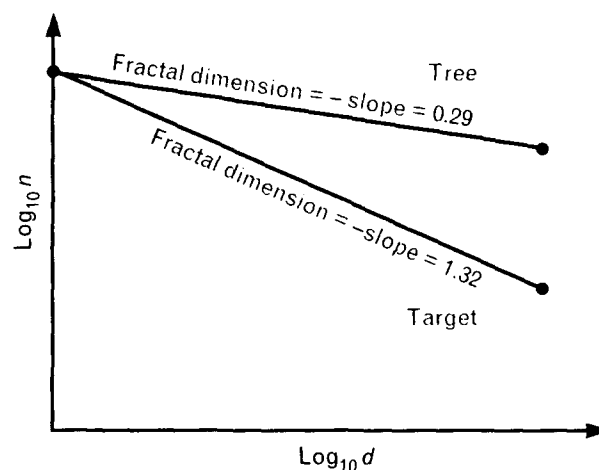
**FIGURE 3.** Calculation of the fractal-dimension feature, which measures the spatial bunching of the brightest pixels in a region of interest. (a) The brightest pixels for a tree tend to be widely separated, which requires a relatively large number of covering boxes and produces a low value for the fractal dimension. (b) The brightest pixels for a target tend to be closely bunched, which requires fewer covering boxes and produces a high value for the fractal dimension.

the scatterers are spatially bunched the value of  $n_2$  will be considerably less than  $N$ . These values are determined for the two specific examples in Figure 3 and plotted in Figure 4, with the logarithm of  $n_1$  and  $n_2$  on the ordinate, and the logarithm of  $d_1$  and  $d_2$  on the abscissa. The negative slope of the line through the two points, which is given by

$$H_d = \frac{\log n_1 - \log n_2}{\log d_2 - \log d_1}, \quad (1)$$

is an estimate of the Hausdorff dimension of the region of interest. For the high-resolution data in this article, we used  $N = 50$ .

The weighted-rank fill-ratio feature is computed from the top  $N$  scatterers in the target-sized box. The feature is computed by totaling the power in the top  $N$  pixels within the target-sized box, and normalizing by the total power of all pixels in the box. This feature attempts to exploit the fact that power returns from



**FIGURE 4.** An estimate of the Hausdorff dimension of the tree and target in Figure 3. For both objects the total number of scatterers is 50; for the tree the minimum number of covering boxes is 41 and for the target the minimum number of covering boxes is 20. The negative slope of the line for each object is the estimate of the fractal-dimension feature. Targets tend to have higher fractal dimensions than natural clutter.

most targets tend to be concentrated in a few bright scatterers, whereas power returns from natural-clutter false alarms tend to be more diffuse. This feature measures a power-related property of the target-sized box, which makes this feature different from the fractal-dimension feature, which measures a spatial property of the entire region of interest.

### ERIM Discrimination Features

The ERIM discrimination features were developed and provided to Lincoln Laboratory under the STAR contract mentioned above. They were modified for this study by altering the thresholds to account for a target dataset that was substantially different from the dataset used for the STAR contract. Instead of using a target-sized box as a preliminary step, as in the Lincoln Laboratory feature algorithms, the ERIM feature algorithms compute a target-shaped blob by performing morphological operations. These operations serve both as a method of grouping spatially related hits from the prescanner and as a method of estimating the size, shape, and orientation of the supposed target.

There are three categories of ERIM discrimination features: size-related features, contrast-based features, and polarimetric features. Each of these three categories contains three features. The size-related features are mass, diameter, and square-normalized rotational inertia. The contrast-based features are maximum constant false-alarm rate (CFAR) statistic, mean CFAR statistic, and percent bright CFAR statistic. The polarimetric features are percent pure, percent pure even, and percent bright even. We describe each feature in detail in the following paragraphs.

The three size-related features utilize only the binary image created by the morphological operations. The mass feature is computed by counting the number of pixels in the morphological blob. The diameter is the length of the diagonal of the smallest rectangle (either horizontally oriented or vertically oriented) that encloses the blob. The square-normalized rotational inertia is the second mechanical moment of the blob around its center of mass, normalized by the inertia of an equal mass square.

The contrast-based features are determined by a CFAR algorithm. This algorithm can be described by

$$\frac{x - \hat{\mu}_c}{\hat{\sigma}_c}, \quad (2)$$

where  $x$  represents the test pixel, and  $\hat{\mu}_c$  and  $\hat{\sigma}_c$  are estimates of the local mean and local standard deviation, respectively, of the surrounding clutter. The estimates of the parameters from the surrounding clutter are accomplished by using the pixels in a window around the supposed target whose opening is large enough to exclude the target return. Figure 5 illustrates this window, and the opening is called the *guard area*.

The CFAR statistic given by Equation 2 is computed for each pixel to create a CFAR image. The maximum CFAR feature is the maximum value in the CFAR image contained within the target-shaped blob. This quantity is similar to the basic feature used in the prescanner algorithm. The mean CFAR feature is the average of the CFAR image taken over the target-shaped blob. The percent bright CFAR feature is the percentage of pixels within the target-shaped blob that exceed a certain CFAR value.

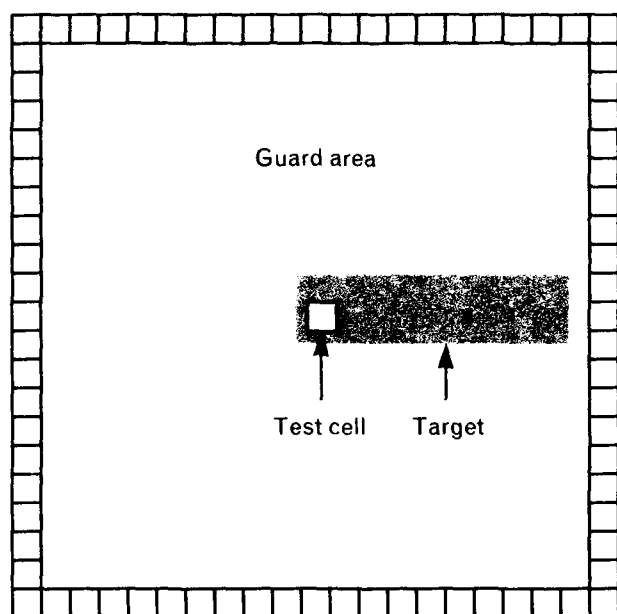
The polarimetric discrimination features are based on a transformation of the linear polarization basis in which the Lincoln Laboratory MMW SAR gathers data to an even-bounce, odd-bounce basis described by the equations

$$\begin{aligned} E_{\text{odd}} &= \frac{|HH + VV|^2}{2} \\ &= 2|RL|^2 \end{aligned}$$

and

$$\begin{aligned} E_{\text{even}} &= \frac{|HH - VV|^2}{2} + 2|HV|^2 \\ &= |RR|^2 + |LL|^2. \end{aligned}$$

The odd-bounce channel given by the first equation corresponds to the radar return from a flat plate or a trihedral; the even-bounce channel corresponds to the radar return from a dihedral. Figure 6 illustrates examples of these reflectors, along with notional diagrams of how they reflect the radar energy. The use-

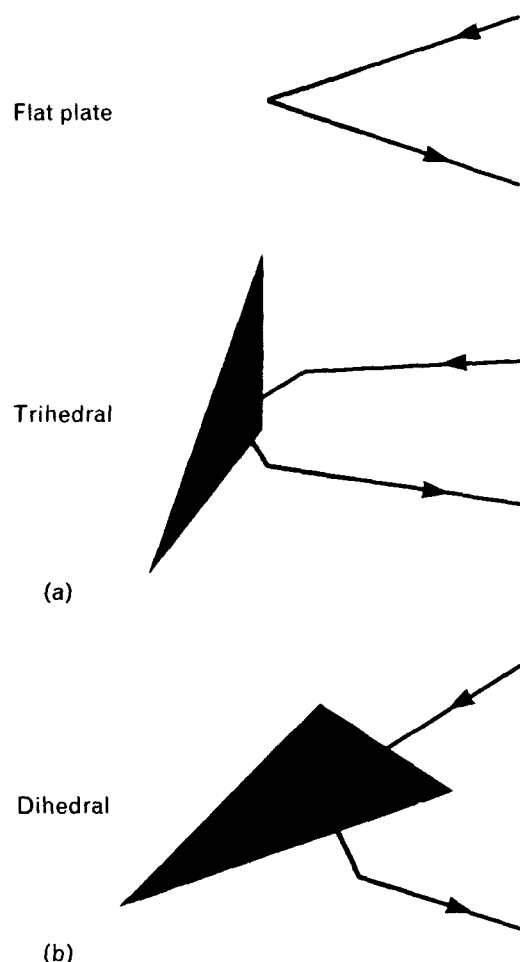


**FIGURE 5.** CFAR template, showing the pixel under test, and the surrounding window of pixels from which clutter estimates are computed. The test pixel and clutter window are separated by a guard area, which protects the clutter estimates from being corrupted by portions of the target return.

fulness of these polarimetric feature resides in the fact that few dihedral structures exist in natural clutter, but these structures are plentiful on most man-made targets. Natural clutter tends to exhibit more odd-bounce reflected energy than even-bounce reflected energy.

The ERIM polarimetric features are formed from the even-bounce and the odd-bounce images. The percent-pure feature is the fraction of pixels within the target-shaped blob for which at least a certain fraction of the scattered energy falls in either the even-bounce channel or the odd-bounce channel. Percent even is the fraction of pixels within the target-shaped blob for which at least a certain fraction of the scattered energy falls in the even-bounce channel. The percent-bright-even feature is the fraction of pixels that exceed a certain value in the CFAR image described above, and which are mainly even-bounce scatterers.

The main impetus for these features is that a man-made object exhibits approximately equal amounts of pure even-bounce energy and odd-bounce energy, whereas a natural-clutter false alarm is more likely to



**FIGURE 6.** Reflection of radar signal from a variety of reflectors. (a) Odd-bounce reflectors include a flat plate and a trihedral. (b) Even-bounce reflectors include a dihedral. Radar backscatter from natural clutter is predominantly odd bounce, while backscatter from man-made objects is typically an equal mixture of even bounce and odd bounce.

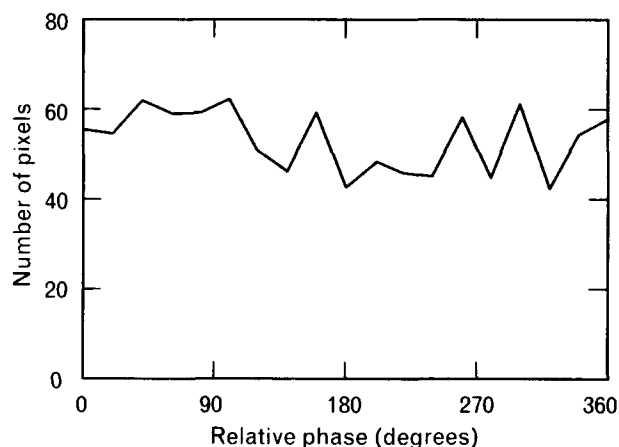
exhibit large amounts of pure odd-bounce energy. Also, man-made objects are more likely to exhibit an equal mixture of even-bounce and odd-bounce energy than a natural-clutter false alarm.

#### *Rockwell Discrimination Features*

The Rockwell discrimination features were also developed and provided to Lincoln Laboratory under the STAR contract. Like the ERIM discrimination features, they were modified to account for the different type of target data used in the present study. These

features use the pixels in a target-sized box for feature calculation; the algorithm used to determine the orientation of the target-sized box is the same as that used for computing the orientation for the Lincoln Laboratory discrimination features described earlier. Some of the Rockwell discrimination features are similar to those already used by Lincoln Laboratory and ERIM. These similar features were not considered. Instead we concentrated on two other Rockwell features: (1) the polarimetric phase ratio feature, and (2) the specific-entropy feature.

The polarimetric phase ratio feature is another attempt to exploit differences in polarization between radar returns from targets and radar returns from clutter. The relative phase between the HH polarization channel and the VV polarization channel is used for this purpose. Only pixels in a target-sized box (the orientation of which is determined by the box-spinning algorithm described in Reference 7) are examined. In addition, to eliminate the low-return pixels that may have a random phase due to corruption by receiver noise, we use only the pixels that exceed a threshold in both the HH polarization channel and the VV polarization channel. This threshold, which is set to a percentage of the maximum power in a pixel in a given image, is set to a low value so the thresholding operation eliminates only the lowest return pixels whose phase is most likely to be corrupted by receiver noise.



**FIGURE 7.** Histogram of relative phase between the HH polarization channel and the VV polarization channel. This type of plot is used in the calculation of the Rockwell polarimetric phase ratio feature.

The relative phase between the HH polarization channel and the VV polarization channel is then calculated for the remaining pixels within the target-sized box. The values are arranged in a histogram plot, such as the example shown in Figure 7, and the feature is calculated from this plot. The polarimetric phase ratio is defined as the number of pixels to fall within  $\pm x^\circ$  of  $180^\circ$  relative phase on the histogram, divided by the number of pixels that fall within  $\pm x^\circ$  of  $0^\circ$  relative phase on the histogram. We used a value of  $x = 90^\circ$ .

Specific entropy is the second Rockwell discrimination feature used in this study. Because of the complicated definition of this feature, it was not clear which step in the calculation provides the ability to separate targets from natural-clutter false alarms. To understand this feature better, we investigated it in considerable detail. A number of steps are involved in computing this feature:

1. Choose a threshold  $T$  that is set to the quantity corresponding to the 98th percentile of the surrounding clutter, and calculate a normalized amplitude by

$$a_i = \max(p_i - T, 0),$$

where  $a$  is the amplitude (in dB) above the threshold,  $p$  is the amplitude of the original pixel (in dB),  $i$  is the pixel tag number (of which there are  $m$ , which is the number of pixels in the target-sized box), and  $T$  is the value (in dB) of the threshold.

2. Normalize the amplitude by

$$f_i = \frac{a_i}{\sum_{i=1}^m a_i},$$

unless  $a_i = 0$  for all  $i = 1, \dots, m$ , in which case the specific-entropy feature is set to zero.

3. Compute the specific-entropy feature by

$$\text{specific entropy} = -\sum_{i=1}^m \frac{f_i \log f_i}{\log m}.$$

The idea behind this feature is to exploit two supposed properties of a target: (1) the pixels exceeding the threshold  $T$  do not vary greatly in amplitude for a



target, but they do vary greatly for a natural-clutter false alarm; and (2) more pixels exceed the threshold  $T$  for a target than for a natural-clutter false alarm. The question that remains is which step in the feature calculation provides the separation between targets and natural-clutter false alarms.

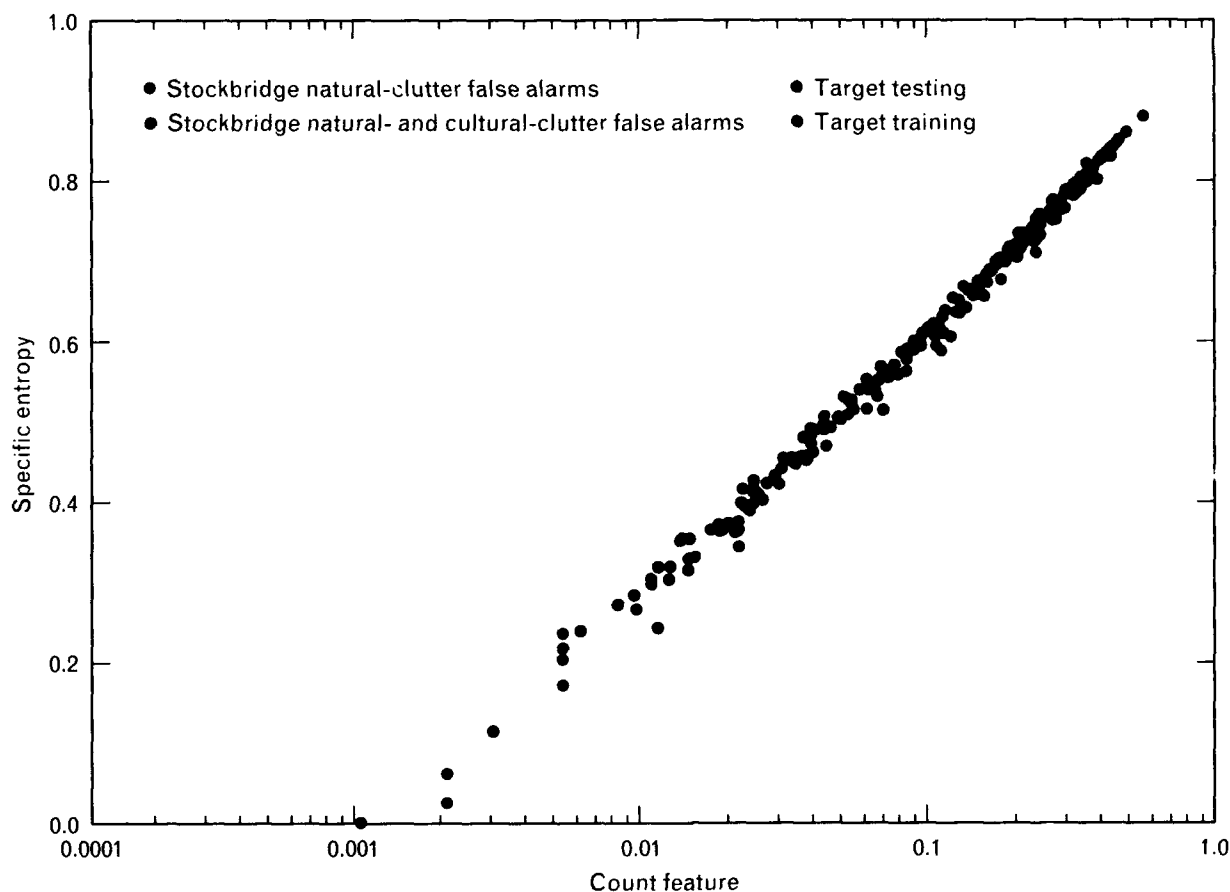
We studied this problem by separating the calculation of the specific entropy into the steps described above, and then we calculated a feature based only on the operation in each separate step. To this end, we invented a simple *count* feature, which counts the number of pixels that exceeded the threshold  $T$  as it was calculated above, and normalizes this value by the total possible number of pixels in a target-sized box. This procedure was done for targets and for clutter false alarms, and the count feature was then plotted as a scatterplot versus the specific-entropy feature, as shown in Figure 8 for a sample target dataset.

In this plot, specific entropy is plotted on the

ordinate, and the logarithm of the count feature is plotted on the abscissa. If the two features are highly correlated, the plot shows the points falling along a straight line, which is indeed the case for this example. In fact, all the target and clutter false-alarm datasets that we examined showed similar scatterplots. For practical purposes, this scatterplot indicates that the count feature and the specific-entropy feature are equivalent. The extra steps given above in the calculation of the specific-entropy feature do little to increase the separation between targets and natural-clutter false alarms.

#### *Loral Discrimination Feature*

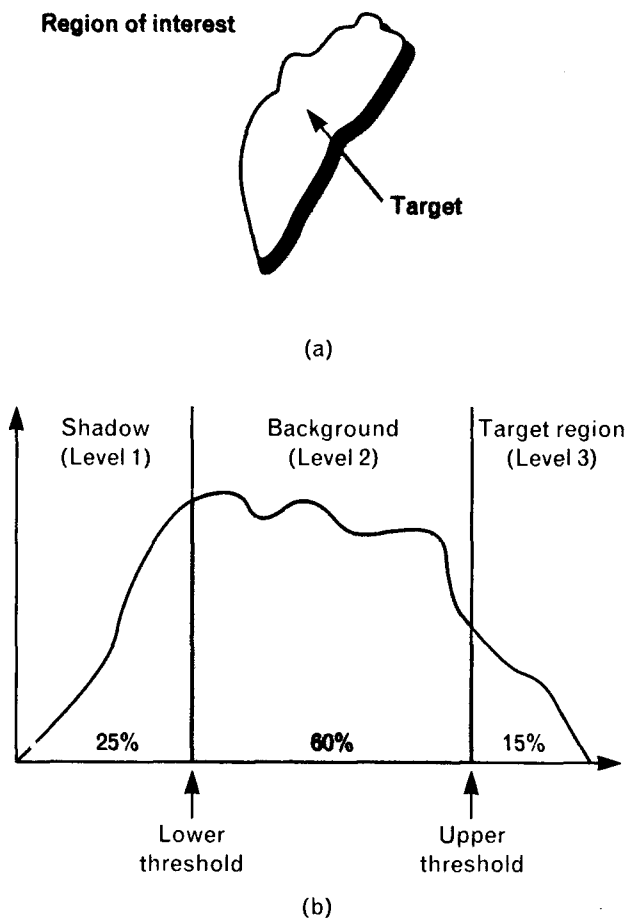
Loral Defense Systems was the third participant in the STAR contract. Many of their discrimination features substantially overlapped the features of the other two contractors and Lincoln Laboratory. For this reason, we examined only one Loral discrimina-



**FIGURE 8.** Scatterplot of the Rockwell specific-entropy feature versus count feature. Points falling on a straight line indicate a high correlation between the two features.

tion feature—the contiguousness feature.

To calculate this feature, we segment each image into three separate images based on the amplitude of individual pixels, as shown in Figure 9. All the pixels in an individual image are histogrammed and then divided into three categories. The lowest 25% of pixels are called *shadow* (Level 1), the middle 60% are called *background* (Level 2), and the top 15% are called the *target region* (Level 3). For the purposes of this study, we modified the procedure by applying the thresholding process only to each region of interest,



**FIGURE 9.** Concept of contiguousness feature. (a) The target within the region of interest has an irregular shape. The radar illuminates this shape from the top, which causes a shadow to extend downward in the image. (b) The two thresholds in the histogram of pixel power (in dB) divide the region of interest into three categories: shadow, background, and target region. The Loral contiguousness feature is computed by first forming six separate regions of interest based on these categories.

which consists of a target-sized area plus a border containing the surrounding clutter. Each region of interest was then segmented into the three categories by using only the clutter area surrounding each target candidate to determine the threshold levels.

The thresholding procedure effectively creates three regions of interest: one that contains the brightest pixels (called the Level 3 image), one that contains the dimmest pixels (called the Level 1 image), and one that contains the midlevel pixels (called the Level 2 image). The same thresholding procedure is performed on the CFAR image, which is derived in the same way as the CFAR image described in the section on ERIM discrimination features and determined by the expression in Equation 1. The contiguousness feature is determined by computing numbers from each of these six regions of interest.

The computation of the contiguousness feature is straightforward. Each contiguousness number is derived from only one image (i.e., the Level 1 CFAR image, the Level 2 CFAR image, the Level 3 CFAR image, the Level 1 image, the Level 2 image, or the Level 3 image). Each pixel included in each particular image is counted, and its immediate neighbors that appear in the same image are counted as well. The count is then normalized by the total number of possibilities that could have occurred (which is nine times the number of pixels in the image). The final number, which has a value between zero and one, is the contiguousness number for that image. This operation is done for every image, so the contiguousness feature gives six separate numbers for each region of interest.

### Discrimination Algorithm

As mentioned at the beginning of the section on algorithm description, the Lincoln Laboratory discrimination algorithm is based on a one-class quadratic discrimination algorithm, the inputs of which are the feature vectors for each candidate region of interest. The algorithm is trained beforehand with representative target data only (no clutter data are used for the training, hence the one-class algorithm). Often these target data consist of images of targets with no countermeasures applied. The tests performed for this article use this training method.

The reasoning behind this type of training is that

predicting the use or modification of enemy targets is impossible, so training on exactly the same types of targets that will be encountered in a real situation is also impossible. Therefore, data gathered by using non-countermeasured targets seem a reasonable choice for a target-training dataset. Any realistic test of the discrimination algorithm, however, must include targets that have some countermeasures applied.

### Theoretical Analysis of the One-Class Quadratic Discrimination Algorithm

The one-class quadratic discrimination algorithm used in the Lincoln Laboratory multistage target-detection algorithm can be described mathematically as

$$Z_i = \frac{1}{n} (\mathbf{X}_i - \hat{\mathbf{M}}_{tr})^T \hat{\mathbf{S}}_{tr}^{-1} (\mathbf{X}_i - \hat{\mathbf{M}}_{tr}), \quad (3)$$

for  $i = 1, 2, \dots, k_t + k_c$ ,

where  $n$  is the number of features used in the discriminator,  $\hat{\mathbf{M}}_{tr}$  and  $\hat{\mathbf{S}}_{tr}$  are the estimates of the mean vector and variance-covariance matrix of the training target set,  $\mathbf{X}_i$  is a random vector representing the observed candidate features, and  $Z_i$  is a random variable representing the distance from the test point to the target-training class. The two variables  $k_t$  and  $k_c$  are the number of targets and the number of clutter false alarms, respectively, that the discriminator receives from the prescreeener stage of the multistage algorithm.

To analyze the discriminator given by Equation 3, we need to find the quantities

$$\text{Prob} \{Z_i < K \mid i \text{ is target}\} = P_d \quad (4)$$

and

$$\text{Prob} \{Z_i < K \mid i \text{ is clutter}\} = P_{fa}, \quad (5)$$

where  $K$  is the hard threshold. This analysis involves finding the probability distribution function (pdf) of  $Z_i$  for the target case and for the clutter false-alarm case, and then integrating the pdf according to Equations 4 and 5.

The distribution of  $Z_i$  for the target-training dataset is easy to calculate if the assumption is made that the estimates of  $\mathbf{M}_{tr}$  and  $\mathbf{S}_{tr}$  take on their true values. For tractability, we also assume that the features are multi-

variate Gaussian distributed. Then the  $Z_i$  values are chi-squared distributed [8], and the pdf can be written as

$$f_n(z) = \frac{z^{\left(\frac{n}{2}-1\right)} \exp\left(-\frac{z}{2}\right)}{\left(\frac{2}{n}\right)^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)},$$

where  $E(z) = 1$  and  $\text{Var}(z) = 2/n$ .

The distribution of  $Z_i$  under the target (non-training) and clutter false-alarm classes with these assumptions is more difficult to calculate. In each of the two cases, a different matrix  $\mathbf{A}$  must be found such that

$$\mathbf{A}_t^T \hat{\mathbf{S}}_{tr} \mathbf{A}_t = \mathbf{I}$$

$$\mathbf{A}_t^T \mathbf{S}_t \mathbf{A}_t = \mathbf{L}_t$$

and

$$\mathbf{A}_c^T \hat{\mathbf{S}}_{tr} \mathbf{A}_c = \mathbf{I}$$

$$\mathbf{A}_c^T \mathbf{S}_c \mathbf{A}_c = \mathbf{L}_c,$$

where  $\mathbf{L}_t$  and  $\mathbf{L}_c$  are diagonal matrices. This operation, which is a simultaneous diagonalization, reduces the problem of evaluating Equations 4 and 5 to one of finding the distribution of

$$\frac{1}{n} \sum_{l=1}^n \lambda_{t,l} (X_l - \omega_{t,l})^2 \quad (6)$$

and

$$\frac{1}{n} \sum_{l=1}^n \lambda_{c,l} (X_l - \omega_{c,l})^2, \quad (7)$$

where

$$\lambda_{t,l} = \text{diag}(\mathbf{L}_t)_l$$

$$\omega_{t,l} = \mathbf{A}_t^T (\mathbf{M}_t - \hat{\mathbf{M}}_{tr})_l$$

and

$$\lambda_{c,l} = \text{diag}(\mathbf{L}_c)_l$$

$$\omega_{c,l} = \mathbf{A}_c^T (\mathbf{M}_c - \hat{\mathbf{M}}_{tr})_l,$$

and where the operator  $\text{diag}(\cdot)$  indicates the extraction of the diagonal vector from the matrix argument. The quantity  $n$  is the number of features used in the discriminator.

Calculating the distribution of Equation 6 or Equation 7 without the multivariate Gaussian assumption would be difficult because the summation would then be over uncorrelated—but not necessarily independent—random variables. Once again, we make an assumption that the estimates of  $\mathbf{M}_{tr}$  and  $\mathbf{S}_{tr}$  take on their true values.

The characteristic function of the distribution of Equations 6 and 7 is given by

$$\phi_{ij}(t) = \exp\left(\sum_{l=1}^n \frac{j t \omega_l^2}{1 - 2 j t \lambda_l}\right) \prod_{h=1}^n (1 - 2 j t \lambda_h)^{-\frac{1}{2}}.$$

In this equation,  $j = \sqrt{-1}$ . We can omit the target and clutter-false-alarm subscripts because the mathematics for the two cases is similar. This characteristic function can be inverted and integrated according to Equations 4 and 5 by using Fourier transform theory, so that

$$\begin{aligned} P_d(K) &= \int_0^{nK} f_t(z) dz \\ &= \frac{1}{2} - \frac{1}{\pi} \int_0^x \left[ \frac{\Im[\phi_{t,n}(q)]}{q} \cos(nKq) \right. \\ &\quad \left. - \frac{\Re[\phi_{t,n}(q)]}{q} \sin(nKq) \right] dq \end{aligned}$$

and

$$\begin{aligned} P_{fa}(K) &= \int_0^{nK} f_c(z) dz \\ &= \frac{1}{2} - \frac{1}{\pi} \int_0^x \left[ \frac{\Im[\phi_{c,n}(q)]}{q} \cos(nKq) \right. \\ &\quad \left. - \frac{\Re[\phi_{c,n}(q)]}{q} \sin(nKq) \right] dq. \end{aligned}$$

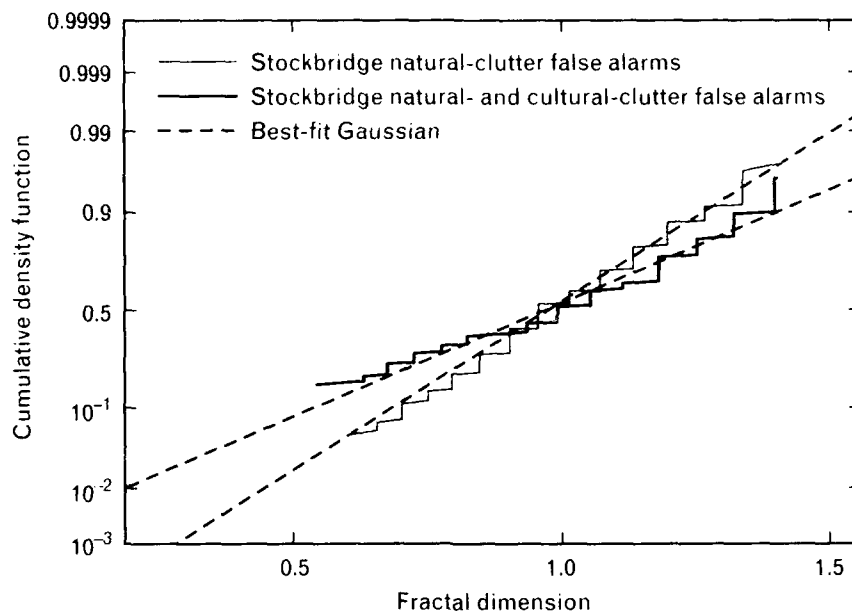
### Confirming the Gaussian Assumption

We make a key assumption in the theoretical performance prediction for the discrimination algorithm given above—we assume that the datasets (target training, target testing, and clutter false alarm) are multivariate Gaussian distributed. There are two tests that are feasible to confirm this assumption; the first test is for the univariate case and the second test is for the bivariate case. Tests for higher dimensions exist, but they can become complicated and difficult to interpret [9] (or they rely on making some other crucial assumption, which can be difficult to check). We chose to perform the univariate and bivariate tests, for which we give the results here. We also created a scatterplot for a trivariate test.

All the tests described here were done as a check on algorithm performance and not as an end in themselves. We did not try to calculate exact quantitative measures for goodness of fit. An exact study would have added considerable complexity to our task, while providing little insight. Instead, our tests were done by using graphical techniques and the fits were performed by eye; only approximate Gaussianity can be ascertained by such techniques. The proof that the theory is an accurate predictor of performance is not contained in these tests, but rather in the comparison of real data results with theoretical results. This comparison is given in the section entitled “Real Data versus Theoretical Performance.”

The univariate test is straightforward. We plot each feature on Gaussian-scaled paper, and test it by examining if the cumulative density function is a straight line. In general, we found that most of the features for most of the datasets were adequately univariate Gaussian. In the few exceptional cases, the distributions were not far off, and the discrepancies were not significant in the final results. Figure 10 is an example of a univariate test with the fractal-dimension feature.

The bivariate case is tested by using scatterplots, which show data points of one feature versus another feature. For Gaussianity, these points should fall in an ellipsoidal bunch around the centroid of the data points. There should be more data points near the center of the ellipse, and fewer data points farther from the center of the ellipse. We could carefully and



**FIGURE 10.** Univariate Gaussian test of the fractal-dimension feature. Any straight line on this graph represents a Gaussian curve.

quantitatively verify the percentage of points within a certain normalized radius of the center of mass, which would give a measure of how Gaussian the data were distributed. We did not, however, perform this quantitative verification; we merely observed how closely the data points were bunched around the center of mass.

These graphs are also useful for checking the correlation between two features; the more linear the data points are, the higher the correlation. If the data points fall in an ellipse that is horizontally or vertically oriented, then the data points are uncorrelated. This test is not just an interesting footnote; the section entitled "Feature Choice Guidelines" describes the importance of choosing features that are orthogonal (i.e., uncorrelated) for good discrimination performance. The scatterplots can also give additional insight into the ability of two features (taken simultaneously) to separate targets from clutter. Ideally, we would like the target-training dataset and the target-testing dataset to be coincident, and the clutter false-alarm dataset to be separated from the other two by a wide margin (measured both along the abscissa and the ordinate).

Figure 11 shows an example of a scatterplot for the fractal-dimension feature versus the weighted-rank fill-ratio feature. The target-training, target-testing,

natural-clutter false-alarm, and natural- and cultural-clutter false-alarm datasets are shown. The target-training and target-testing datasets seem to be reasonably elliptically distributed around their centers of mass, and they seem to be close to each other; both of these properties are desirable. The clutter false-alarm datasets seem to be somewhat less elliptical, but are reasonably well separated from both target datasets.

The three-dimensional scatterplot shown in Figure 12 illustrates all three Lincoln Laboratory discrimination features for the target-training, the target-testing, and the natural-clutter false-alarm datasets described in the section entitled "Data Used." There are two things to be noticed about this figure. First, the figure clearly shows the separation between targets and natural-clutter false alarms, and it shows that the clutter false alarms intermingled with the target datasets tend to be those created by man-made objects (i.e., cultural clutter). Second, the figure helps confirm the approximate Gaussianity of the target datasets and the natural-clutter false-alarm dataset.

Notice the distribution of the red points (the target-training dataset) in the figure. If these red points are Gaussian distributed, they should form an ellipsoidal pattern around the center of the red point cloud with greater density of points toward the center. Likewise, the dark blue points (the target-testing

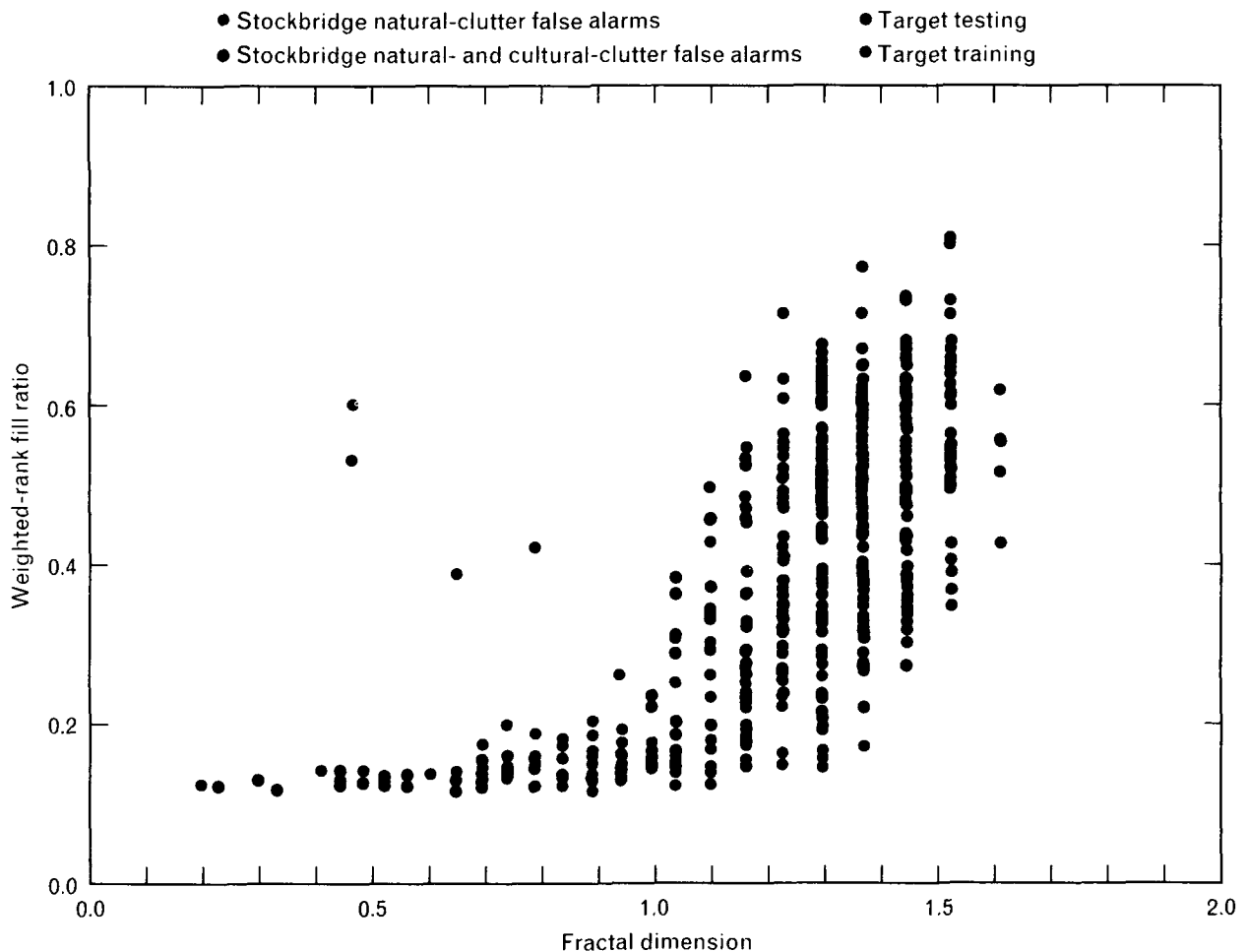
dataset), the green points (the natural-clutter false-alarm dataset), and the light blue points (the cultural-clutter false-alarm dataset) in the figure should be distributed in a similar manner for Gaussianity to hold.

Figure 12 shows that the red points and the dark blue points are distributed in an approximately ellipsoidal pattern around their respective centers. The green points, however, are less ellipsoidal in their distribution, and the light blue points are clearly non-ellipsoidal. As we demonstrate in a later section, the minor deviation of the green points (i.e., the natural-clutter false alarms) from Gaussianity does not greatly affect the agreement between the theory and the real data. The lack of Gaussianity in the light blue points (i.e., the cultural-clutter false alarms) is not critical because the discriminator is designed to eliminate the

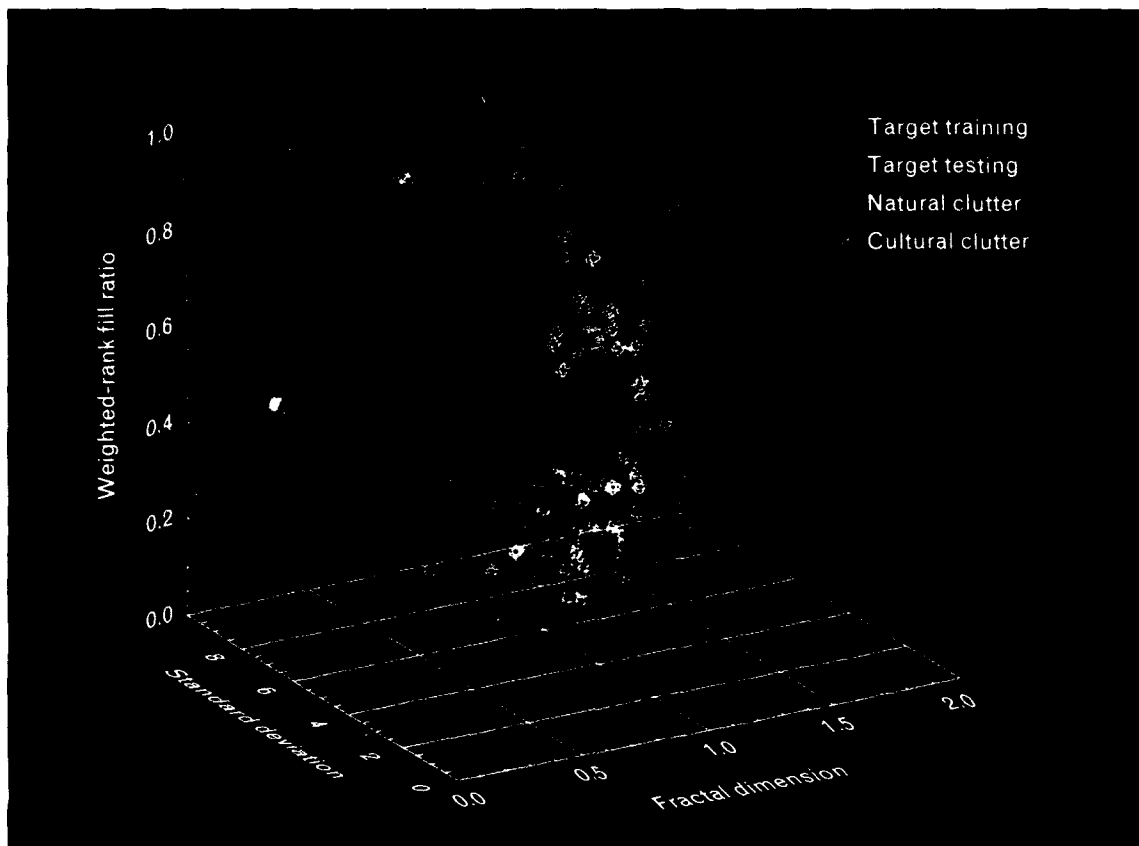
natural-clutter false alarms and pass the cultural-clutter false alarms to the classification algorithm.

The goal of the discrimination algorithm, as stated earlier in this article, is to reject false alarms caused by natural clutter. For most of this article, we do not distinguish between clutter false alarms caused by natural clutter and clutter false alarms caused by cultural clutter, because it is impossible to know, in any kind of realistic scenario, which type of clutter false alarm a given region of interest is (or even if the region of interest is a clutter false alarm or a legitimate target).

In Figure 12 we separate the two types of false alarms for analysis purposes. For the discrimination algorithm to perform well, the targets must be separated from the natural-clutter false alarms. Figure 12 shows that the targets are indeed separated from the natural-clutter false alarms but not from the cultural-



**FIGURE 11.** Scatterplot of fractal-dimension feature versus weighted-rank fill-ratio feature. For good discrimination performance with these two features, the target datasets should be separate from the false-alarm datasets.



**FIGURE 12.** Three-dimensional scatterplot of the three Lincoln Laboratory discrimination features: fractal dimension, weighted-rank fill ratio, and standard deviation.

clutter false alarms. In other words, the discrimination algorithm does well in the role for which it was designed.

### Goals

An important goal for this study is to choose the best set of features from the discrimination feature list given above in the section on discrimination features. As we already stated, these features are standard deviation, fractal dimension, weighted rank fill ratio, mass, diameter, square normalized rotational inertia, maximum CFAR statistic, mean CFAR statistic, percent bright CFAR statistic, percent pure, percent pure even, percent bright even, polarimetric phase ratio, specific entropy, and contiguousness. What determines a best set of these features? Initially we do not know whether the best set contains some combination of three features or four features or more, or all fifteen features, or even if the best set is composed of certain of these features and not others.

An additional goal is that the features chosen must be a robust set of features. Ideally, the features should work equally well, regardless of the target deployment, the countermeasures used, or the type of clutter being imaged. This invariance to data is impossible to achieve; a more realistic goal is for the same features to be part of the best feature set, regardless of data. As a test of this goal, we examine a number of different datasets.

Another goal is to understand the operation of the discrimination algorithm. We would like the theoretical expressions for the performance of the one class quadratic discrimination algorithm to predict the behavior of the discrimination algorithm accurately. If the theory is accurate, then we can predict the performance of the algorithm for different combinations of features. A strategy for feature selection would then be to derive the parameters needed for the theory from the real data, compute the theoretical results for all combinations of features, and choose the best set

of features based on these predictions. We shall see that this strategy is a reasonable one.

We would also like to examine discrimination results for different resolutions and different polarizations. The Lincoln Laboratory MMW SAR has a resolution of 1 ft by 1 ft, but we can easily construct lower-resolution, single-look radar imagery from the data at hand. Additionally, the Lincoln Laboratory MMW SAR is fully polarimetric (fully polarimetric SAR data allow synthesis of any polarization or combination of polarizations). Many radar sensors are not fully polarimetric, and use only a single polarization. The most common polarization used is HH; therefore, we use HH data as well as fully polarimetric data in this study. We expect the best feature set to change, depending on our choice of resolution and polarization.

### Choosing a Feature Set

The method we use to choose the best feature set is straightforward. First, the data are prescreened by using a simple two-parameter CFAR algorithm [2, 10]. This stage is designed to eliminate (with a minimum of computation) only the most obviously non-targetlike clutter. The prescreening algorithm operates on imagery that has already been reduced to a resolution of 1 m by 1 m. The resolution was reduced by taking a noncoherent average of each 4-pixel-by-4 pixel non-overlapping box (a pixel has a nominal resolution of approximately 0.23 m). This method of resolution reduction has two advantages: (1) it reduces the amount of data we need to process, and (2) it reduces the speckle that is present in the high-resolution SAR imagery (the article by Leslie M. Novak et al. in this issue gives an explanation of speckle in SAR imagery).

In the prescreener for this study we use a threshold value that allows the detection of 80% of the targets. This percentage was chosen for consistency among datasets; it was also chosen by considering the number of clutter chips that are passed to the discrimination stage. A higher probability of detection in the prescreener stage necessarily increases the number of clutter false alarms passed to the discriminator. Computation time and storage limitations preclude using a higher percentage value for the prescreener prob-

ability of detection. The data used in the prescreener algorithm were also processed by using the polarimetric whitening filter (PWF) [1], which combines the HH, HV, and VV polarization channels together in a manner that optimally decreases speckle. The HH polarization results use only the HH polarization SAR data, and hence do not use the PWF imagery for the prescreener algorithm.

The candidates identified by the prescreener (either on targets or on clutter false alarms) are then grouped spatially. The grouping algorithm is a simple one; all hits within a target-sized area are grouped into a single detection. This grouping operation exploits some of the spatial information inherent in the proximity of prescreener hits.

The discrimination algorithm is run on all the regions of interest selected by the prescreener and the grouping algorithm. First, all the features described in the section on discrimination features are computed for all regions of interest. The features are computed for the following four combinations of data: (1) 1-ft resolution and PWF polarization, (2) 1-m resolution and PWF polarization, (3) 1-ft resolution and HH polarization, and (4) 1-m resolution and HH polarization. The features were originally tuned (in terms of the thresholds used in the feature calculations themselves) for the 1-ft resolution, PWF case. The features are used without modification for this case as well as for the 1-ft resolution, HH polarization case.

Naturally, the polarimetric features cannot be calculated for the HH polarization case because the polarimetric features use polarizations other than HH. We therefore use a reduced set of features. For the 1-m resolution cases, we retune the features by computing them for a range of thresholds, and we choose the threshold that provides the best separation between targets and clutter false alarms for all datasets. This retuning is done separately for the PWF case and the HH polarization case. Therefore, these features are intended to give best-case results. Any use of these tuned features in other datasets can only approach the results shown in the article in general. Certainly, the 1-m resolution tests provide a better indication of the performance of the discrimination algorithm than is likely to be obtained in a real situation.

The parameters necessary for a theoretical evalua-



## INTERPRETING PLOTS OF $P_D$ VERSUS $FA/KM^2$ <sup>2</sup>

THE METHOD OF EVALUATION for the discrimination algorithm described in this paper involves plotting a curve that shows the probability of detection ( $P_D$ ) versus the number of false alarms per square kilometer ( $FA/km^2$ ). The measure of  $FA/km^2$  scales directly to the probability of false alarm, which was theoretically derived for the discrimination algorithm in the section entitled "Theoretical Analysis of the One-Class Quadratic Discrimination Algorithm." Such curves are often referred to as receiver operating-characteristic (ROC) curves.

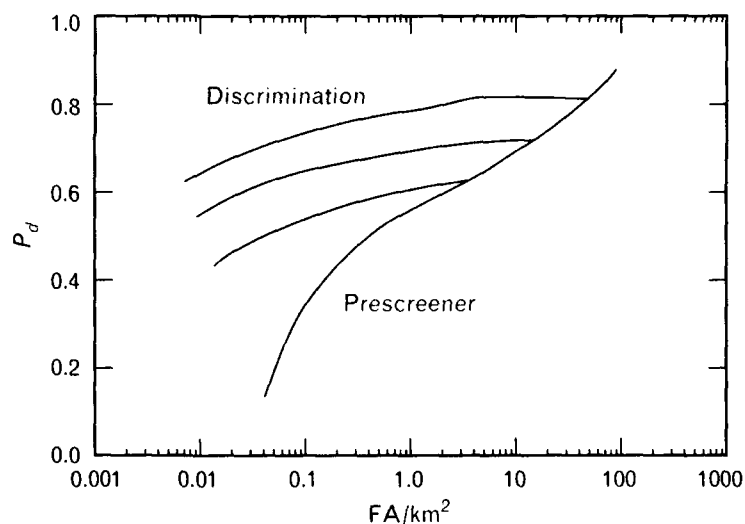
Figure A gives an example of a simple ROC curve (in red). Better performance is indicated in these types of plots by a curve moving upward and leftward. A plot such as this one might be used to evaluate the prescreener stage or the discrimination stage separately. A more complicated plot is necessary to evaluate the combination of the prescreener and discrimination stages.

Figure A also shows an example of a plot (blue and red) that might be used to evaluate both the prescreener and discrimination stages combined. Notice that the original ROC has grown a

number of extra lines, or *tails*. These extra lines (in blue) represent the improved performance provided by the discrimination stage of the multistage target-detection algorithm. Each extra line meets the curve of the original prescreener stage at a certain point. Each discrimination line emanating from these points describes the operating characteristic of the discrimination algorithm

for the particular set of inputs being provided by the prescreener at that operating point.

As the threshold of the prescreener is varied, the set of inputs provided to the discrimination algorithm varies as well. The evaluation criterion for performance in an ROC curve works here as well; the line moving upward and leftward indicates better performance.



**FIGURE A.** Example of a  $P_D$  versus  $FA/km^2$  curve, which is also known as a receiver operating-characteristic curve, for a multistage target-detection algorithm. The additional lines represent the performance of the discrimination stage of the algorithm. Three of these performance lines are shown; in fact, an infinite number of them are possible, because their intersections with the prescreener curve are dictated by the level at which the prescreener stage is operated.

tion of the discrimination performance are computed in each case from a target region-of-interest dataset and a clutter false-alarm region-of-interest dataset. Additionally, the assumptions necessary for the theory

to hold are checked in most cases. These checks are more fully detailed in the section entitled "Confirming the Gaussian Assumption." All combinations of the discrimination features are tested by using the

theory. Theoretical  $P_d$  versus  $P_{fa}$  plots are produced for each combination, and the best combinations are chosen for further analysis. The best combinations of features are given in the section entitled "Best Features for Discrimination." The results we get by using the real data are then generated for the short list of good combinations. The real-data results are then compared with the theoretical results; these comparisons are given in the section entitled "Real Data versus Theoretical Performance."

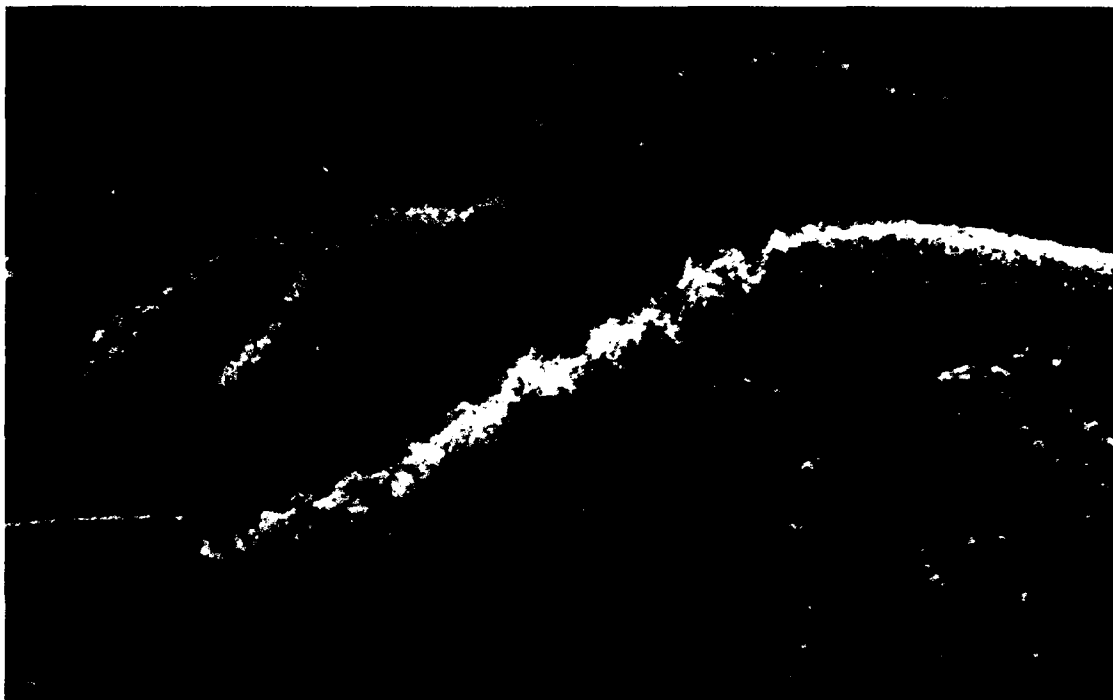
Performance evaluation is done by plotting the probability of detection  $P_d$  versus the number of false alarms per square kilometer (FA/km<sup>2</sup>). The measure of false alarms per square kilometer is merely a rescaling of the probability of false alarm ( $P_{fa}$ ) into an operationally meaningful measure. This rescaling is performed to remove the effect of sensor resolution (because a higher resolution image inherently gives more opportunities for false alarms to occur, the same  $P_{fa}$  value at different resolutions means different numbers of false alarms per square kilometer). The inter-

pretation of plots of  $P_d$  versus FA/km<sup>2</sup> is reviewed in the sidebar entitled "Interpreting Plots of  $P_d$  versus FA/km<sup>2</sup>."

### Data Used

All the target data used in this study were gathered with the Lincoln Laboratory MMW SAR in Stockbridge, New York. The targets consisted of two datasets of the same targets in different deployment conditions. The first dataset, which we use for discrimination algorithm training, is called the target-training dataset. The second dataset, which we use for discrimination algorithm testing, is called the target-testing dataset. There are three distinct clutter datasets; two gathered at Stockbridge, New York, and a smaller clutter dataset gathered in Concord, Massachusetts.

The first clutter dataset, which consists of mostly natural clutter, is called the Stockbridge natural-clutter dataset. Figure 13 is an example of this dataset; it shows a river with treelined banks (the river is the dark area curving through the middle of the image).



**FIGURE 13.** SAR image of natural clutter in Stockbridge, New York. The sensor is flying parallel to the top of the image, and the shadows extend downward in the image. Areas of high radar return are colored in bright yellow; areas of low radar return are in dark colors. The dark band in the middle of the image is a river with trees lining each bank. The smooth green areas are open fields.

The remainder of the image is an open field. Freshly plowed furrows in the open field can also be seen. The radar illuminates the area from the top of the image; therefore, the shadows cast by the trees point downward in the image. The Stockbridge natural-clutter dataset also includes some man-made objects (which are impossible to avoid entirely in the Stockbridge area), including the farmhouse shown in an earlier article by Novak [1]. The clutter in this dataset is considered to be relatively benign.

The second clutter dataset is called the Stockbridge natural- and cultural-clutter dataset. This dataset was gathered from a different area of the same Stockbridge collection site; it includes a farm-supply store that is shown in Figure 14 both as a SAR image and in an aerial photograph. The clutter in this dataset is considered to be moderately difficult.

The third clutter dataset is a small dataset gathered in Concord, Massachusetts, which is a few miles from Lincoln Laboratory. This dataset, which we refer to as the Concord dataset, consists entirely of man-made clutter, and is considered to be a very difficult dataset.

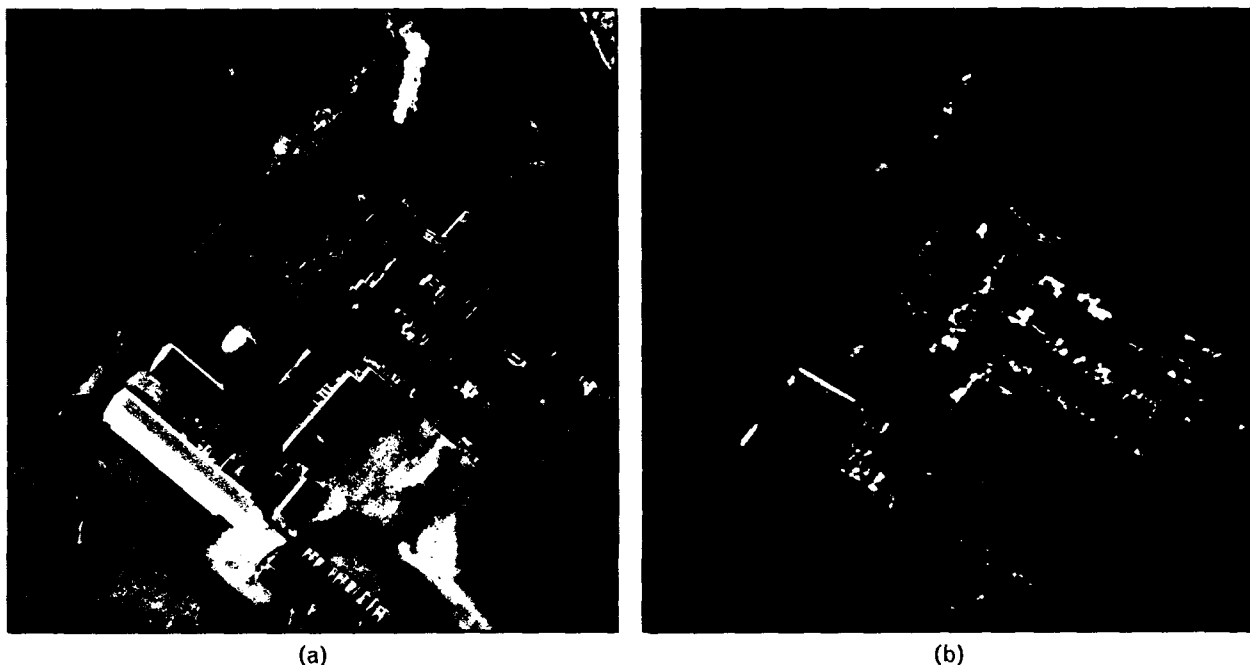
Figure 15 shows an example of imagery from this dataset.

### Best Features for Discrimination

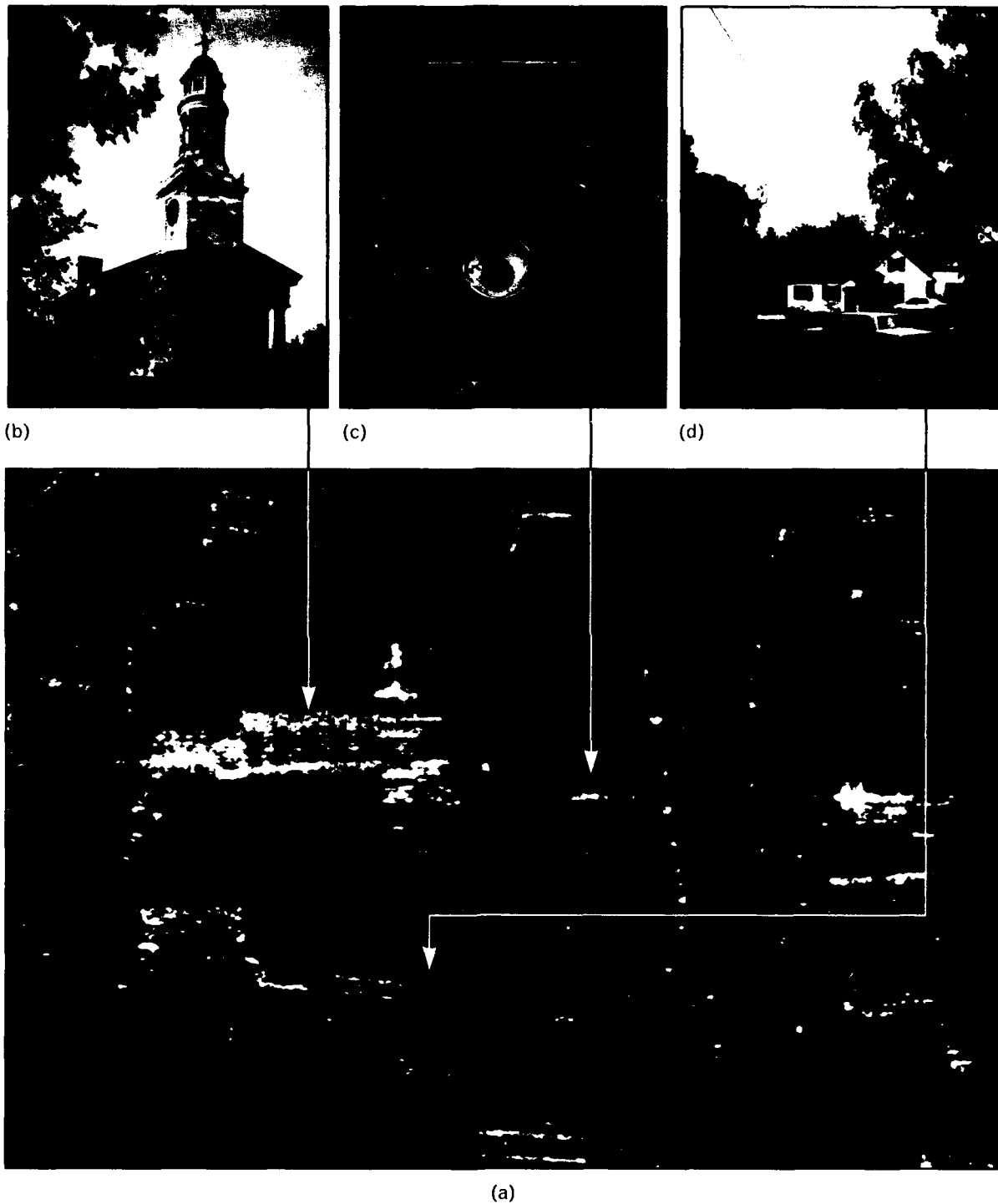
The method used to determine the best features for discrimination is fully described in the earlier section entitled "Choosing a Feature Set." For two cases—the 1-ft resolution, PWF data of the Stockbridge natural-clutter dataset and the Stockbridge natural- and cultural-clutter dataset—we found the best features to be those given in Table 1. For the case of the 1-ft resolution, PWF, Concord man-made clutter, the feature set reduced to those features given in Table 2.

As stated earlier, we did not attempt to pick the best features for the 1-ft resolution, HH-polarization case. Instead we evaluated performance with the same features as the best-case features for the PWF data.

For the 1-m resolution, PWF, Stockbridge natural-clutter, and the natural- and cultural-clutter case, we found the best features to be those given in Table 3. The "optional" qualifier given in the table means that the feature does not increase or decrease any perfor-



**FIGURE 14.** (a) An optical photograph and (b) a SAR image of a farm-supply store in Stockbridge, New York. This store is an example of a man-made clutter discrete. The store parking lot is in the bottom of each image. Although the photograph and the SAR image were taken at different times, passenger cars can be seen in the parking lot in both images. The bright spots in the middle right area of the SAR image are caused by various metallic objects in the yard of the supply store.



**FIGURE 15.** (a) A SAR image of man-made clutter in Concord, Massachusetts. The other three photographs illustrate specific objects visible in the SAR image: (b) the church and steeple, (c) a spotlight that illuminates the church at night, and (d) a house and a telephone wire suspended overhead. Note the bright columns along the side of the church in the SAR image. These columns clearly correspond in number and placement to the areas between the windows of the church in the optical photograph. Also notice the bright circular feature—the clock—on the church steeple.

**Table 1. Best Features for 1-ft, PWF, Natural-Clutter Dataset and Natural- and Cultural-Clutter Dataset**

<i>Feature</i>	<i>Description</i>
Fractal dimension	Lincoln Laboratory
Weighted-rank fill ratio	Lincoln Laboratory
Diameter	ERIM size
Mean CFAR or percent bright CFAR	ERIM CFAR
Percent pure	ERIM polarimetric

**Table 2. Best Features for 1-ft, PWF, Man-Made Clutter Dataset**

<i>Feature</i>	<i>Description</i>
Fractal dimension	Lincoln Laboratory
Percent bright CFAR	ERIM CFAR
Percent pure	ERIM polarimetric

**Table 3. Best Features for 1-m, PWF, Natural-Clutter Dataset and Natural- and Cultural-Clutter Dataset**

<i>Feature</i>	<i>Description</i>
Fractal dimension	Lincoln Laboratory
Diameter	ERIM size
Percent bright even	ERIM polarimetric
Percent pure	ERIM polarimetric
Mean CFAR	ERIM CFAR (optional)

mance ability with these datasets, but it could add or subtract a certain amount of robustness for other datasets. The best features for the 1-m resolution, HH-polarization case are given in Table 4. For this case, however, the discrimination algorithm provides

**Table 4. Best Features for 1-m, HH, Natural-Clutter Dataset and Natural- and Cultural-Clutter Dataset**

<i>Feature</i>	<i>Description</i>
Fractal dimension	Lincoln Laboratory
Diameter	ERIM size
Mean CFAR	ERIM CFAR (optional)

little or no performance gain over the prescreener alone.

### Feature Choice Guidelines

An examination of the list of best features from the previous section, along with the scatterplots shown in the section entitled "Confirming the Gaussian Assumption," reveals some interesting and important guidelines for choosing the best features. There are two general criteria for feature choice for this discrimination algorithm—*separation* and *orthogonality*. The separation criterion is the common-sense consideration that the feature must adequately separate the target training (and target testing) dataset from the clutter false-alarm dataset. The orthogonality criterion is less intuitive, and can be summarized by the idea that different features used in the discrimination algorithm must measure different attributes of the region of interest.

Unfortunately, we cannot easily predict exactly which attribute of a region of interest a feature measures. Sometimes two features that beforehand would seem to be highly correlated ultimately exhibit a low degree of correlation. We show an example of this type of behavior later in this section.

The best features listed in Table 1 are a good example of the orthogonality criterion. We see that the table includes two of the three Lincoln Laboratory discrimination features, which is not surprising because the three Lincoln Laboratory features were designed with orthogonality in mind. The first feature (fractal dimension) exploits the spatial relationship of the top  $N$  scatterers in the region of interest, while the second feature (weighted-rank fill ratio) exploits the

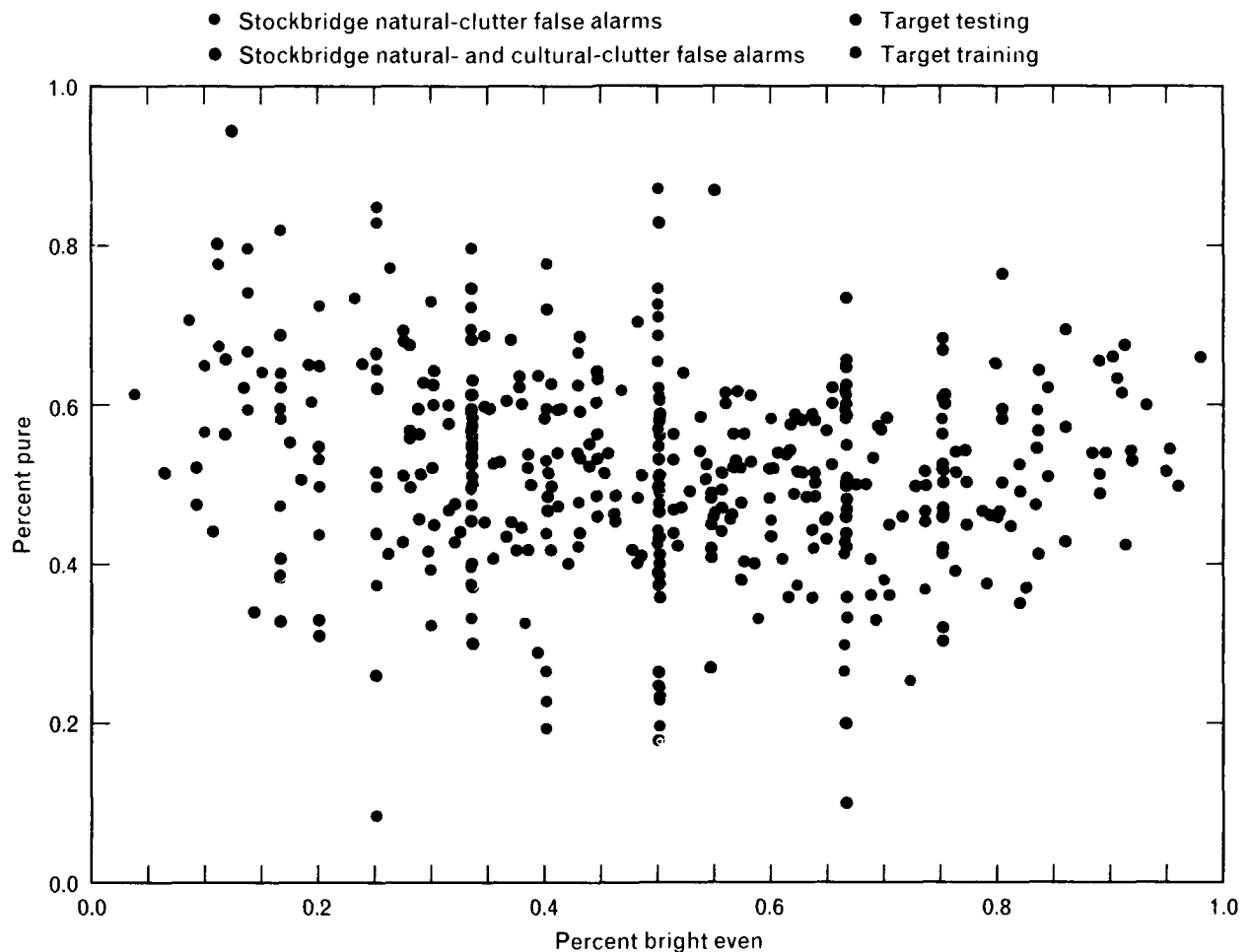
distribution of reflected power among all the scatterers on the target. Clearly, these two features were designed to measure different characteristics of the region of interest.

The three other features included as best features in the case shown in Table 1 all come from the ERIM discrimination features. Interestingly, the three chosen features each come from a different subset of features: the first comes from the ERIM size features, the second comes from the ERIM CFAR features, and the third comes from the ERIM polarimetric features. Even if the ERIM features were not designed with the orthogonality criterion in mind, we find it interesting that the choice of best features naturally selects one feature from each category.

A subset of the features listed in Table 1 works best in the man-made clutter dataset, as shown in Table 2.

The orthogonality criterion holds here as well, except that the two features not included, which were included in Table 1, no longer provide reasonable separation between targets and clutter false alarms.

For the case of 1-m resolution, there is an apparent exception to the two criteria given above in the best feature choices. Notice that Table 3 contains two ERIM polarimetric features. Figure 16 shows a scatterplot of these two features (percent bright even and percent pure) for the 1-m resolution datasets. From the scatterplot we can see that these two features are, in fact, uncorrelated and are therefore orthogonal in some meaningful sense. Apparently, in the 1-m resolution dataset the thresholding involved in calculating the percent-bright-even feature causes this feature to measure something other than the polarimetric properties of the region of interest. The



**FIGURE 16.** Scatterplot of percent bright even feature versus percent pure feature for 1-m-by-1-m resolution data. These two features are uncorrelated because the data points do not fall along a straight line.

five features chosen in Table 3 for the 1-m resolution case obey both the separation and orthogonality criteria given above. The same holds true for the best features in the HH-polarization case, which are given in Table 4.

### Real Data versus Theoretical Performance

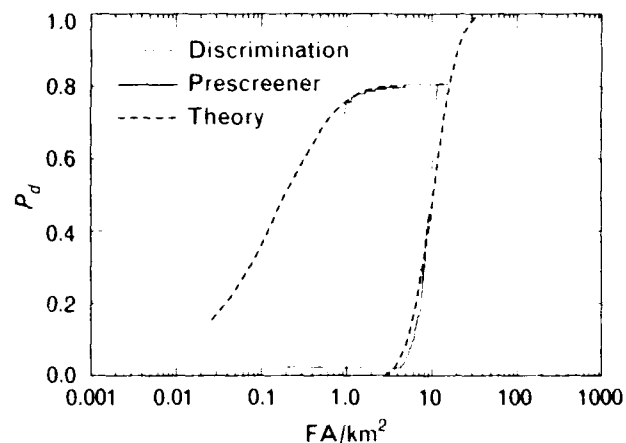
This section gives the real-data prescreener and discrimination results in the form of plots of  $P_d$  versus  $FA/km^2$ . These plots are explained in the sidebar entitled "Interpreting Plots of  $P_d$  versus  $FA/km^2$ ." We also plot on the same graphs the predictions computed from the theoretical analysis given in the section entitled "Theoretical Analysis of the One-Class Quadratic Discrimination Algorithm." In all cases, the theory and real data coincide closely. This fact demonstrates that the one-class quadratic discrimination algorithm is well understood as it is implemented in the Lincoln Laboratory multistage target-detection algorithm.

Figure 17 gives the combined prescreener and discrimination results for the 1-ft resolution, PWF data for the Stockbridge natural-clutter dataset, while Figure 18 gives the prescreener and discrimination results for the Stockbridge natural- and cultural-clutter dataset, and Figure 19 gives the prescreener and discrimination results for the Concord man-made-clutter dataset. Figures 20 and 21 show the prescreener and discrimination results for the Stockbridge natural-clutter dataset and the Stockbridge natural- and cultural-clutter dataset, respectively. Both results are for 1-ft resolution, HH-polarization data.

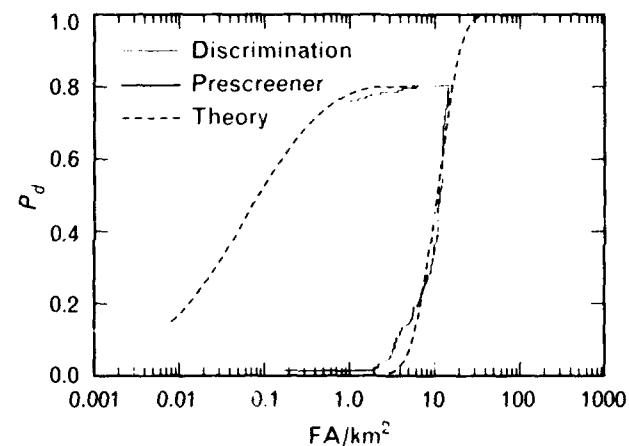
The remaining results are for the 1-m resolution case. Figures 22 and 23 show the prescreener and discrimination results for the Stockbridge natural-clutter dataset and the Stockbridge natural and cultural-clutter dataset, respectively, for PWF data. Figures 24 and 25 show the prescreener and discrimination results for the same two datasets for the HH-polarization case.

### Polarization Comparisons

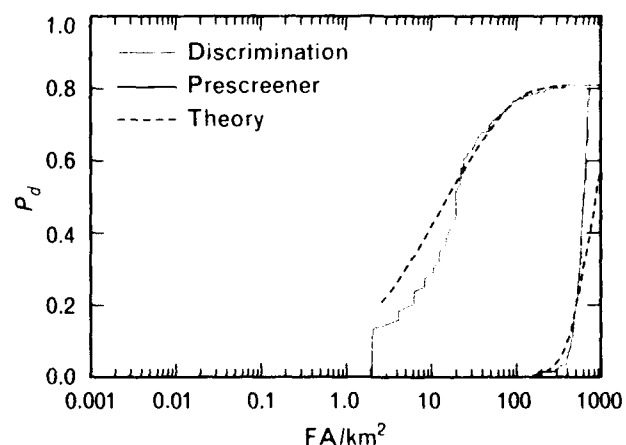
We can compare the discrimination results from the PWF data and the HH-polarization data for the same cases to draw a conclusion regarding the advantage of using a fully polarimetric radar versus using the more



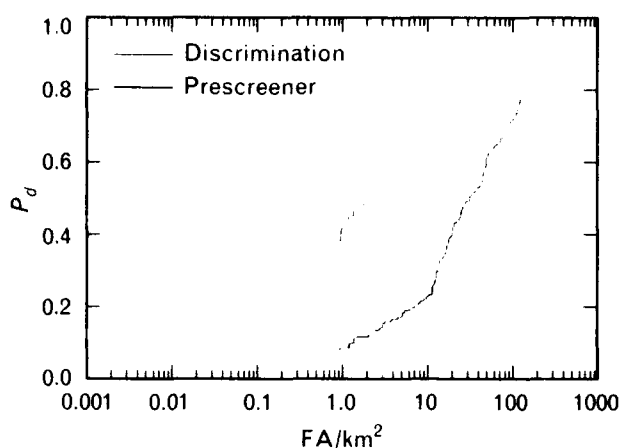
**FIGURE 17.** Comparison of real data and theoretical results for the 1-ft-by-1-ft resolution, PWF, Stockbridge natural-clutter case.



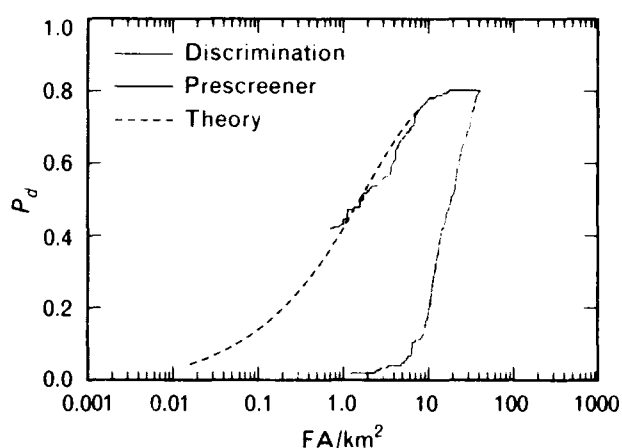
**FIGURE 18.** Comparison of real data and theoretical results for the 1-ft-by-1-ft resolution, PWF, Stockbridge natural- and cultural-clutter case.



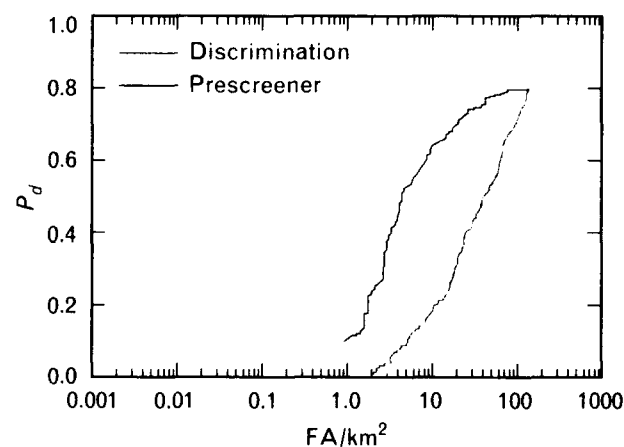
**FIGURE 19.** Comparison of real data and theoretical results for the 1-ft-by-1-ft resolution, PWF, Concord man-made clutter case.



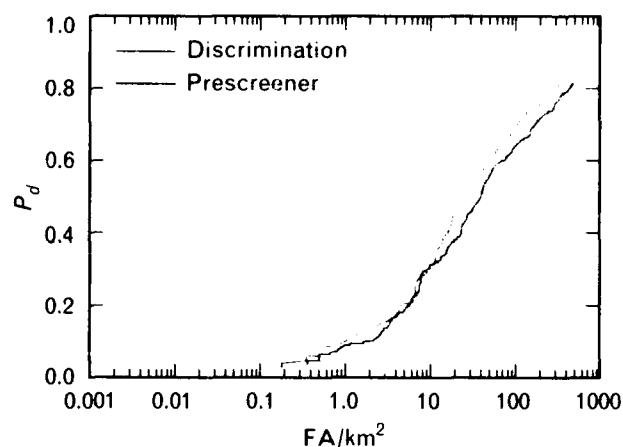
**FIGURE 20.** Comparison of real data and theoretical results for the 1-ft-by-1-ft resolution, HH-polarization, Stockbridge natural-clutter case.



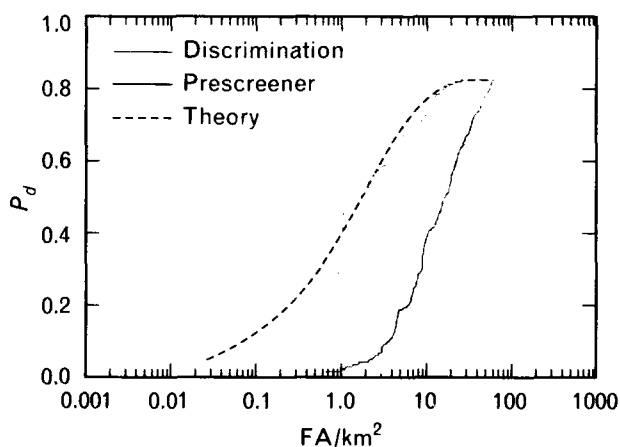
**FIGURE 23.** Comparison of real data and theoretical results for the 1-m-by-1-m resolution, PWF, Stockbridge natural- and cultural-clutter case.



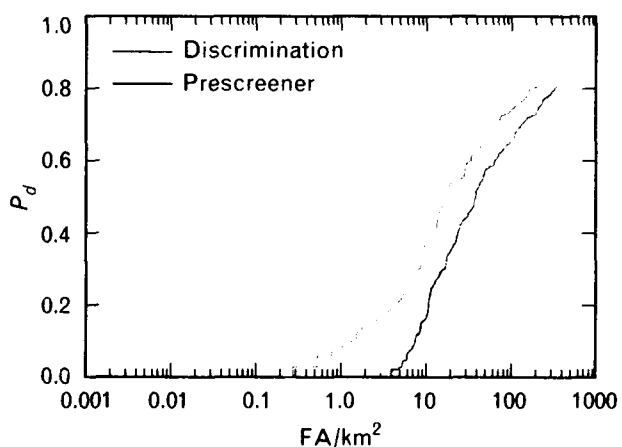
**FIGURE 21.** Comparison of real data and theoretical results for the 1-ft-by-1-ft resolution, HH-polarization, Stockbridge natural- and cultural-clutter case.



**FIGURE 24.** Comparison of real data and theoretical results for the 1-m-by-1-m resolution, HH-polarization, Stockbridge natural-clutter case.



**FIGURE 22.** Comparison of real data and theoretical results for the 1-m-by-1-m resolution, PWF, Stockbridge natural-clutter case.



**FIGURE 25.** Comparison of real data and theoretical results for the 1-m-by-1-m resolution, HH-polarization, Stockbridge natural- and cultural-clutter case.



common HH single-polarization radar. We see that the performance increase is significant (a reduction of one to two orders of magnitude in false-alarm rate for equal probabilities of detection) in both the 1-ft resolution and the 1-m resolution cases. The performance difference is more pronounced for the higher-resolution data.

In general, during the course of this study, we noticed that the combination of higher-resolution data and fully polarimetric data provided a significant increase in performance. Either capability alone is not nearly as effective as the two capabilities together for the discrimination features we have studied in this article. In other words, building radars with both higher resolution and with fully polarimetric capability makes sense.

In the 1-m resolution case, the difference between the PWF data and the HH-polarization data is clear. Using the HH data alone, the discrimination algorithm provides little or no performance improvement over using the prescreeener algorithm alone. The features for the 1-m, HH-polarization case were tuned specifically for these datasets, so this result should be considered a best case. Clearly, there is no point in using the discrimination algorithm with these features for the 1-m resolution, HH-polarization dataset, because it provides little benefit and it requires additional computational capacity.

### Resolution Comparisons

We can also compare the results from the 1-ft resolution case with the results from the 1-m resolution case. We see that the higher-resolution data allows a performance increase of more than an order of magnitude in terms of the false-alarm rate for a given probability of detection. This performance increase is approximately constant over the different cases given in Figures 17 to 25.

### Conclusion

In this article, we discuss and evaluate the discrimination algorithm used in the Lincoln Laboratory multistage target-detection algorithm. This one-class quadratic discriminator uses features calculated from SAR imagery. The discrimination algorithm uses candidate regions of interest identified by the prescreeener, and

ideally eliminates all natural-clutter false alarms from further consideration, passing only targets and man-made clutter false alarms to the classification algorithm.

Fifteen discrimination features were evaluated for this study; three of the features were developed by Lincoln Laboratory and the remainder were developed by the three STAR contractors. These features were modified to account for the different types of data used in this study, and the best set of features was chosen for a number of different datasets and a number of different types of data. The best features remained constant from the natural-clutter dataset to the natural- and cultural-clutter dataset, which was a surprising and pleasing result. For best performance, we needed to select different feature sets for PWF and HH-polarization data, as well as for 1-ft and 1-m resolution data, which was not a surprising result.

We evaluated the features by using a theoretical expression that accurately predicted the real-data performance of the discrimination algorithm. This accuracy reflects a good understanding of how the discrimination algorithm functions as a part of the Lincoln Laboratory multistage target-detection algorithm for SAR data.

### Acknowledgments

The authors would like to thank Dr. Leslie Novak for the help and encouragement given throughout the course of this study.

## REFERENCES

1. L.M. Novak, M.C. Burl, R.D. Chaney, and G.J. Owirka, "Optimal Processing of Polarimetric Synthetic-Aperture Radar Imagery," *Lin. Lab. J.* 3, 273 (1990).
2. W.W. Irving, G.J. Owirka, and L.M. Novak, "Adaptive Processing of POL-SAR Imagery," *Proc. IEEE 24th Asilomar Conf. on Circuits, Systems, and Computers, Pacific Grove, CA, November 5-7 1990*, p. 388.
3. K. Fukunaga, R.R. Hayes, and L.M. Novak, "The Acquisition Probability for a Minimum Distance One-Class Classifier," *IEEE Trans. Aerosp. Electron. Syst.* 23, 493 (1987).
4. L.M. Novak, "On the Sensitivity of Bayes and Fisher Classifiers in Radar Target Detection," *Proc. IEEE 18th Asilomar Conf. on Circuits, Systems, and Computers, Pacific Grove, CA, November 5-7, 1984*, p. 367.
5. D.E. Kreithen and S.D. Halversen, "A Theoretical Analysis of a Ranking Discrimination Algorithm," *Proc. IEEE 26th Asilomar Conf. on Circuits, Systems, and Computers, Pacific Grove, CA, October 26-28, 1992*, p. 431.
6. M.C. Burl, G.J. Owirka, and L.M. Novak, "Texture Discrimination in Synthetic Aperture Radar Imagery," *Proc. IEEE 23rd Asilomar Conf. on Circuits, Systems, and Computers, Pacific Grove, CA, October 30-November 1, 1989*, p. 399.
7. S.D. Halversen, "Calculating the Orientation of a Rectangular Target in SAR Imagery," *Proc. IEEE 1992 National Aerospace and Electronics Conf. (NAECON '92), Dayton, Ohio, 18-22 May 1992*, p. 260.
8. N.L. Johnson and S. Kotz, *Continuous Univariate Distributions 2* (Wiley, New York, 1970).
9. R.R. Parenti and E.W. Tung, "A Statistical Analysis of the Multiple Target Multiple Shot Target Acquisition Problem," *Project Report TT-43*, MIT Lincoln Laboratory, 28 Jan. 1981.
10. G.B. Goldstein, "False-Alarm Regulation in Log-Normal and Weibull Clutter," *IEEE Trans. Aerosp. Electron. Syst.* 9, 84 (1973).



**DANIEL E. KREITHEN** is a staff member in the Surveillance Systems group. His research speciality is in the detection of stationary targets in synthetic-aperture radar imagery. He received a Sc.B. degree in engineering from Brown University, and an M.S.E. degree in electrical engineering from Princeton University. Dan has been at Lincoln Laboratory since 1988.



**SHAWN D. HALVERSEN** is an associate staff member in the Surveillance Systems group. His research speciality is in the detection and discrimination of stationary targets. He received a B.A. degree in mathematics and a B.S. degree in physics from the University of West Florida, and an M.A. degree in mathematics from the University of Wisconsin. He has been at Lincoln Laboratory since 1990.



**GREGORY J. OWIRKA** is an assistant staff member in the Surveillance Systems group. He received a B.S. degree (*cum laude*) in applied mathematics from Southeastern Massachusetts University, and he is currently working on an M.S. degree in electrical engineering at Northeastern University. Greg's current research interests are in automatic target recognition. He has been at Lincoln Laboratory since 1987.

# Improving a Template-Based Classifier in a SAR Automatic Target Recognition System by Using 3-D Target Information

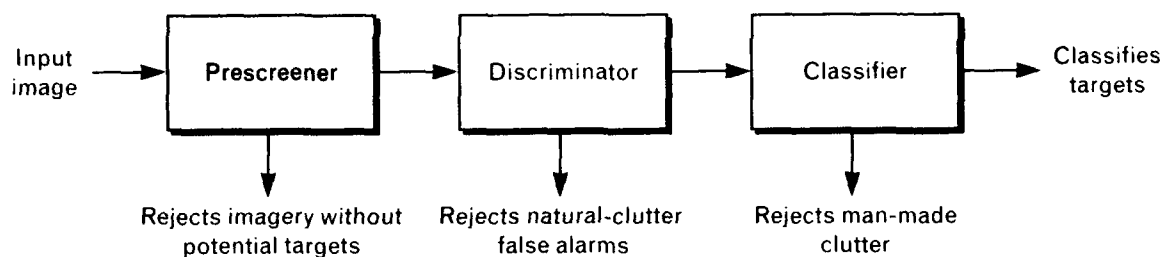
Shawn M. Verbout, William W. Irving, and Amanda S. Hanes

■ In this article we propose an improved version of a conventional template-matching classifier that is currently used in an operational automatic target recognition system for synthetic-aperture radar (SAR) imagery. This classifier was originally designed to maintain, for each target type of interest, a library of 2-D reference images (or *templates*) formed at a variety of radar viewing directions. The classifier accepts an input image of a target of unknown type, correlates this image with a reference template selected (by matching radar viewing direction) from each target library, and then classifies this image to the target category with the highest correlation score. Although this algorithm seems reasonable, it produces surprisingly poor classification results for some target types because of differences in SAR geometry between the input image and the best-matching reference image. Each reference library is indexed solely by radar viewing direction, and is thus unable to account for radar motion direction, which is an equally important parameter in specifying SAR imaging geometry. We correct this deficiency by incorporating a model-based reference generation procedure into the original classifier. The modification is implemented by (1) replacing each library of 2-D templates with a library of 3-D templates representing complete 3-D radar-reflectivity models for the target at each radar viewing direction, and (2) including a mathematical model of the SAR imaging process so that any 3-D template can be transformed into a 2-D image corresponding to the appropriate radar motion direction before the correlation operation is performed. We demonstrate experimentally that the proposed classifier is a promising alternative to the conventional classifier.

**A**N AUTOMATIC TARGET RECOGNITION (ATR) system is an integrated collection of algorithms designed to process sensor measurements so that targets can be efficiently detected and identified. The algorithms that comprise an ATR system are applied on a computer and are organized so that human intervention is not required.

An important component of any ATR system is its

classifier. The function of the classifier is to take input measurements that represent detected targets and categorize these inputs according to target type. The classifier is designed with the assumption that each input belongs to one and only one category from a predetermined set (e.g., *tank*, *truck*, *gun*), and that the input has certain observable characteristics that aid in its assignment to this category. The classifier output



**FIGURE 1.** Block diagram of the three-stage SAR automatic target recognition system developed by Novak. The input consists of SAR imagery representing many square kilometers of terrain and potentially containing several targets of interest; the output consists of locations and classification labels for these targets. This article proposes an improved version of the classifier stage.

corresponding to each input is an estimate of the correct category label, based on the observable characteristics of the input.

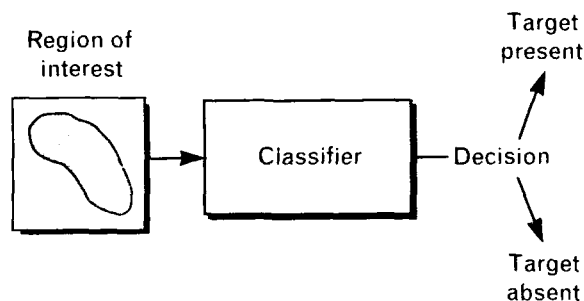
Of course, the issues involved in building a classifier vary according to the kinds of sensor measurements being processed. In this article, we are concerned exclusively with a classifier whose inputs are 2-D synthetic-aperture radar (SAR) images. In particular, we analyze and suggest extensions to the structure of the classifier in the SAR ATR system developed by Leslie M. Novak (see the article entitled "Performance of a High-Resolution Polarimetric SAR Automatic Target Recognition System" in this issue). This ground-based system has been designed to operate in an off-line, experimental setting. It has been rigorously tested over the past five years, and it is one of the first systems of its kind to process large quantities of actual SAR data.

Novak's SAR ATR system is conveniently decomposed into a sequence of three processors: a detector (or *prescreener*), a discriminator, and a classifier (see Figure 1). The detector searches through imagery representing many square kilometers of terrain, and outputs a collection of regions of interest centered at possible target locations. (Each region of interest is a subimage extracted from the original SAR dataset; collectively, all of the regions of interest comprise only a small fraction of this original dataset.) The discriminator applies further processing to distinguish between two kinds of regions of interest: those containing man-made objects (i.e., either targets or man-made clutter) and those containing natural clutter.

All regions of interest that appear to contain natural clutter are discarded. Finally, the classifier assigns each remaining region of interest to a predefined target category, or to a none-of-the-above category if the region of interest appears to contain man-made clutter.

Although Novak's ATR system is usually applied to the multiclass target identification problem (i.e., the problem in which two or more kinds of targets must be distinguished), for convenience we consider only the one-class problem in this article. In the one-class problem only one kind of target is of interest; therefore, the classifier output reduces to a simple yes/no decision that indicates whether a target of this kind is present in the region of interest. Figure 2 illustrates the input-output operation of a one-class classifier.

Novak's classifier, which we refer to as the *baseline* classifier, uses a conventional template-matching al-



**FIGURE 2.** Illustration of the input-output operation of the baseline one-class classifier. The input is a subimage, or region of interest, extracted from the original SAR dataset; the output is a decision indicating whether a target of interest is present.

gorithm. The one-class version of this classifier is implemented in the following way. For a particular target of interest, the classifier has a database of stored reference images, each formed by using a different radar viewing direction. The reference image whose associated radar viewing direction best approximates that of the incoming test image (i.e., the incoming region of interest) is called up from the database and is correlated with the test image to generate a match score. If this score exceeds a predetermined threshold, the classifier declares that a target of interest is present.

The template-matching algorithm is attractive because it is readily implemented on a computer and it has an intuitively pleasing structure. For a database formed by using a typical imaging configuration, however, the classifier produces poor results for some target types. This fact is not surprising, because the system was originally designed to process images formed with a fixed SAR geometry, whereas in the most commonly used imaging configurations the SAR geometry is continually changing. In this article, we seek to generalize the structure of the classifier to account for variability in SAR geometry.

SAR geometry can be characterized as a function of two parameters—radar viewing direction and radar motion direction. The baseline classifier does not account for radar motion direction, however, and is therefore equipped with an incomplete set of reference images in its database. The baseline classifier was designed with the assumption that the radar viewing direction is the only parameter that can be varied to produce different images of a target.

In reality, the direction of radar motion is an equally important parameter in defining the SAR imaging geometry, which implies that two images formed with the same radar viewing direction, but with different radar motion directions, will look different. Even though the same physical target scatterers are illuminated in both cases, the 3-D scatterer positions become mapped to two different 2-D SAR image locations. Because the baseline classifier ignores the direction-of-motion parameter, it often correlates a test image and a reference image that are formed with different SAR imaging geometries. These differences in imaging geometry cause the test image and reference image to have dissimilar characteristics, and con-

sequently to have a low correlation score.

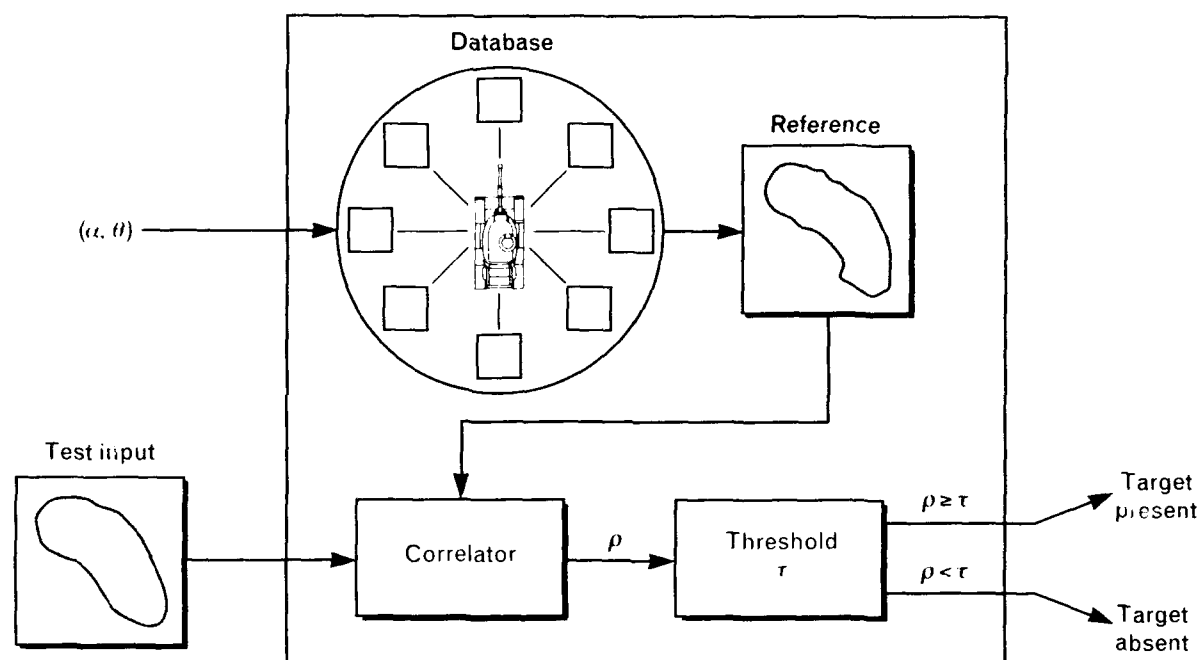
An immediate solution to this problem is to include in the classifier database additional reference images formed by using the SAR geometries that are not currently represented. This solution is undesirable, however, because it would require a costly data collection, and it would also increase the storage requirements for the database by roughly an order of magnitude.

In this article we describe a more elegant solution for improving classifier performance than the mere tenfold augmentation of the reference set described above. This new solution, which maintains the traditional template-matching engine, calls for two major modifications to the baseline classifier: (1) the replacement of the present set of 2-D reference images with a set of 3-D templates, and (2) the incorporation of a mathematical model of the SAR imaging process so that any 3-D template can be appropriately transformed to synthesize a 2-D reference image for the correlation operation. Later in this article we describe a novel method for creating 3-D templates from currently existing 2-D target images.

The body of the article is divided into three major sections. In the first section we describe in detail how the baseline classifier works. In the second section we demonstrate the problem with this classifier and explain why this problem exists. In the third section we describe specifically how we can modify the baseline classifier to improve its overall performance. Finally, we summarize the key points of the article and suggest directions for future work toward improving classification performance in a SAR ATR system.

### **How the Baseline Classifier Works**

The algorithm used by the baseline classifier is described schematically in Figure 3. As shown in this figure, the input to the classifier consists of two components. The first input component is a 2-D test image representing a region of interest from the original SAR dataset. As mentioned above, this image has passed through the first two stages of the ATR system (i.e., the detection and discrimination stages) and thus contains an object that appears sufficiently targetlike to be considered for classification. The second input component is a pair of angle values that



**FIGURE 3.** Schematic description of the algorithm used by the baseline one-class classifier. The classifier uses the aspect angle  $\alpha$  and the depression angle  $\theta$  to select a 2-D reference image from the database. This reference image is then correlated with the input test image; if the correlation score  $\rho$  is greater than or equal to the threshold  $\tau$ , the *target present* decision is declared.

define the radar viewing direction with respect to the imaged object. These values are estimates of the angles  $\alpha$  (the *aspect*) and  $\theta$  (the *depression*), which are defined pictorially in Figure 4.

Because the database is conveniently indexed according to these two radar viewing angles, the classifier can readily select the reference image whose aspect and depression are closest to the input estimates of  $\alpha$  and  $\theta$  computed for the test image. Once the appropriate reference image is selected, it is scaled so that the sum of the squares of its pixel values is equal to unity; the test image is also scaled in this way. Next, the normalized test and reference images are correlated to yield a correlation score  $\rho$  whose value is between 0 and 1.

This correlation operation is mathematically defined in the following way. Let us assume that the test and reference images are equal in size, each having  $M$  cells in the range dimension and  $N$  cells in the cross-range dimension. Let the function  $T(\cdot, \cdot)$  be defined

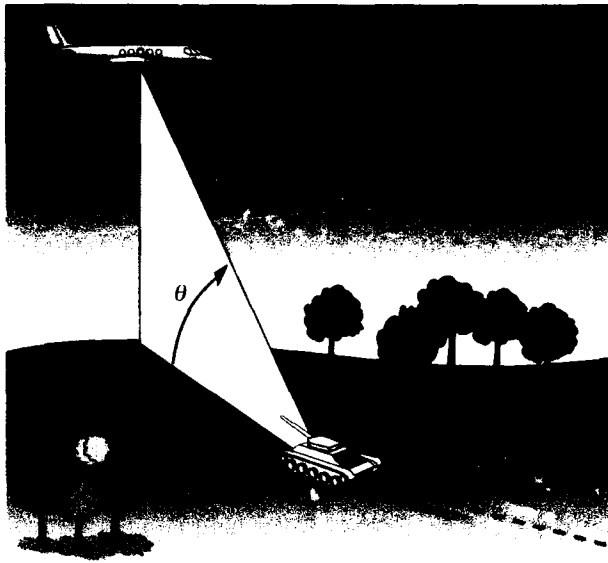
for integer values of its arguments such that  $T(m, n)$  is equal to the amplitude of the test image at the range and cross-range location  $(m, n)$  for  $1 \leq m \leq M$  and  $1 \leq n \leq N$ , and is equal to zero for all other values of  $m$  and  $n$ . Let the function  $R(\cdot, \cdot)$  be defined analogously with respect to the reference image. Then the correlation score  $\rho$  for the two images is defined by

$$\rho = \max_{i,j} \left\{ \frac{1}{s} \sum_{m=1}^M \sum_{n=1}^N T(m, n) R(i + m, j + n) \right\},$$

where  $s$  is the overall normalization factor given by

$$s = \sqrt{\sum_{m=1}^M \sum_{n=1}^N [R(m, n)]^2} \sqrt{\sum_{m=1}^M \sum_{n=1}^N [T(m, n)]^2}.$$

As shown in Figure 3, the classifier declares that a target is present in the test image only if  $\rho$  is greater than or equal to the prespecified threshold  $\tau$ .



**FIGURE 4.** Pictorial definition of the aspect angle  $\alpha$  and the depression angle  $\theta$ . These angles specify the radar viewing direction with respect to the imaged object.

We can more clearly understand the fundamental problem with the baseline classifier by analyzing how the target reference images are generated for the classifier database. Each reference image is formed from data collected by the Lincoln Laboratory millimeter-wave airborne radar [1]. Once a target of interest is deployed in an open area, the data are collected by using a special mode of the radar known as *spotlight* mode, which is illustrated in Figure 5. In this mode,



**FIGURE 5.** Imaging configuration for spotlight-mode SAR. In this mode the airplane moves in a straight line at a constant altitude, while the antenna is steered continuously so that it always points at a fixed patch of terrain.

the airplane flies in a straight line at constant altitude, and the radar antenna is steered continuously so that it always points in the direction of the target.

With the radar beam illuminating the target like a spotlight throughout the flight, a new image of the target can be formed approximately every degree of azimuth. Each new image can be used as a reference for the classifier database. A set of reference images representing  $360^\circ$  of aspect coverage is created by flying four such linear paths to view the target from all sides, as shown in Figure 6.

Although spotlight mode is not the only mode that can be used to generate reference images, it is the most convenient and efficient mode for imaging a target at a variety of radar viewing angles. To see why this statement is true, consider the database of spotlight imagery that can be generated by flying the basic pattern shown in Figure 6 at a sequence of increasing altitudes. Clearly, if the difference between successive flight-path altitudes is small enough, then the database will contain a representative image of the target that is close to any desired aspect-angle and depression-angle pair. Moreover, this complete coverage is obtained without ever having to move the target. In spite of the many advantages to using this kind of data-collection procedure, there is a serious deficiency associated with it. This deficiency is analyzed in detail in the next section.

### Why the Baseline Classifier Needs Improvement

Now that we have discussed the method used to generate target reference images for the database, we are better equipped to analyze why the baseline classifier can make a gross error in categorizing an input test image. In this section we explain how such misclassifications occur, even though the database is densely populated with target reference images from all desired radar viewing directions.

We begin by using Figure 7 to demonstrate what is wrong with the baseline classifier. Figure 7(a) shows an optical photograph of an M48 tank, and Figures 7(b), 7(c), and 7(d) show three simulated SAR images of the tank. The SAR images are color coded with a scale that makes a gradual transition from black (low intensity) to green (medium intensity) to white (high intensity). In each SAR image, the front part of the



tank is pointing toward the upper left corner of the image. All three SAR images shown in this figure were formed by using the same aspect angle and depression angle. In other words, the same scatterers on the target were illuminated by the radar from the same viewing direction for each image. Note, however, that the images look dramatically different. This phenomenon contradicts the key design assumption that fixing the radar viewing direction uniquely specifies the SAR image of the target.

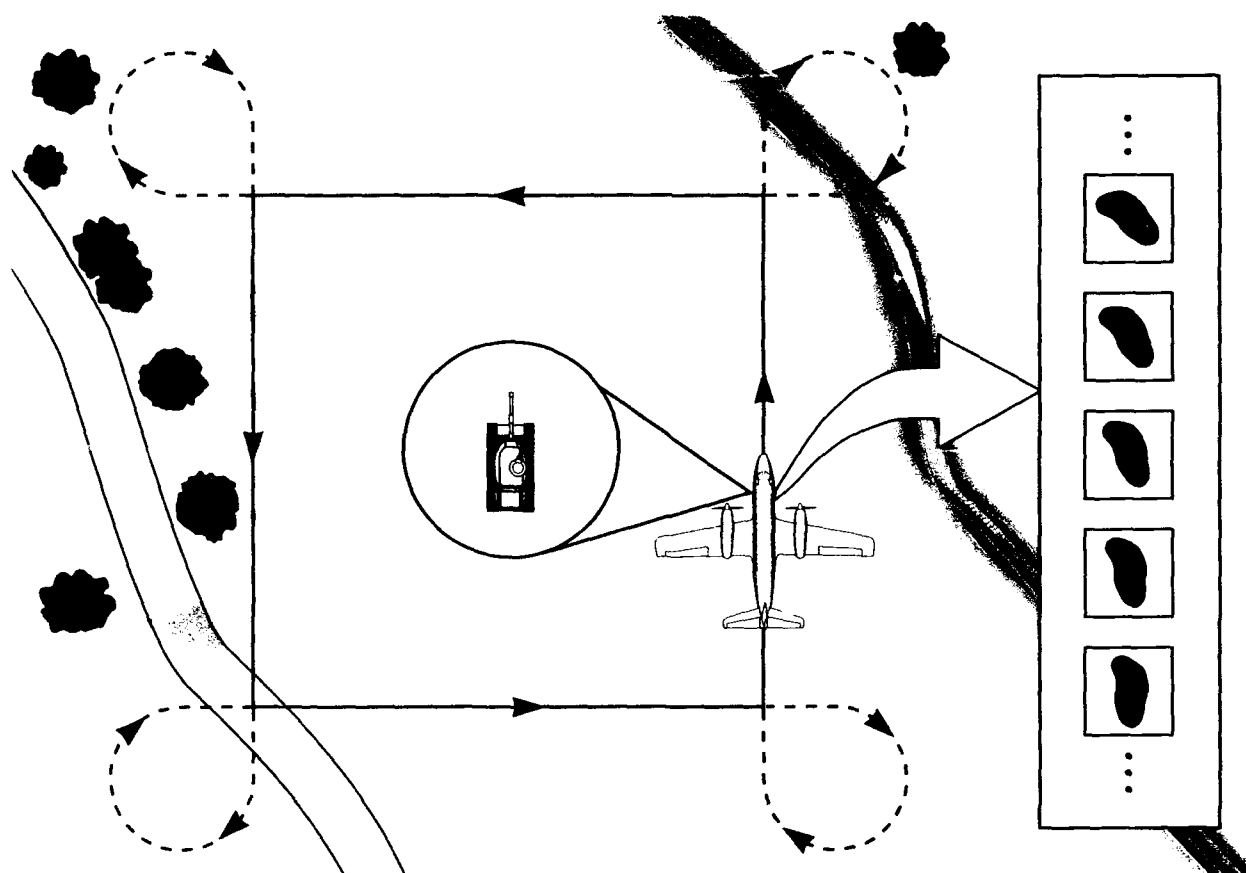
To understand how the existence of three such images affects the classifier, let us assume that Figure 7(b) is a stored reference image and that Figure 7(d) is an incoming test image. Because the two images were formed by using exactly the same radar viewing directions, the image in Figure 7(b) would be chosen as the reference image most likely to match the test image. But because the two images are so dissimilar, their correlation score would be low, and consequently the

test image could be erroneously labeled as containing no target.

The only difference between the SAR imaging configurations used to generate the images in Figures 7(b), 7(c), and 7(d) was the direction in which the radar was moving with respect to the viewing direction. This change alone is sufficient to yield SAR images that look quite different, and yet the direction-of-motion parameter has been completely ignored in the design of the baseline classifier.

To see how this parameter directly affects the appearance of a SAR image, we devote much of this section to the description and application of a widely used mathematical model of the SAR imaging process. In particular, we model the SAR transformation as a *projection* of the 3-D distribution of target scatterers onto a 2-D image plane, and we demonstrate the usefulness of this model by a simple example.

The projection model is conceptually important



**FIGURE 6.** Top view of the flight path used to create a set of target reference images representing 360° of aspect coverage. A sequence of the generated reference images is shown notionally at right.



(a)

 $\phi = -40^\circ$ 

(b)

 $\phi = 0^\circ$ 

(c)

 $\phi = +40^\circ$ 

(d)

**FIGURE 7.** (a) Photograph of an M48 tank. The computer-generated images in (b), (c), and (d) are simulated SAR images of an M48 tank, all created with the same aspect and depression angles but with different squint angles  $\phi$ , where  $\phi$  specifies the direction in which the radar is moving with respect to the viewing direction. The SAR images are color coded with a scale that makes a gradual transition from black (low intensity) to green (medium intensity) to white (high intensity). In each SAR image, the front part of the tank is pointing toward the upper left corner of the image. (The SAR images were generated with the SARTOOL signature simulation software developed by The Analytical Sciences Corporation.)

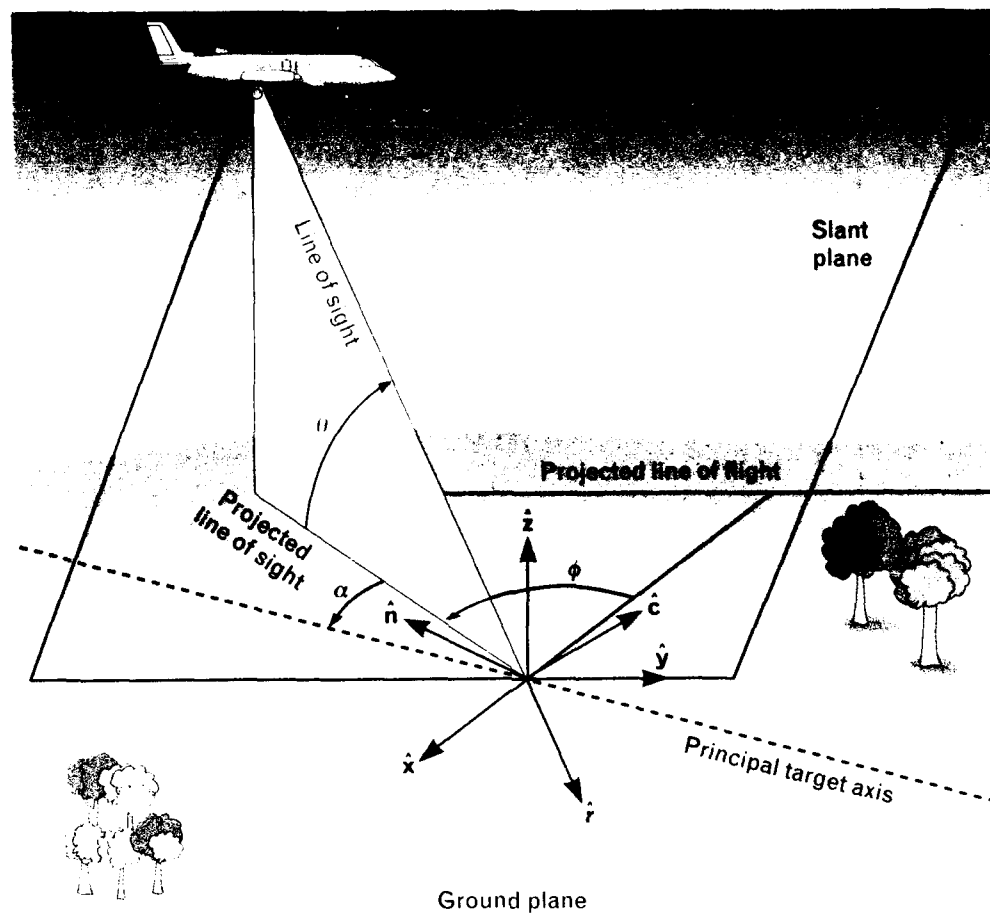
for the remainder of the article, particularly in the final section in which we incorporate this model into an improved version of the baseline classifier. We now prepare to introduce the projection model with some fundamental definitions associated with the SAR imaging process.

#### *Description of SAR Imaging Geometry*

Figure 8 illustrates the basic elements that define the spotlight SAR imaging geometry. In this figure, we see the airborne radar as it moves in a straight line while its antenna is steered to illuminate a fixed ground

location known as the *aimpoint*. We have imposed a mathematical structure on this geometry by using a Cartesian coordinate system (known as the *world coordinate system*) whose origin coincides with the aimpoint, and whose coordinate locations ( $x$ ,  $y$ ,  $z$ ) are measured in terms of the unit basis vectors  $\hat{x}$ ,  $\hat{y}$ , and  $\hat{z}$  (shown in blue). In this system, we use the convention that  $\hat{y}$  points in the direction of radar motion. Also, note that in the vicinity of the aimpoint we model the local earth surface as a *ground plane* defined by the equation  $z = 0$ .

The line that passes through the radar position and



**FIGURE 8.** Illustration of the basic elements that define the spotlight SAR imaging geometry. Two coordinate systems are represented that share a common origin coinciding with the aimpoint. The world coordinate system (in blue) is defined by the unit vectors  $\hat{x}$ ,  $\hat{y}$ , and  $\hat{z}$ . The radar coordinate system (in red) is defined by the unit vectors  $\hat{r}$ ,  $\hat{c}$ , and  $\hat{n}$ .

coincides with the direction of radar motion is called the *line of flight*, while the line that intersects both the radar position and the radar aimpoint is called the *line of sight*. The plane defined by these two lines is called the *slant plane*; the projections of these two lines downward onto the ground plane are called, respectively, the *projected line of flight* and the *projected line of sight*.

Throughout our discussion, we often refer to the three angles  $\alpha$ ,  $\theta$ , and  $\phi$ , which are also shown in Figure 8. The angles  $\alpha$  and  $\theta$  were introduced earlier in the text in Figure 4, but we define them more precisely here. The aspect  $\alpha$  is the angle between the principal target axis in the ground plane and the projected radar line of sight. The depression  $\theta$  is the angle between the radar line of sight and the ground

plane. The angle  $\phi$ , which is known as the *squint*, is defined as the geometric complement of the angle between the projected line of flight and the projected line of sight. Note that the two angles  $\alpha$  and  $\theta$  determine the radar viewing direction, whereas  $\phi$  determines the direction of radar motion relative to the viewing direction.

Each of these definitions is useful as we develop our mathematical abstraction of the SAR imaging process. Specifically, we use the above definitions to describe the SAR transformation as a projection of a 3-D distribution of target scatterers onto the 2-D slant plane. In the appendix we give a detailed justification for representing the SAR transformation this way, based on the actual physical quantities measured by a SAR. In the next subsection, however, we omit

this justification and merely provide a concise mathematical description of our SAR imaging model. Following this description, we give an application of our SAR imaging model in the form of a simple visual example.

### A Mathematical Model for SAR Imaging

We begin our description of the SAR transformation by introducing the *radar coordinate system* (shown in red) in Figure 8. The origin of this Cartesian coordinate system coincides with the aimpoint, and the coordinate locations are represented in terms of the unit basis vectors  $\hat{\mathbf{r}}$  (the *range vector*),  $\hat{\mathbf{c}}$  (the *cross-range vector*), and  $\hat{\mathbf{n}}$  (the *slant-plane normal vector*). This coordinate system can be defined in terms of the basic elements of the spotlight SAR geometry defined above.

We begin with the observation that the range vector  $\hat{\mathbf{r}}$ , which points in the direction of the radar line of sight, can be expressed in world coordinates as

$$\hat{\mathbf{r}} = \begin{bmatrix} \cos \phi \cos \theta \\ \sin \phi \cos \theta \\ -\sin \theta \end{bmatrix}.$$

To check that this expression is correct, the reader can easily verify the following three properties of  $\hat{\mathbf{r}}$ : (1)  $\hat{\mathbf{r}}$  is unit length, (2) the projection of  $\hat{\mathbf{r}}$  onto the  $x$ - $y$  plane is rotated counterclockwise by the angle  $\phi$  with respect to the  $x$ -axis, and (3)  $\hat{\mathbf{r}}$  is tilted downward by the angle  $\theta$  with respect to the  $x$ - $y$  plane.

By using the vector  $\hat{\mathbf{r}}$  and the world-coordinate basis vector  $\hat{\mathbf{y}} = [0 \ 1 \ 0]^T$ , both of which lie in the slant plane, we can construct the slant-plane normal vector  $\hat{\mathbf{n}}$  with the cross-product formula

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{r}} \times \hat{\mathbf{y}}}{\|\hat{\mathbf{r}} \times \hat{\mathbf{y}}\|} = k \begin{bmatrix} \sin \theta \\ 0 \\ \cos \phi \cos \theta \end{bmatrix},$$

where  $k$  is the normalizing constant required to make  $\hat{\mathbf{n}}$  a unit-length vector. The value of  $k$  is given by

$$k = \frac{1}{\sqrt{\sin^2 \theta + \cos^2 \phi \cos^2 \theta}}. \quad (1)$$

Because the cross-range vector  $\hat{\mathbf{c}}$  must be perpendicular to both  $\hat{\mathbf{r}}$  and  $\hat{\mathbf{n}}$ , it is constructed by using the formula

$$\hat{\mathbf{c}} = \hat{\mathbf{n}} \times \hat{\mathbf{r}} = k \begin{bmatrix} -\sin \phi \cos \phi \cos^2 \theta \\ \cos^2 \phi \cos^2 \theta + \sin^2 \theta \\ \sin \phi \cos \theta \sin \theta \end{bmatrix}.$$

From this coordinate-system construction, we see that the vectors  $\hat{\mathbf{r}}$  and  $\hat{\mathbf{c}}$  form an orthonormal basis for the slant plane, just as the vectors  $\hat{\mathbf{x}}$  and  $\hat{\mathbf{y}}$  form an orthonormal basis for the ground plane.

Now we consider imaging a point reflector at location  $\mathbf{p} = [p_x \ p_y \ p_z]^T$  in the world coordinate system. We can express this point in the radar coordinate system by using the standard dot product to project the point  $\mathbf{p}$  onto each of the unit basis vectors  $\hat{\mathbf{r}}$ ,  $\hat{\mathbf{c}}$ , and  $\hat{\mathbf{n}}$ . The resulting vector  $\mathbf{q}$  can be written in radar coordinates as

$$\mathbf{q} = \begin{bmatrix} q_r \\ q_c \\ q_n \end{bmatrix} = \begin{bmatrix} \mathbf{p} \cdot \hat{\mathbf{r}} \\ \mathbf{p} \cdot \hat{\mathbf{c}} \\ \mathbf{p} \cdot \hat{\mathbf{n}} \end{bmatrix}.$$

According to our basic model for the SAR transformation, we must now project the 3-D vector  $\mathbf{q}$  onto the 2-D slant plane to obtain its location in the SAR image. We can do this projection by retaining the first two components of  $\mathbf{q}$  and neglecting the third component, because the entire third dimension of the radar coordinate system becomes collapsed in the projection process. This procedure gives the slant-plane coordinates of the original point reflector as

$$\begin{bmatrix} q_r \\ q_c \end{bmatrix} = \begin{bmatrix} p_x \cos \phi \cos \theta + p_y \sin \phi \cos \theta - p_z \sin \theta \\ -kp_x \sin \phi \cos \phi \cos^2 \theta \\ + kp_y (\cos^2 \phi \cos^2 \theta + \sin^2 \theta) \\ + kp_z \sin \phi \cos \theta \sin \theta \end{bmatrix}. \quad (2)$$

We can use the above expression for the range and cross-range coordinates of a point to show math-

ematically that when the point is imaged at a fixed aspect angle and depression angle, but at different squint angles, it will appear at different SAR image locations.

To demonstrate this concept, we conduct two imaging experiments in which we keep the aspect and depression angles constant but allow the squint angle to vary. In particular, for the first imaging experiment we use the values  $\alpha = 0$ ,  $\theta = \pi/4$ , and  $\phi = 0$ ; for the second experiment we use the values  $\alpha = 0$ ,  $\theta = \pi/4$ , and  $\phi = \pi/4$ . In each experiment, once these angles have been fixed, we consider imaging a point  $\mathbf{p}$  that lies on the principal target axis in the ground plane at a distance of one unit from the aimpoint. Based on the diagram in Figure 8, this point must have the coordinates

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos \phi \\ \sin \phi \\ 0 \end{bmatrix}. \quad (3)$$

For the first imaging experiment (with the angles  $\alpha = 0$ ,  $\theta = \pi/4$ ,  $\phi = 0$ ), we can readily verify from Equation 1 that  $k = 1$ , and from Equation 3 that

$$\mathbf{p} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

We can now compute the slant-plane coordinates of  $\mathbf{p}$  by substituting these numerical values into Equation 2. This computation yields the 2-D slant-plane location  $\mathbf{s}_1$  associated with the first set of imaging angles; this location is given by

$$\mathbf{s}_1 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}.$$

For the second imaging experiment (with the angles  $\alpha = 0$ ,  $\theta = \pi/4$ ,  $\phi = \pi/4$ ), we find that  $k = 2/\sqrt{3}$ , and

$$\mathbf{p} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}.$$

As before, we compute the slant-plane coordinates of  $\mathbf{p}$  by substituting these values into Equation 2. This yields the 2-D slant-plane location  $\mathbf{s}_2$ , given by

$$\mathbf{s}_2 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} \end{bmatrix}.$$

For the two different squint angles above, the range coordinate of the imaged point remains constant but the cross-range coordinate changes dramatically. This observation is an example of the more general result that, given a fixed radar viewing direction, a change in the squint angle causes the cross-range coordinate of a point to change. Thus the above example provides quantitative proof that the slant-plane location of a point is not uniquely determined by the aspect and depression angles alone.

In the next subsection, we give a simple qualitative example that visually demonstrates the effects of the squint angle on the appearance of a SAR image, and thus demonstrates the importance of incorporating information about the squint angle into the baseline classification algorithm.

### SAR Imaging Example

Figure 9(a) shows a perspective view of a simple object that is being imaged by an airborne SAR. The object consists of a square grid of point reflectors (shown in blue) in the ground plane, and one additional point reflector (shown in red) above the ground plane and directly over the center of the grid. Figure 9(b) shows a top view of the same imaging configuration. From this top view, we can see that the grid of point reflectors is perfectly aligned with the projected radar line of sight; we arbitrarily define this orientation to correspond to a  $0^\circ$  aspect angle. The object is also being imaged at a  $0^\circ$  squint angle, because the radar is looking in a direction perpendicular to the line of flight.

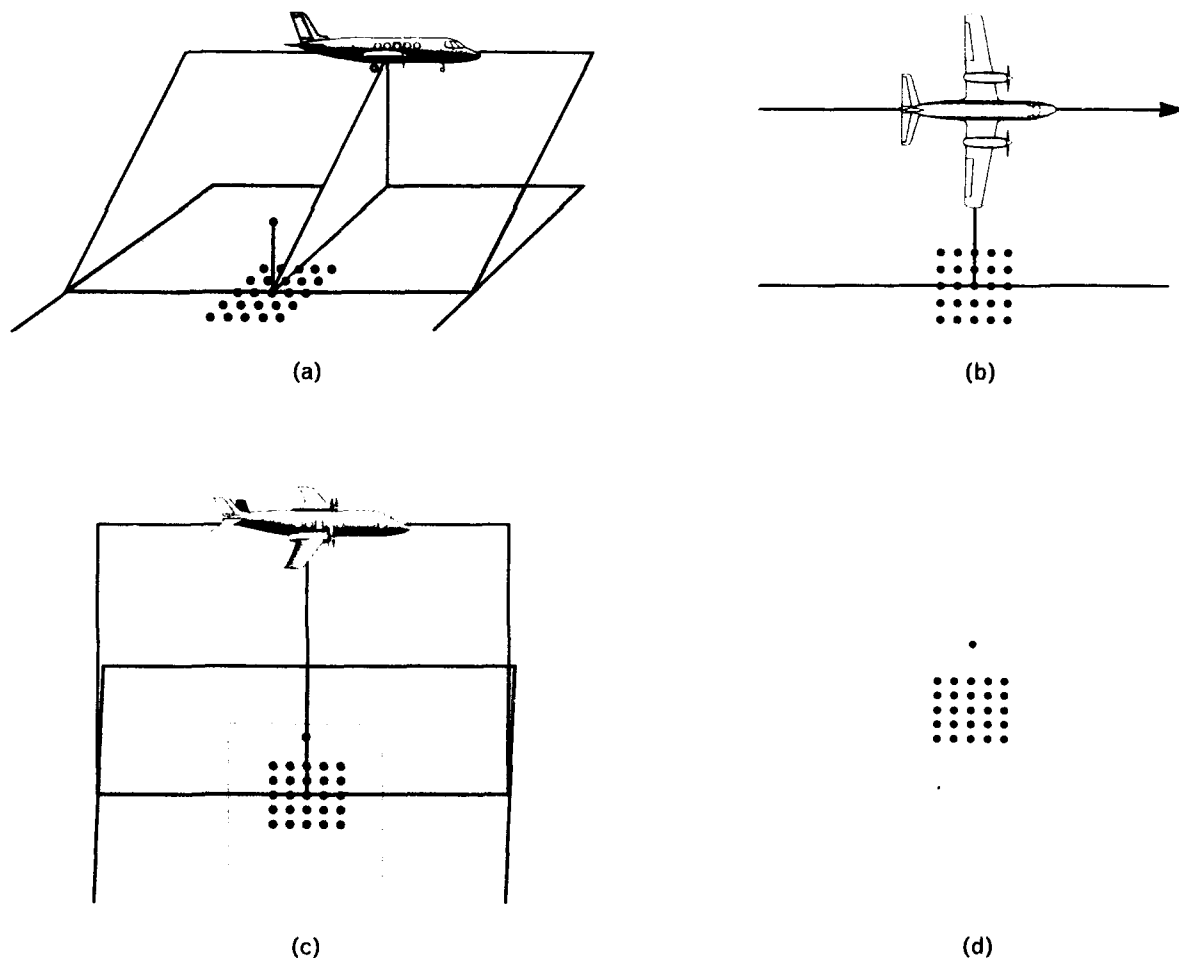
Figure 9(c) shows the same imaging configuration once again, but from a viewing direction perpendicular to the slant plane. Thus we see the projection of the object onto the slant plane, which (according to our mathematical SAR model) corresponds directly to the result produced by the SAR imaging process.

Because of the projection operation, the grid of point reflectors (in blue) appears foreshortened in the vertical dimension, and the point reflector above the ground (in red) appears just above the grid. Finally, Figure 9(d) shows an image-sized portion of the slant-plane projection (displayed according to the convention that range increases in the downward direction) that represents the SAR image of the object at a  $0^\circ$  aspect angle and a  $0^\circ$  squint angle.

Figure 10(a) shows a perspective view of the same object being imaged with a different SAR geometry. For this example, we assume that the slant plane has been adjusted so that the depression angle matches that of the previous example. From the top view

shown in Figure 10(b) we can see that the aspect angle has not changed (i.e., the grid is still aligned with the projected radar line of sight), but that the squint angle has changed from  $0^\circ$  to  $45^\circ$ .

Figure 10(c) shows the object projected onto the slant-plane under this new imaging configuration. The grid of point reflectors (in blue), which appeared as a diamond from the top view, now appears as a foreshortened diamond in the vertical dimension because of the projection operation; the additional point reflector above the ground plane (in red) appears over the upper corner of this diamond because of its height. Figure 10(d) shows an image-sized portion of the slant-plane projection (again displayed according to



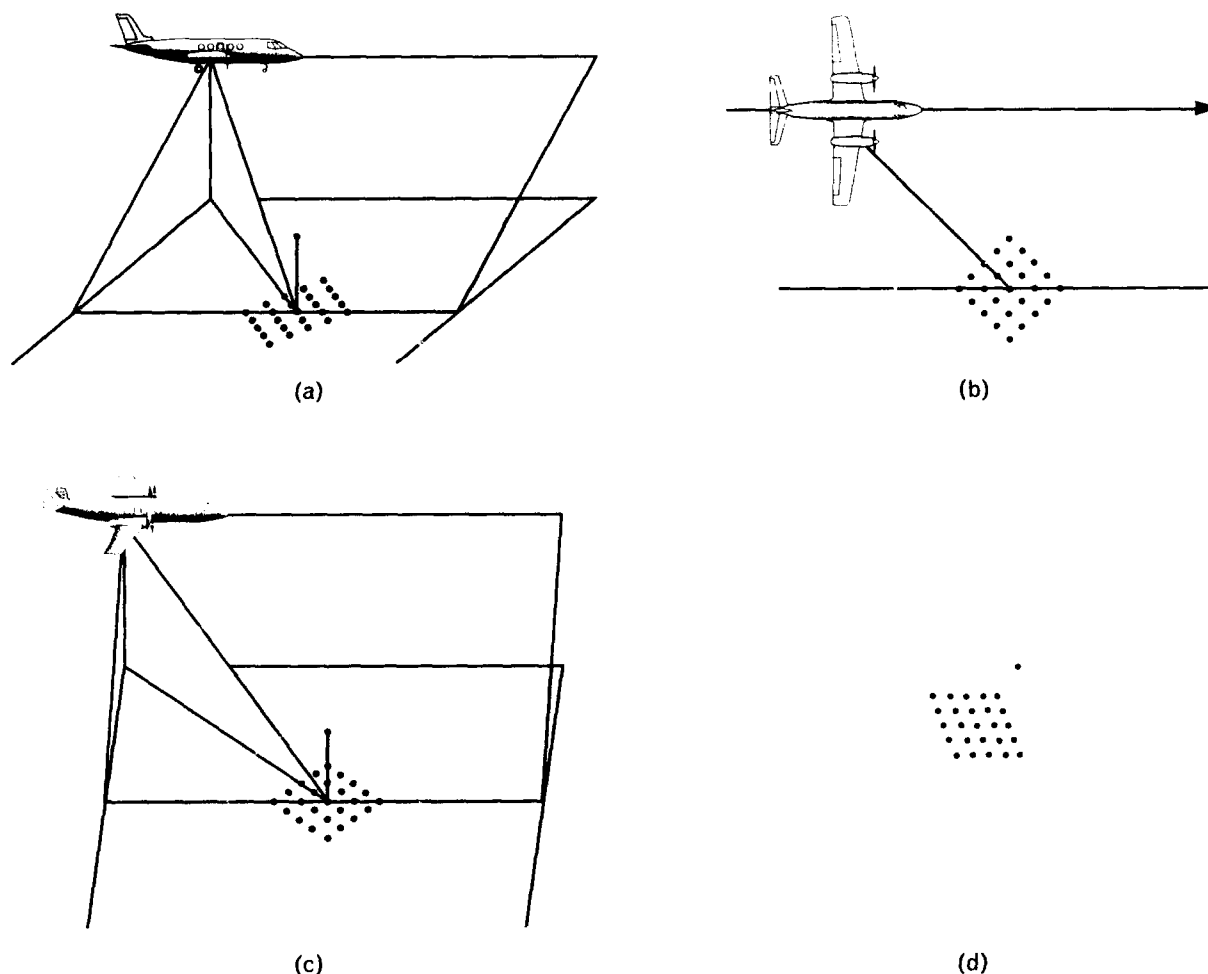
**FIGURE 9.** Illustration of SAR imaging as a projection (broadside case). The collection of point reflectors being imaged is shown from (a) perspective view, (b) top view, and (c) slant-plane view. (d) The resulting SAR image can be interpreted as an image-sized portion of the slant-plane projection, as indicated by the orange outline in part c.

the convention that range increases in the downward direction) that represents the SAR image of the object at a  $0^\circ$  aspect angle and a  $45^\circ$  squint angle.

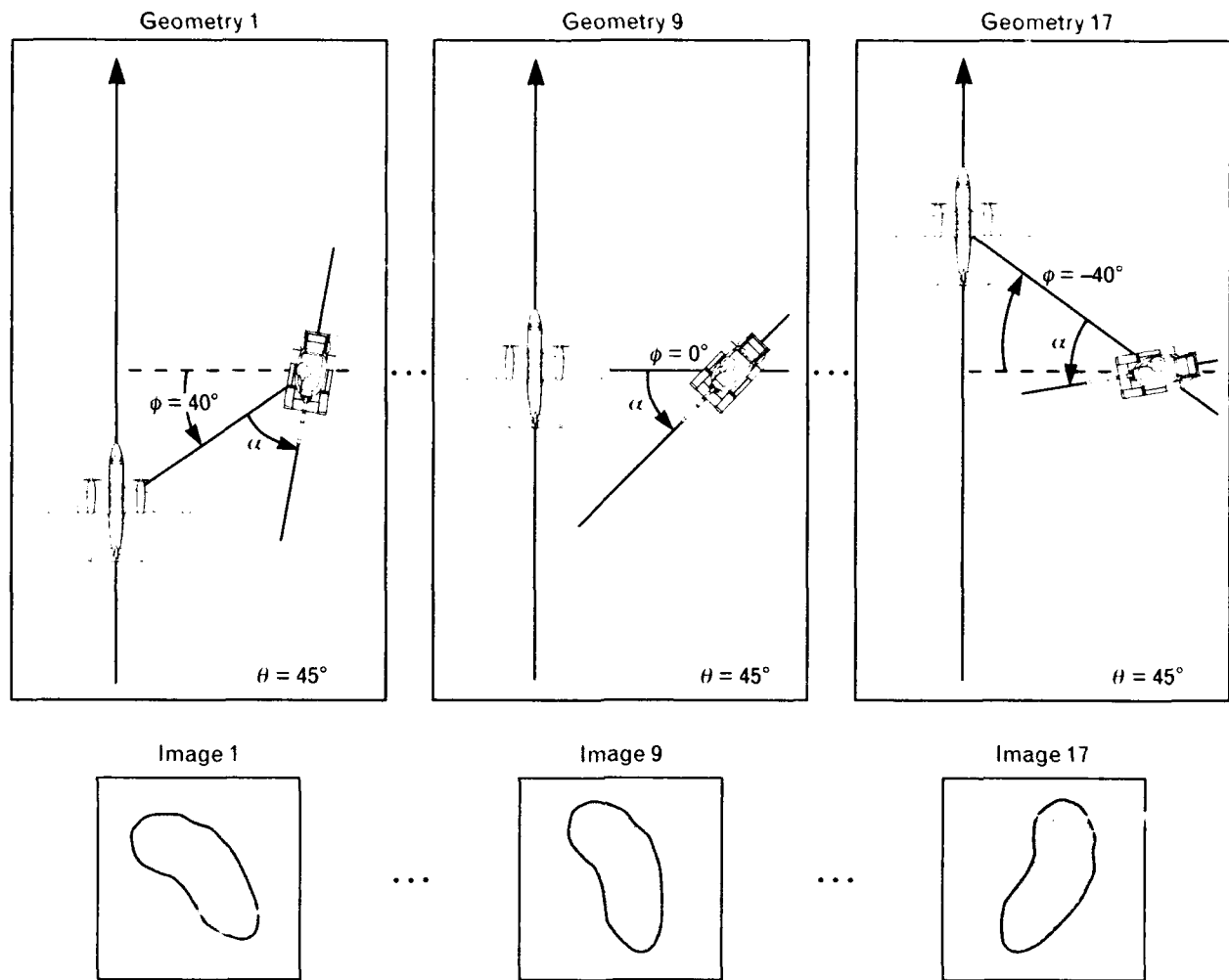
The primary difference between this SAR image and the one shown in Figure 9(d) is that the point reflectors now appear shifted in the cross-range dimension. The shift for each reflector is not, however, a simple function of the range of the reflector, as it may appear at first glance. Rather, the shift is a function of the 3-D location of the reflector, which is demonstrated by the large shift of the point reflector above the ground plane. This shift has caused the reflector to move out of alignment with the middle column of the grid, which can be seen by comparing Figures 9(d) and 10(d).

### *Sensitivity of Baseline Classifier to Changes in Squint*

The simple geometric examples given in Figures 9 and 10 show that we can produce two different images of an object by using two different squint angles for a fixed set of aspect and depression angles. From a qualitative standpoint, these differences adversely affect the performance of the baseline classifier, because the classifier has only one reference image for each aspect-angle and depression-angle pair. In this section, we describe an experiment that demonstrates quantitatively that the classification statistic used by the baseline classifier—the correlation score—changes significantly as a function of squint angle for a fixed aspect-angle and depression-angle pair.



**FIGURE 10.** Illustration of SAR imaging as a projection (forward-looking case). The sequence of figures—(a) perspective view, (b) top view, (c) slant-plane view, and (d) resulting SAR image—corresponds directly to the sequence shown in Figure 9.



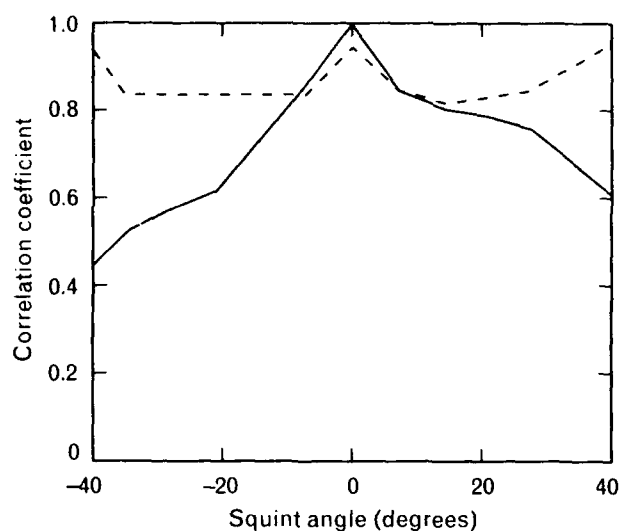
**FIGURE 11.** Depiction of the imaging configurations used to quantify the robustness of the baseline classifier with respect to variations in squint angle. The test images, which are shown notionally below their respective configuration diagrams, were all formed by using the same aspect angle ( $\alpha = 45^\circ$ ) and depression angle ( $\theta = 45^\circ$ ), but each had a unique squint angle  $\phi$  (a multiple of  $5^\circ$  in the range from  $-40^\circ$  to  $+40^\circ$ ).

We conducted the experiment by using a target-signature simulation package from The Analytic Sciences Corporation known as SARTOOL [2], which was designed to model the dominant electromagnetic characteristics of a target. We used the SARTOOL model of an M48 tank oriented such that both the aspect and depression angles were fixed at  $45^\circ$ . We carried out the experiment by using a single reference image, which was created at a  $0^\circ$  squint angle, and 17 test images, which were created at squint angles ranging from  $-40^\circ$  to  $+40^\circ$  in  $5^\circ$  increments. Figure 11 illustrates the imaging configurations we used to generate these test images. (Note that the aspect angle

and the depression angle are the same for each configuration.) The experiment consisted of correlating the reference image with each of the test images to generate a plot of correlation score versus squint angle. This plot is shown as a solid line in Figure 12.

Of course, the classifier gave perfect performance (i.e., correlation score  $\rho = 1.0$ ) for the test image formed at a  $0^\circ$  squint angle, because the reference image was formed at this same squint angle. The correlation scores progressively decline, however, as the squint angle for the test set varies in either direction from  $0^\circ$ . At the extremes of  $-40^\circ$  and  $+40^\circ$ , the correlation scores are approximately 0.5. This result





**FIGURE 12.** Plots of correlation score versus squint angle for the baseline classifier (solid line) and the proposed classifier (dashed line). The test images used for the experiments were those described in Figure 11. The 2-D template used by the baseline classifier and the 3-D template used by the proposed classifier both had aspect and depression angles matching those of the test images.

suggests that the performance of the baseline classifier is sensitive to changes in squint angle. Thus, to improve the baseline classifier we must account for the effects of squint (in addition to the already recognized effects of aspect and depression) on the process of SAR image formation.

### How to Develop a Better Classifier

Our analysis in the previous section suggests that we can improve the performance of the baseline classifier by taking into account the effects of both radar viewing direction and radar motion direction on the SAR imaging process. In this section we propose a new classifier that maintains the conventional template-matching engine, but calls for two major modifications to the baseline classifier: (1) the replacement of the present set of 2-D reference images with a set of 3-D templates, and (2) the incorporation of our mathematical model of SAR imaging as a projection so that any 3-D template can be transformed appropriately to synthesize a 2-D reference image for the correlation operation.

We begin by giving a definition of a 3-D template, and we then describe how a 3-D template is transformed into a 2-D reference image. In addition, we

present a novel technique for creating 3-D templates from our existing database of 2-D reference images. Finally, in a continuation of the experiment discussed in the previous section, we show that the proposed classifier is much more robust with respect to changes in squint angle than the baseline classifier.

### Description of 3-D Templates

A 3-D template is a finely sampled 3-D grid of points representing the volume occupied by the target of interest, in which each grid point corresponds to a scatterer on the target. Each 3-D template is associated with a distinct radar viewing direction, specified by the aspect angle  $\alpha$  and the depression angle  $\theta$ . We thus index the collection of 3-D templates by these radar viewing angles, and we let  $T_{\alpha\theta}$  denote the template corresponding to the particular pair  $(\alpha, \theta)$ . The value stored at each point in the template  $T_{\alpha\theta}$  represents the radar reflectivity of the scatterer at that point, when the target is illuminated from the direction corresponding to  $(\alpha, \theta)$ . To prepare for the development that follows, we assume that the template  $T_{\alpha\theta}$  contains  $K$  grid points. The location of the  $j$ th point in the 3-D grid is denoted by  $\mathbf{p}_j$ , and the radar-reflectivity value stored at this point is denoted by  $A_j$ , for  $j = 1, \dots, K$ .

To transform  $T_{\alpha\theta}$  into a 2-D reference image, we use our projection model of the SAR imaging process. Specifically, we project the points in the template  $T_{\alpha\theta}$  onto the slant plane defined by the depression angle  $\theta$  and the squint angle  $\phi$ , to yield a reference image that we denote by  $I_{\alpha\theta\phi}$ . Let  $I_{\alpha\theta\phi}(m, n)$  be the value at the range/cross-range location  $(m, n)$  in this reference image. The relation between the values in the template  $T_{\alpha\theta}$  and the reference image value  $I_{\alpha\theta\phi}(m, n)$  is given by

$$I_{\alpha\theta\phi}(m, n) = \sum_{j \in Q_{\alpha\theta\phi}(m, n)} A_j,$$

where  $Q_{\alpha\theta\phi}(m, n)$  is the set of indices specified by

$$Q_{\alpha\theta\phi}(m, n) = \left\{ j \left| \begin{array}{l} \mathbf{p}_j \text{ projects to location} \\ (m, n) \text{ in the SAR image} \\ \text{corresponding to } (\alpha, \theta, \phi) \end{array} \right. \right\}.$$

Because a SAR image is composed of discrete pixels in the range and cross-range dimensions, the location  $(m, n)$  actually corresponds to a locus of points in the slant plane. Let  $\Delta_r$  and  $\Delta_c$  be the range and cross-range pixel spacing intervals, respectively, associated with the image. Then any slant plane location  $(q_r, q_c)$  such that

$$\begin{aligned} r_m &\leq q_r < r_m + \Delta_r \\ c_n &\leq q_c < c_n + \Delta_c, \end{aligned}$$

(where  $r_m$  and  $c_n$  are appropriate constants) is mapped to SAR image location  $(m, n)$ . Thus  $\mathbf{p}_j$  projects to the SAR image location  $(m, n)$  if

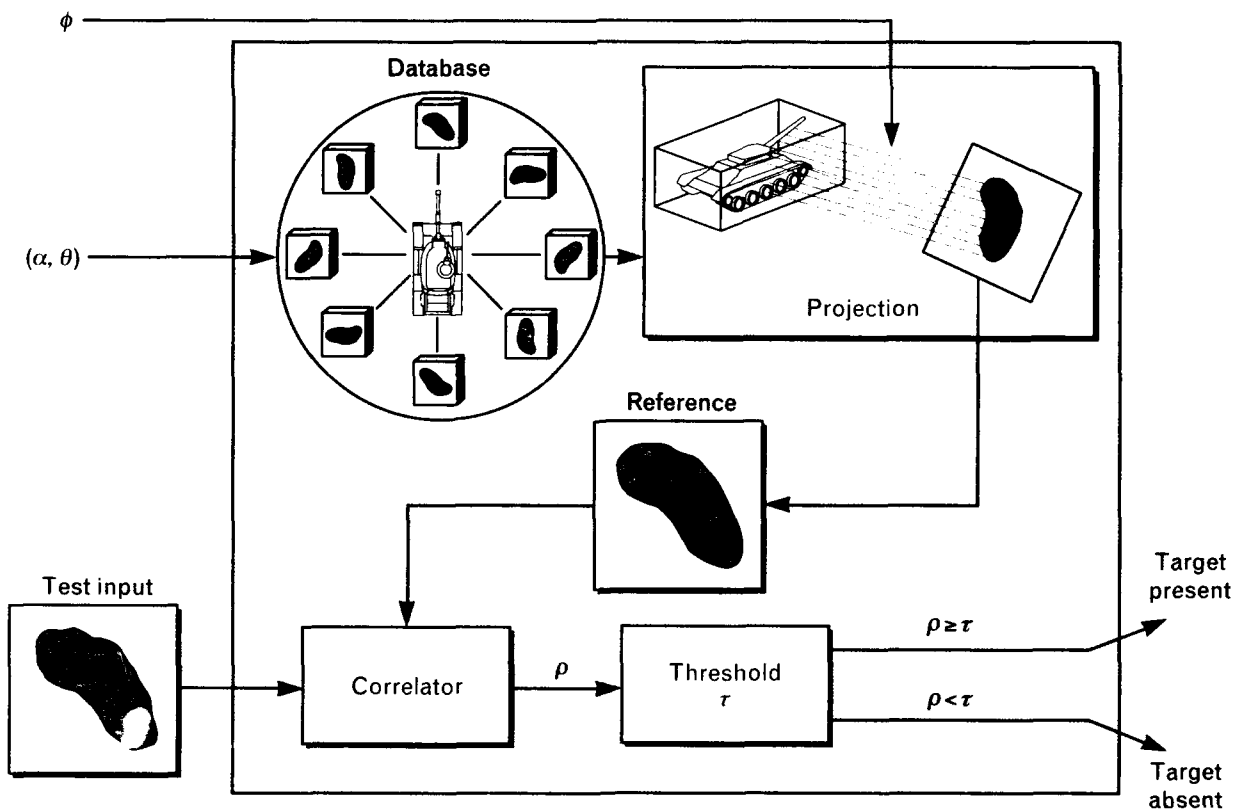
$$\begin{aligned} r_m &\leq \mathbf{p}_j \cdot \hat{\mathbf{r}} < r_m + \Delta_r \\ c_n &\leq \mathbf{p}_j \cdot \hat{\mathbf{c}} < c_n + \Delta_c, \end{aligned}$$

where  $\hat{\mathbf{r}}$  and  $\hat{\mathbf{c}}$  are the unit range and cross-range vectors in the radar coordinate system that was defined earlier.

### Description of Proposed Classifier

Figure 13 illustrates the algorithm used by the proposed version of the baseline classifier (note the similarity between this figure and Figure 3). The classifier uses the aspect angle  $\alpha$  and the depression angle  $\theta$  to select a 3-D template from the database. The points in this 3-D template are projected onto the slant plane specified by the squint angle  $\phi$  to produce a 2-D image, which is then correlated with the input test image. If the correlation score exceeds the threshold  $\tau$ , the target of interest is declared to be present.

Note that the new classifier continues to use the conventional template-matching engine, so that the overall structure of the algorithm is unchanged. The



**FIGURE 13.** Schematic description of the algorithm used by the proposed one-class classifier. The classifier uses the aspect angle  $\alpha$  and the depression angle  $\theta$  to select a 3-D template from the database. This 3-D template is projected onto the slant plane specified by the squint angle  $\phi$  to produce a 2-D image, which is then correlated with the input test image. If the correlation score  $\rho$  is greater than or equal to the threshold  $\tau$ , the *target present* decision is declared.

principal modification to the proposed classifier is that the original database of 2-D reference images has been replaced by a database of 3-D templates. In addition, the proposed classifier is equipped with a processor that transforms a 3-D template into a 2-D image.

### Creation of 3-D Templates

Let us now consider how a 3-D template can be created from the existing set of 2-D reference images. For a target of interest, we wish to construct a 3-D template corresponding to a particular aspect-angle and depression-angle pair. To perform this construction we require two or more SAR images of the target, all formed by using the fixed aspect and depression angles of the template, but each formed by using a different squint angle. Recall that each of these SAR images represents a projection of the 3-D distribution of target scatterers onto a 2-D slant plane. Our goal is to use the information contained in these projections to reconstruct the locations and amplitudes of the target scatterers.

We fix the locations of the scatterers  $\mathbf{p}_1, \dots, \mathbf{p}_K$  so that they represent a uniform sampling of a parallelepiped that is approximately the size of the target. Once these scatterer locations are fixed, we then solve for the unknown amplitudes  $A_j$  corresponding to the points  $\mathbf{p}_j$ . Mathematically, we formulate the problem of determining the  $A_j$  values in the following way. Let us assume we have  $L$  actual SAR images  $I_1, \dots, I_L$  formed at squint angles  $\phi_1, \dots, \phi_L$ , respectively. Corresponding to this sequence of actual images, we let  $\hat{I}_1, \dots, \hat{I}_L$  be a sequence of synthetic images formed from the template amplitude values. By using our projection model for the SAR imaging process, the value in the  $i$ th synthetic image corresponding to the range/cross-range location  $(m, n)$  is computed by

$$\hat{I}_i(m, n) = \sum_{j \in Q_i(m, n)} A_j, \quad (4)$$

where  $Q_i(m, n)$  is the set of indices specified by

$$Q_i(m, n) = \left\{ j \left| \begin{array}{l} \mathbf{p}_j \text{ projects to location } (m, n) \text{ in the} \\ \text{SAR image corresponding to } \phi_i \end{array} \right. \right\}.$$

Let the total mean-square difference between the set of synthetic images and the set of actual images be given by

$$\varepsilon(A_1, \dots, A_K) = \sum_{i=1}^L \left\{ \sum_{m=1}^M \sum_{n=1}^N [\hat{I}_i(m, n) - I_i(m, n)]^2 \right\}, \quad (5)$$

where  $M$  and  $N$  are the range and cross-range dimensions, respectively, of the SAR images. (Note that the dependency of  $\varepsilon$  on each  $A_j$  enters Equation 5 implicitly through Equation 4.)

We can now cast the template construction problem as a multivariable minimization problem. Specifically, we compute the values of  $A_1, \dots, A_K$  by solving

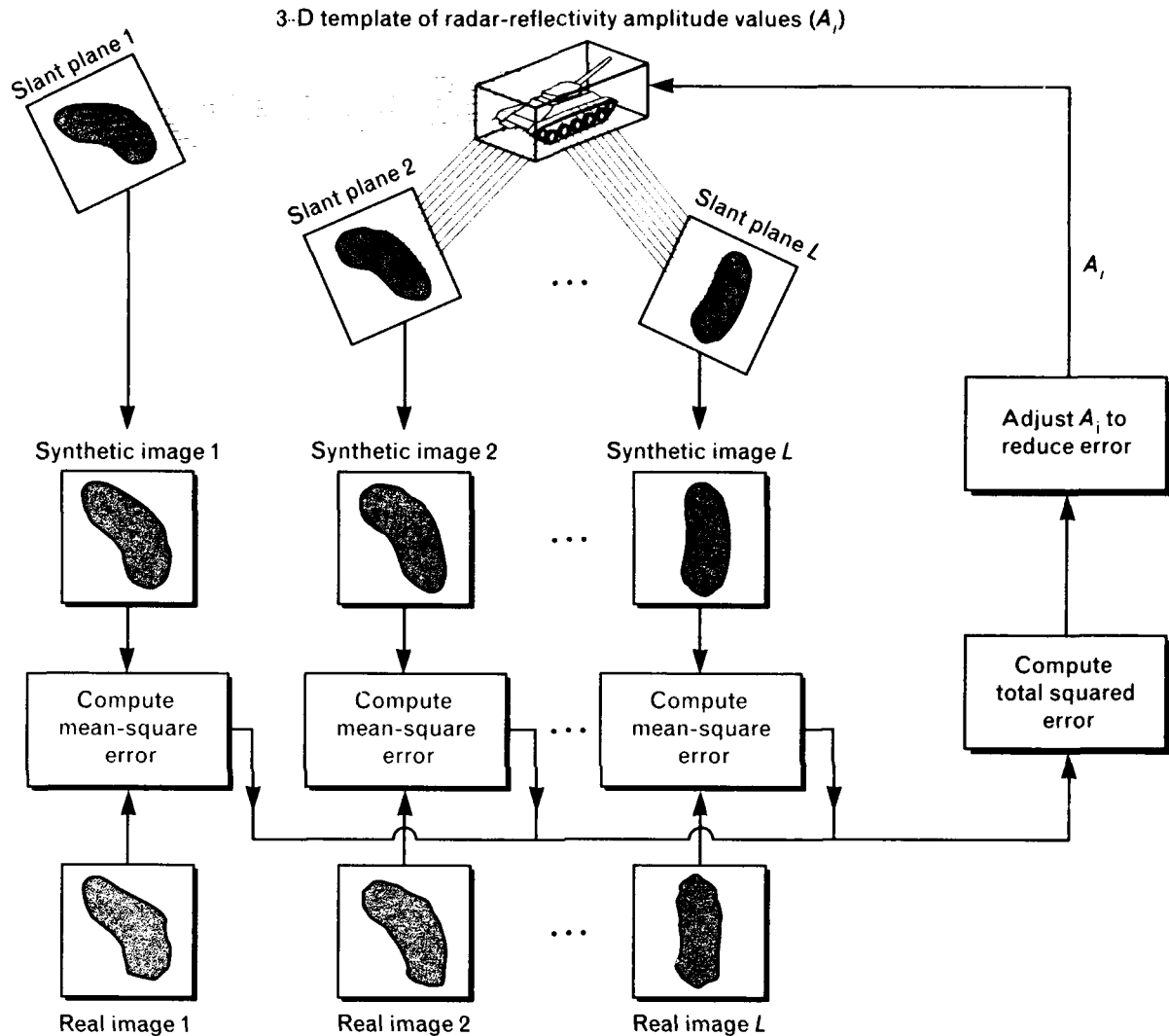
$$\text{Minimize } \varepsilon(A_1, \dots, A_K)$$

$$\text{such that } A_j \geq 0, \text{ for } j = 1, 2, \dots, K.$$

To determine an optimal solution, we begin by assigning to each unknown amplitude  $A_j$  an initial amplitude that represents our best *a priori* estimate of the actual radar reflectivity at that point. In the absence of *a priori* knowledge, we assign a random initial value to each  $A_j$ . Figure 14 illustrates the iterative procedure we use to compute the template amplitude values.

The points  $\mathbf{p}_j$  are projected onto each of the  $L$  slant planes (each slant plane corresponds to an actual SAR image supplied to the algorithm), which results in a sequence of synthetic images that can be compared to the actual images. The total squared error is computed from these two sets of images (synthetic and actual) and the amplitude values are adjusted such that this total error is reduced. The iteration then cycles through the stages of synthetic image formation, error computation, and amplitude adjustment. The procedure is terminated when the total squared error is less than some prespecified tolerance.

Many standard gradient-descent techniques are available for implementing this iterative minimization; for more details on these techniques see the book by D.G. Luenberger [3].



**FIGURE 14.** Illustration of the 3-D template-creation procedure. The procedure begins with a random assignment of radar-reflectivity values at points in the 3-D template; thereafter the procedure becomes an iterative refinement process. The amplitudes are projected onto  $L$  slant planes (each slant plane corresponds to an actual SAR image supplied to the algorithm), which results in a sequence of synthetic images that can be compared to the actual images. The total squared error is computed from these two sets of images (synthetic and actual), and the amplitude values are adjusted such that this total error is reduced. The iteration then cycles through the stages of synthetic image formation, error computation, and amplitude adjustment. The procedure is terminated when the total squared error is less than some prespecified tolerance.

#### *Sensitivity of Proposed Classifier to Changes in Squint*

Earlier we described an experiment with the baseline classifier that provided quantitative proof that the correlation score varies significantly as a function of squint angle for a fixed aspect-angle and depression-angle pair. Because the correlation score is the classification statistic used by the baseline classifier, overall performance is extremely sensitive to changes in squint

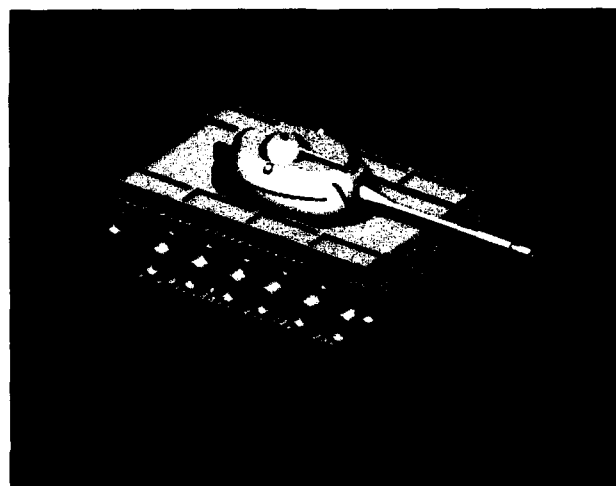
angle. This section summarizes a continuation of the experiment in which we measure the sensitivity of the proposed classifier to changes in squint angle.

For this experiment we used the same set of 17 test images described earlier. Recall that these images were formed by using the SARTOOL model of the M48 tank oriented such that both the aspect angle and depression angle were fixed at  $45^\circ$ . These images were created with squint angles ranging from  $-40^\circ$  to  $+40^\circ$

in  $5^\circ$  increments. Let us denote the  $i$ th image in this set by  $I_i$ , with the corresponding squint angle  $\phi_i$  given by the expression

$$\phi_i = -40 + 5(i - 1), \quad i = 1, \dots, 17.$$

Our 2-D reference image, which was created at a squint angle of  $0^\circ$ , was replaced by a 3-D template



(a)



(b)

**FIGURE 15.** (a) Simple solids model of an M48 tank; (b) the same model of the tank with an overlay of the most significant radar-reflectivity information contained in the 3-D template for the tank. (The overall intensity of the underlying solids model has been reduced to give emphasis to the radar-reflectivity information.) Note that there are significant radar returns from the front right fender, the turret, the track region, and the front left portion of the tank.

corresponding to an aspect angle of  $45^\circ$  and a depression angle of  $45^\circ$ . We constructed this 3-D template by applying the algorithm described in the previous section to three SARTOOL images formed at squint angles of  $-40^\circ$ ,  $0^\circ$ , and  $40^\circ$ . These three images are displayed in Figures 7(b), 7(c), and 7(d), respectively.

The result of this 3-D template construction is interesting to observe. Recall that the value stored at each point in the template is the radar reflectivity of the scatterer at that location, when the target is imaged from the given viewing direction. Figure 15(a) shows a simple solids model that represents the basic features and dimensions of the M48 tank; Figure 15(b) shows the same model overlaid with the most significant radar-reflectivity values contained in the template. Note that there are significant radar returns from the front right fender, as well as from the turret, the track region, and the front left portion of the tank.

In our previous experiment we correlated the reference image with each of the 17 test images to generate a plot (denoted by the solid line in Figure 12) of correlation score versus squint angle. In this experiment we first transformed the 3-D template into a sequence of reference images, which we denote by  $\hat{I}_1, \dots, \hat{I}_{17}$ , corresponding to squint angles  $\phi_1, \dots, \phi_{17}$ . We then correlated  $I_i$  with  $\hat{I}_i$ , for  $i = 1, \dots, 17$ , to obtain the dashed line in Figure 12.

We observe in this plot that the highest correlation scores occur at the squint angles  $-40^\circ$ ,  $0^\circ$ , and  $+40^\circ$ . This result is not surprising because the template was constructed by using images formed at these three squint angles. The average correlation score obtained by using the 3-D template is approximately 0.85, which is much higher than the average correlation score obtained by using the baseline classifier in the previous experiment. Moreover, the scores do not change significantly as the squint angle varies in either direction from  $0^\circ$ , which suggests that the proposed classifier is more robust with respect to changes in squint angle than the baseline classifier.

### Summary

In this article, we have analyzed and suggested improvements to a conventional template-matching classifier currently used in an operational ATR system.

This conventional classifier uses a collection of 2-D SAR reference images to represent a full range of radar viewing directions for a prespecified set of targets. For each target category, the input image is correlated with the reference image that was formed from the most similar radar viewing direction; the input is then classified to the category with the highest correlation score.

Although this algorithm seems reasonable, we found that it produces surprisingly poor classification results for some target types. We explained these poor results by using a simple mathematical model of the SAR imaging process. As our model reveals, radar motion direction is as important as radar viewing direction in specifying SAR imaging geometry. Thus two target images formed with the same radar viewing direction but different radar motion directions can appear quite different. Because the conventional classifier does not explicitly account for radar motion direction, its performance is degraded.

Accordingly, we have proposed and demonstrated an improved version of the conventional template-based classifier that accounts for both direction parameters. In our improved classifier, each 2-D image from the reference library is replaced by a 3-D template so that more target scattering information is available at each viewing direction. As in the conventional classifier, the reference image is selected on the basis of radar viewing direction; by using the mathematical SAR imaging model the improved classifier then transforms the selected 3-D template to a 2-D image whose radar motion direction matches that of the input image.

After comparing the experimental correlation scores between the original 2-D template-based classifier and the improved 3-D template-based classifier, we conclude that the new classifier is significantly more robust with respect to changes in squint angle.

### Acknowledgments

The authors gratefully acknowledge the encouragement and support of Leslie M. Novak and Gerald B. Morse. We also thank Michael C. Burl for many technical discussions on the topic of SAR imaging. This research was sponsored by the Advanced Research Projects Agency.

## REFERENCES

1. J.C. Henry, "The Lincoln Laboratory 33-GHz Airborne Polarimetric SAR Imaging System," *IEEE National Telesystems Conf., Atlanta, GA, 26-27 March 1991*, p. 353.
2. S. Zabele, S. Bachinsky, B. Myers, A. Stiehl, and R. Pinto, *Signature Prediction User's Manual, version 7.1* (The Analytic Science Corporation, Reading, MA, 1990).
3. D.G. Lucnberger, *Linear and Nonlinear Programming* (Addison-Wesley, Reading, MA, 1984).

## APPENDIX: MODELING SPOTLIGHT SAR IMAGING AS A PROJECTION

THROUGHOUT THE TEXT of this article we modeled the SAR imaging process as a projection of the 3-D distribution of target scatterers onto a 2-D slant plane. We relied heavily on this projection model as we analyzed problems with the baseline classifier and developed improvements to it. In this appendix we provide justification for using the projection model, and we state the conditions under which this model is valid.

Our strategy for justifying the projection model consists of four main steps. In the first step, we construct the basis vectors for the radar coordinate system and perform the projection operation on a point reflector to obtain approximate expressions for the SAR image location of the reflector. In the second step, we build a foundation for analyzing the projection approximations by writing the exact nonlinear expressions for the physical quantities that are measured by a SAR when imaging the point reflector. In the third step, we expand these nonlinear expressions into first-order Taylor series in the vicinity of the radar aimpoint, and then observe that the resulting linear approximations are identical to the original projection approximations. Finally, in the fourth step, we quantify the accuracy of the projection model by deriving simple bounds on the approximation error.

We begin by obtaining expressions for the projected location of a point reflector in the radar slant plane. In keeping with the notation we established for Figure 8, we define the position of the point reflector in world coordinates as

$$\mathbf{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}.$$

In addition, we define the time-dependent sensor position in world coordinates as

$$\mathbf{s}(t) = \begin{bmatrix} s_x(t) \\ s_y(t) \\ s_z(t) \end{bmatrix}.$$

Because the sensor is moving at a fixed altitude parallel to the  $y$ -axis in Figure 8, we explicitly remove the time dependency from the first and third coordinates of  $\mathbf{s}(t)$  by setting  $s_x(t) \equiv s_x$  and  $s_z(t) \equiv s_z$ .

With the sensor and point-reflector positions defined, we can now construct the basis vectors for the radar coordinate system. Recall from the main text that we originally expressed the basis vectors  $\hat{\mathbf{r}}$ ,  $\hat{\mathbf{c}}$ , and  $\hat{\mathbf{n}}$  in terms of the imaging angles  $\theta$  and  $\phi$ . In this section, we reconstruct the same basis vectors  $\hat{\mathbf{r}}$ ,  $\hat{\mathbf{c}}$ , and  $\hat{\mathbf{n}}$ , but we express them in a form that is more convenient and more useful for our derivations. In particular, rather than using fixed angles from a single imaging geometry, we express these vectors in a time-dependent form in terms of the sensor coordinates.

At time  $t$ , the range vector  $\hat{\mathbf{r}}(t)$  can be constructed by using the formula

$$\hat{\mathbf{r}}(t) = -\frac{\mathbf{s}(t)}{\|\mathbf{s}(t)\|} = \frac{-1}{\sqrt{s_x^2 + s_y^2(t) + s_z^2}} \begin{bmatrix} s_x \\ s_y(t) \\ s_z \end{bmatrix}.$$

Also, as before, the slant-plane normal vector  $\hat{\mathbf{n}}(t)$  can be constructed by using the formula

$$\hat{\mathbf{n}}(t) = \frac{\hat{\mathbf{r}}(t) \times \hat{\mathbf{y}}}{\|\hat{\mathbf{r}}(t) \times \hat{\mathbf{y}}\|} = \frac{1}{\sqrt{s_x^2 + s_z^2}} \begin{bmatrix} s_z \\ 0 \\ -s_x \end{bmatrix}.$$

(Note that this normal vector is constant, because the radar slant plane does not change with time.) Finally, the cross-range vector  $\hat{\mathbf{c}}(t)$  is determined by the cross product of the other two vectors, as given by

$$\hat{\mathbf{c}}(t) = \hat{\mathbf{n}}(t) \times \hat{\mathbf{r}}(t)$$

$$= \frac{1}{\sqrt{s_x^2 + s_z^2} \sqrt{s_x^2 + s_y^2(t) + s_z^2}} \begin{bmatrix} -s_x s_y(t) \\ s_x^2 + s_z^2 \\ -s_y(t) s_z \end{bmatrix}.$$

The projection of the point  $\mathbf{p}$  onto each of these basis vectors yields the new vector

$$\mathbf{q}(t) = \begin{bmatrix} q_r(t) \\ q_c(t) \\ q_n(t) \end{bmatrix}$$

in radar coordinates. In particular, the range and cross-range coordinates of the point  $\mathbf{p}$  are given by

$$q_r(t) = \mathbf{p} \cdot \hat{\mathbf{r}}(t) = \frac{-1}{\sqrt{s_x^2 + s_y^2(t) + s_z^2}} [p_x s_x + p_y s_y(t) + p_z s_z]$$

and

$$q_c(t) = \mathbf{p} \cdot \hat{\mathbf{c}}(t) = \frac{1}{\sqrt{s_x^2 + s_z^2} \sqrt{s_x^2 + s_y^2(t) + s_z^2}} \times [-p_x s_x s_y(t) + p_y (s_x^2 + s_z^2) - p_z s_y(t) s_z].$$

Having obtained these projection approximations for the range and cross-range coordinates of the point  $\mathbf{p}$ , we now seek expressions for the actual quantities measured by a SAR with respect to the location of  $\mathbf{p}$ . Specifically, these quantities are (1) the *relative range* of the point reflector (i.e., the difference between the distance from the sensor to the point reflector and the distance from the sensor to the aimpoint), and (2) a scaled version of the *relative range rate* of the point reflector (i.e., the rate of change of the relative range with respect to time). We can express the relative range of the point reflector as

$$r[\mathbf{s}(t), \mathbf{p}] = \|\mathbf{s}(t) - \mathbf{p}\| - \|\mathbf{s}(t)\|.$$

In addition, by differentiating the above expression with respect to time, we can write the relative range rate  $\dot{r}[\mathbf{s}(t), \mathbf{p}]$  of the point reflector as

$$\begin{aligned} \dot{r}[\mathbf{s}(t), \mathbf{p}] &= \frac{\partial}{\partial t} r[\mathbf{s}(t), \mathbf{p}] \\ &= \dot{s}_y(t) \left[ \frac{s_y(t) - p_y}{\|\mathbf{s}(t) - \mathbf{p}\|} - \frac{s_y(t)}{\|\mathbf{s}(t)\|} \right]. \end{aligned}$$

In reality, the relative range rate is rarely used in its raw form as it appears above, because it is so highly dependent on both the speed of the sensor and the distance of the sensor from the aimpoint. Rather, these undesirable dependencies on the absolute sensor velocity and position are usually removed through preprocessing, so that the cross-range dimension of the resulting SAR images is normalized, and SAR images created under different imaging conditions can be directly compared. Thus we introduce two simple time-dependent corrections to  $\dot{r}[\mathbf{s}(t), \mathbf{p}]$  (one correction for the absolute sensor velocity, and the other correction for the absolute sensor position) to obtain the *compensated relative range rate*, which we denote by  $\dot{r}_c[\mathbf{s}(t), \mathbf{p}]$ .

To compute the correction for sensor motion, we begin by decomposing the sensor velocity vector into two velocity components in the slant plane, with one component along the radar line of sight, and the other component orthogonal to the radar line of sight. We then note (under the assumption that the sensor is far from the aimpoint) that the relative range rate is affected only by the component of the sensor velocity vector that is orthogonal to the radar line of sight. (This statement is true because the velocity component along the radar line of sight, considered separately, induces exactly the same range rate on all points in the imaging area, resulting in a relative range rate of zero for these points.) The sensor speed in the direction orthogonal to the radar line of sight can be expressed as

$$\dot{s}_y^\perp(t) = \left( -\frac{\sqrt{s_x^2 + s_z^2}}{\|\mathbf{s}(t)\|} \right) \dot{s}_y(t).$$



Because the sensor speed appears in the numerator of the expression for relative range rate, the above compensation for sensor speed will appear in the denominator of the overall range-rate correction term.

To obtain the correction for sensor position, we note that the denominator of each term in the expression for relative range rate is on the order of  $\|\mathbf{s}(t)\|$ . Thus a suitable compensation for the distance of the sensor from the aimpoint is simply this factor  $\|\mathbf{s}(t)\|$ , which will appear in the numerator of the overall range-rate correction term. By applying both of the computed corrections to the original expression for relative range rate, we can express the compensated relative range rate as

$$\begin{aligned}\dot{r}_c[\mathbf{s}(t), \mathbf{p}] &= \left( \frac{\|\mathbf{s}(t)\|}{\dot{s}_y^\perp(t)} \right) \dot{r}[\mathbf{s}(t), \mathbf{p}] \\ &= \frac{-\|\mathbf{s}(t)\|^2}{\sqrt{s_x^2 + s_z^2}} \left[ \frac{s_y(t) - p_y}{\|\mathbf{s}(t) - \mathbf{p}\|} - \frac{s_y(t)}{\|\mathbf{s}(t)\|} \right].\end{aligned}$$

Having produced explicit expressions for relative range and compensated relative range rate, we rewrite them more suggestively as range and cross-range measurements by using the notation

$$\bar{q}_r(t) = r[\mathbf{s}(t), \mathbf{p}]$$

and

$$\bar{q}_c(t) = \dot{r}_c[\mathbf{s}(t), \mathbf{p}].$$

We now show that these actual radar measurements  $\bar{q}_r(t)$  and  $\bar{q}_c(t)$  are well approximated by the previously computed projections  $q_r(t)$  and  $q_c(t)$ , respectively. We begin by separately expanding the expressions for  $\bar{q}_r(t)$  and  $\bar{q}_c(t)$  into Taylor series around the radar aimpoint (i.e., around  $\mathbf{p} = \mathbf{0}$ ), retaining only the first-order terms. For the range component, this procedure yields

$$\begin{aligned}\bar{q}_r(t) &\approx r[\mathbf{s}(t), \mathbf{p}] \Big|_{\mathbf{p}=\mathbf{0}} + p_x \cdot \frac{\partial r[\mathbf{s}(t), \mathbf{p}]}{\partial p_x} \Big|_{\mathbf{p}=\mathbf{0}} \\ &\quad + p_y \cdot \frac{\partial r[\mathbf{s}(t), \mathbf{p}]}{\partial p_y} \Big|_{\mathbf{p}=\mathbf{0}} + p_z \cdot \frac{\partial r[\mathbf{s}(t), \mathbf{p}]}{\partial p_z} \Big|_{\mathbf{p}=\mathbf{0}},\end{aligned}$$

which reduces to

$$\begin{aligned}\bar{q}_r(t) &\approx 0 + \frac{-1}{\sqrt{s_x^2 + s_y^2(t) + s_z^2}} [p_x s_x + p_y s_y(t) + p_z s_z] \\ &= \mathbf{p} \cdot \hat{\mathbf{r}}(t).\end{aligned}$$

For the cross-range component, the Taylor series expansion yields

$$\begin{aligned}\bar{q}_c(t) &\approx \dot{r}_c[\mathbf{s}(t), \mathbf{p}] \Big|_{\mathbf{p}=\mathbf{0}} + p_x \cdot \frac{\partial \dot{r}_c[\mathbf{s}(t), \mathbf{p}]}{\partial p_x} \Big|_{\mathbf{p}=\mathbf{0}} \\ &\quad + p_y \cdot \frac{\partial \dot{r}_c[\mathbf{s}(t), \mathbf{p}]}{\partial p_y} \Big|_{\mathbf{p}=\mathbf{0}} + p_z \cdot \frac{\partial \dot{r}_c[\mathbf{s}(t), \mathbf{p}]}{\partial p_z} \Big|_{\mathbf{p}=\mathbf{0}},\end{aligned}$$

which reduces to

$$\begin{aligned}\bar{q}_c(t) &\approx 0 + \frac{1}{\sqrt{s_x^2 + s_z^2} \sqrt{s_x^2 + s_y^2(t) + s_z^2}} \\ &\quad \times \left[ -p_x s_x s_y(t) + p_y (s_x^2 + s_z^2) - p_z s_z s_y(t) \right] \\ &= \mathbf{p} \cdot \hat{\mathbf{c}}(t).\end{aligned}$$

Note that these linear Taylor series expansions are identical to the original projections  $q_r(t)$  and  $q_c(t)$ , indicating that  $q_r(t)$  and  $q_c(t)$  are good approximations to  $\bar{q}_r(t)$  and  $\bar{q}_c(t)$  in the vicinity of the radar aimpoint.

### Error Analysis

Let us now quantify the error incurred by using these linear approximations instead of the actual measurements. To keep the analysis concise, we examine only the error in the relative range approximation. To simplify notation, we arbitrarily choose a time  $t_0$  and remove the explicit time dependency of variables by setting  $\mathbf{s} \equiv \mathbf{s}(t_0)$ ,  $\bar{q}_r \equiv \bar{q}_r(t_0)$ , and  $q_r \equiv q_r(t_0)$ . In addition, for convenience we express the length of  $\mathbf{p}$  as a fraction  $\delta$  of the length of  $\mathbf{s}$  (i.e.,  $\|\mathbf{p}\| = \delta \|\mathbf{s}\|$ ), and we define  $\beta$  to be the angle between  $\mathbf{p}$  and  $\mathbf{s}$ .

With these definitions, we can rewrite the expression for the true relative range measurement  $\bar{q}_r$  by using the law of cosines to yield

$$\begin{aligned}
\bar{q}_r &= \sqrt{\|s\|^2 + \|p\|^2 - 2\|s\|\|p\|\cos\beta} - \|s\| \\
&= \sqrt{\|s\|^2 + \delta^2\|s\|^2 - 2\delta\|s\|^2\cos\beta} - \|s\| \\
&= \|s\|(\sqrt{1 + \delta^2 - 2\delta\cos\beta} - 1).
\end{aligned}$$

Also, we can rewrite the expression for the relative range approximation  $q_r$  by using the identity

$$\mathbf{p} \cdot \mathbf{s} = \|\mathbf{p}\|\|\mathbf{s}\|\cos\beta$$

to yield

$$\begin{aligned}
q_r &= -\frac{\|\mathbf{p}\|\|\mathbf{s}\|\cos\beta}{\|s\|} \\
&= -\delta\|s\|\cos\beta.
\end{aligned}$$

Because  $q_r$  always underestimates  $\bar{q}_r$ , we can write the absolute approximation error  $e$  simply as

$$\begin{aligned}
e &= \bar{q}_r - q_r \\
&= \|s\|(\sqrt{1 + \delta^2 - 2\delta\cos\beta} - 1 + \delta\cos\beta).
\end{aligned}$$

For a fixed value of  $\delta$ , the error reaches its maximum value at the angle  $\beta = \cos^{-1}(\delta/2)$ . Substituting this angle back into the formula for  $e$  yields the upper bound

$$e_{\max} = \frac{\delta^2}{2}\|s\| = \frac{\delta}{2}\|p\|.$$

For the specific case of data collection with the Lincoln Laboratory millimeter-wave sensor,  $\delta$  is typically no larger than 0.05, so the error incurred by using the projection approximation for a given point  $\mathbf{p}$  is no more than 2.5% of the distance of  $\mathbf{p}$  from the aimpoint.

The derivation of the error bound for the cross-range approximation is similar in spirit to the derivation given above, but it is much more lengthy and tedious, and hence is omitted.



**SHAWN M. VERBOUT** is an assistant staff member in the Surveillance Systems group. His research interests are in the development of algorithms for signal detection, estimation, and classification. He received a B.S. degree in mathematics from Illinois State University, and he is currently pursuing a master's degree in electrical engineering at MIT as a member of the Lincoln Laboratory Staff Associate Program. Shawn has been at Lincoln Laboratory since 1987.



**WILLIAM W. IRVING** is an associate staff member in the Surveillance Systems group. His research interests are in the areas of decentralized Bayesian detection theory and multi-resolution stochastic modeling. He received S.B., S.M., and E.E. degrees in electrical engineering from MIT, where he is currently pursuing a Ph.D. degree as a member of the Lincoln Laboratory Staff Associate Program. Bill has been at Lincoln Laboratory since 1987.



**AMANDA S. HANES** is an assistant staff member in the Surveillance Systems group. Her research interests are in the detection, discrimination, and classification of stationary targets. She received a B.S. degree in mathematics from Tufts University, and she also worked at the Johns Hopkins University Applied Physics Laboratory through the Professional Summer Student Program. Amanda has been at Lincoln Laboratory since 1989.

---

# Neural Systems for Automatic Target Learning and Recognition

Allen M. Waxman, Michael Seibert, Ann Marie Bernardon, and David A. Fay

■ We have designed and implemented several computational neural systems for the automatic learning and recognition of targets in both passive visible and synthetic-aperture radar (SAR) imagery. Motivated by biological vision systems (in particular, that of the macaque monkey), our computational neural systems employ a variety of neural networks. Boundary Contour System (BCS) and Feature Contour System (FCS) networks are used for image conditioning. Shunting center-surround networks, Diffusion-Enhancement Bilayer (DEB) networks, log-polar transforms, and overlapping receptive fields are responsible for feature extraction and coding. Adaptive Resonance Theory (ART-2) networks perform aspect categorization and template learning of the targets. And Aspect networks are used to accumulate evidence/confidence over temporal sequences of imagery.

In this article we present an overview of our research for the past several years, highlighting our earlier work on the unsupervised learning of three-dimensional (3-D) objects as applied to aircraft recognition in the passive visible domain, the recent modification of this system with application to the learning and recognition of tactical targets from SAR imagery, the further application of this system to reentry-vehicle recognition from inverse SAR, or ISAR, imagery, and the incorporation of this recognition system on a mobile robot called the Mobile Adaptive Visual Navigator (MAVIN) at Lincoln Laboratory.

**F**ROM THE STUDY of biological vision systems, we can learn much that applies to the design of computational neural systems for target recognition. These insights are most relevant to passive vision systems, such as visible and multispectral infrared imaging systems, but similar organizing principles are also useful in the radar imaging domain. In the next section, we summarize the primary lessons that have been learned from the anatomical, physiological, and psychophysical study of vision systems in the macaque monkey and man. These insights are then applied throughout the remaining sections of this review. (Note: An introduction to biological vision, learning, and memory can be found in the September 1992 special issue of *Scientific American*, which is entitled "Mind and Brain.")

## Design Constraints from Biological Vision

The vision systems of primates contain two primary processing streams: the *parvocellular* stream, which processes shape information, and the *magnocellular* stream, which processes motion information (see References 1 and 2, and the references cited therein). Both streams begin in the retina and culminate in the parietal and temporal lobes of the cerebral cortex. Our automatic target recognition (ATR) systems have focused on the modeling of the parvocellular stream for the learning and recognition of three-dimensional (3-D) objects, although we have utilized image sequences to accumulate evidence over time. The image motion of objects can also be useful for recognizing potential targets, and we have developed

neurocomputational systems [3] to extract such information in real time (30 velocity fields per second) on the Pipelined Image Processing Engine (PIPE), a video-rate parallel-processing computer. The integration of an object's image motion with its shape information can potentially enhance the ATR process, and is a topic we are currently investigating.

The early visual processing that takes place in the retina, lateral geniculate nucleus, geniculo-cortical connections, and visual cortical areas V1, V2, and V4 of the occipital lobe are responsible for

1. conditioning imagery so as to render it invariant to the prevailing illumination (while producing smoothly shaded percepts of objects),
2. localizing features (such as edges, high-curvature points, and high-contrast points) that describe 2-D shapes, and
3. transforming the resulting feature pattern so as to render it invariant to object location, scale, orientation around the line of sight, and small deformation due to any foreshortening resulting from a rotation in depth (i.e., a rotation around an axis perpendicular to the line of sight), while still retaining measurements of these spatial attributes.

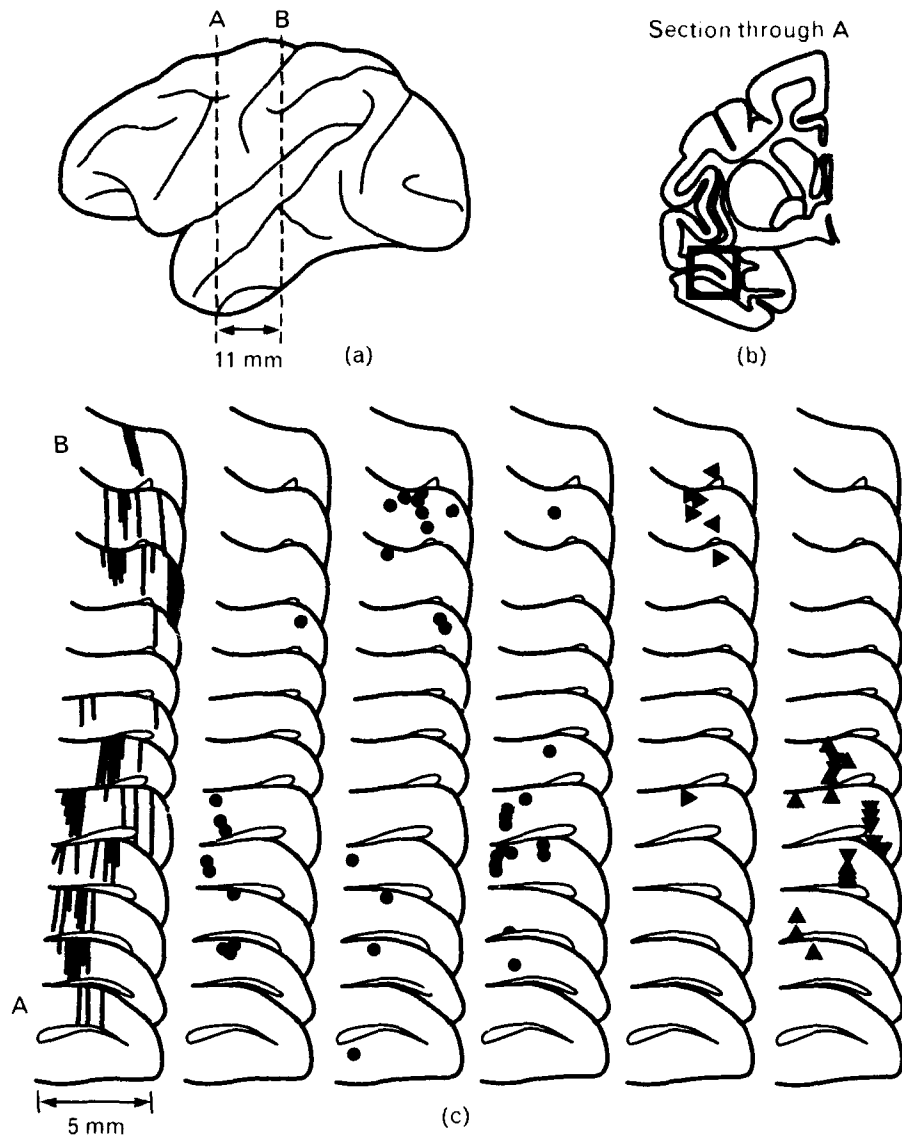
These invariant representations of 2-D object shapes make their way to the inferior temporal cortex via connections between the occipital and temporal lobes, whereas the location/scale/orientation information is relayed to the posterior part of the parietal lobe via connections between the occipital and parietal lobes. Object-location information is conveyed to the parietal lobe also via the superior colliculus, which receives direct connections from the geniculate nucleus and is intimately involved in attentional processes. These two cortical pathways—one subserving *object vision* (in the temporal lobe) and the other subserving *spatial vision* (in the parietal lobe)—have come to be known as the *what and where systems* [4]. Fusion of the *what* and *where* information is achieved via reciprocal connections between the temporal and parietal lobes, as well as by indirect connections between other regions of the brain such as the hippocampus, although the details are not yet understood.

Insight into the later stages of visual processing and

3-D object representation can be gained by studying the superior temporal sulcus (STS) in the temporal lobe of the macaque monkey. This area is known to be the site of cells tuned for the recognition of faces and other body parts. Of course, the faces that a monkey recognizes are indicative of the monkey's visual experiences, and reflect the visual learning process itself. We have learned much from the work of D.I. Perrett and his colleagues at the University of St. Andrews in Scotland [5–8].

The notion of cells specifically tuned to the recognition of certain objects (analogous to the orientationally tuned *edge sensitive neurons* in V1 discovered by D. Hubel and T. Wiesel in 1959) was popularized by H. Barlow in 1972, and became known as the *grandmother-cell hypothesis*, as if to emphasize that a single neuron becomes active to signal the recognition of one's grandmother. And, for the past 20 years, a debate has raged over this notion of single-cell versus distributed-network coding of visual objects. In fact, this seemingly absurd notion of single-cell coding seems to have much supporting evidence, as illustrated in Perrett's work below (and confirmed by other investigators). The strict notion of grandmother cells, however, must be reinterpreted in light of the fact that many layers of processing precede the view-specific coding of objects, and a hierarchical pooling of cells is required to influence the object-specific cell. Moreover, many visual objects may activate this cell, although it is maximally active for a specific object, whereas other cells are more active in the case of the other objects. Hence, a recognition decision must follow a neural competition between grandmother cells, and possibly an evidence-accumulation phase among multiple views when such views are available.

Figure 1 (from Reference 5) illustrates the STS area in the macaque monkey brain. The figure shows the locations of neurons detected by Perrett that are highly tuned to the face and profile views of heads, rotations of heads between specific views, and conjunctions of face views with up/down/left/right motions. Perrett's subsequent work [7] indicates the existence of view-specific cells, each one tuned for a particular view around a certain class of heads, and still other cells, called view-general cells, that respond to any view of a

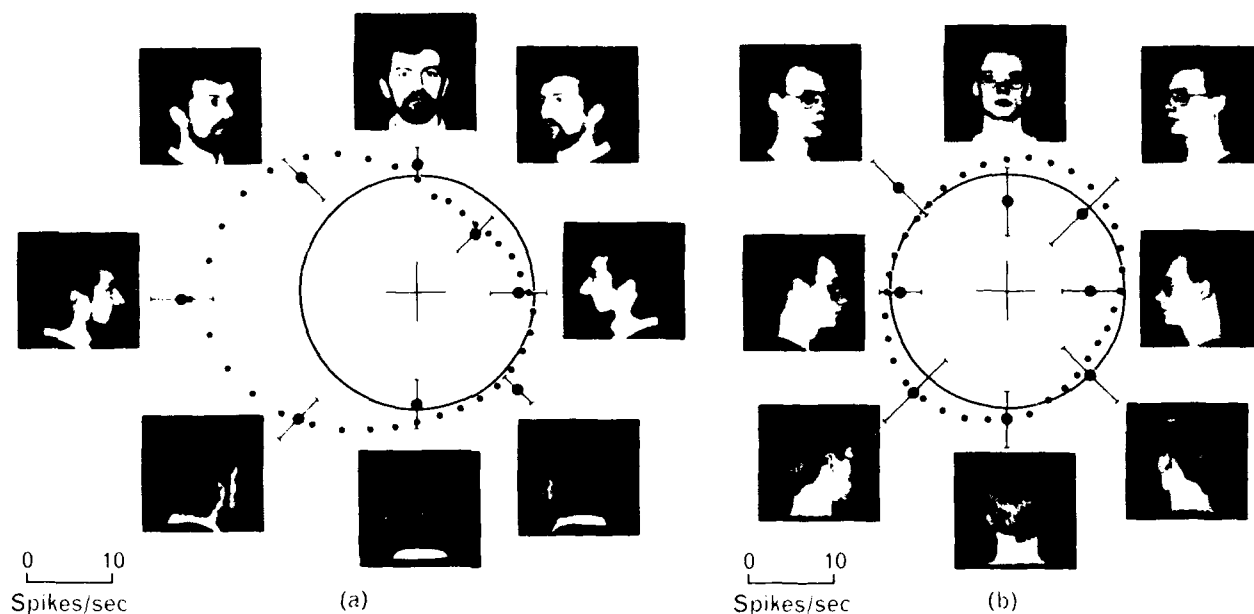


**FIGURE 1.** View-based coding of faces in the temporal cortex of the macaque monkey: (a) lateral view of the monkey brain, (b) coronal cross section with a red box around the superior temporal sulcus (STS), and (c) serial sections of the STS area investigated. From left to right, the sections illustrate the electrode tracks, cells selective to face views, cells selective to profile views, cells selective to transitions between views during head rotations, cells selective to faces moving left/right, and cells selective to faces moving up/down. (Adapted from D.I. Perrett et al. [5], with permission from *Trends in Neurosciences*. Elsevier Science Publishers B.V.)

specific head (as if the view-general cells were connected to all of the corresponding view-specific cells). View-specific cells respond to the same face views with similar activity levels, regardless of the illumination strength or color, the size or 2-D orientation of the face, and the position of the face in the field of

view. Such cells have apparently learned 2-D-invariant shape codes.

Figure 2 (from Reference 6) provides a striking example of view and identity coding in the macaque temporal cortex. In the experiment, a monkey was shown different views of the faces of two familiar



**FIGURE 2.** View and identity coding in the macaque temporal cortex for (a) subject 1 and (b) subject 2. In the experiment, a monkey was shown different views of the faces of two familiar people (subjects 1 and 2), and the activity of a single STS neuron in the monkey's brain was monitored with an electrical probe. The results are plotted in spikes/sec radially from the "+" symbol; the black solid circle denotes the spontaneous background activity level. The experimental measurements are represented by the large red dots with error bars indicating standard deviations over several repeated trials. Note that the neuron has a clear preference for the right profile view of subject 1, and no significant response to any view of subject 2 (Adapted from Perrett et al. [6], with permission from the *Journal of Experimental Biology*, the Company of Biologists Ltd.)

people, while an electrical probe monitored the activity of a single STS neuron in the monkey's brain. The results are shown in Figure 2, in which cell activity is plotted radially from the "+" symbol and the solid circle denotes the spontaneous background activity level. The experimental measurements are represented by the large dots with error bars indicating standard deviations over several repeated trials. Figure 2(a) shows that the neuron is highly tuned for the right profile view of subject 1. Nearby views (some at a  $\pm 15^\circ$  angle from the right profile) still generate cell activity, though at a much reduced rate. All views of subject 2 (a rather different looking face) generate no significant activity above the background level, as shown in Figure 2(b). Thus this neuron might someday become a *grandfather cell*!

In summary, monkeys learn to recognize faces by employing a view-based strategy. Representations of 2-D shapes are learned and stored in view-specific STS cells. These cells code shape information that is invariant to illumination, position, scale, orientation

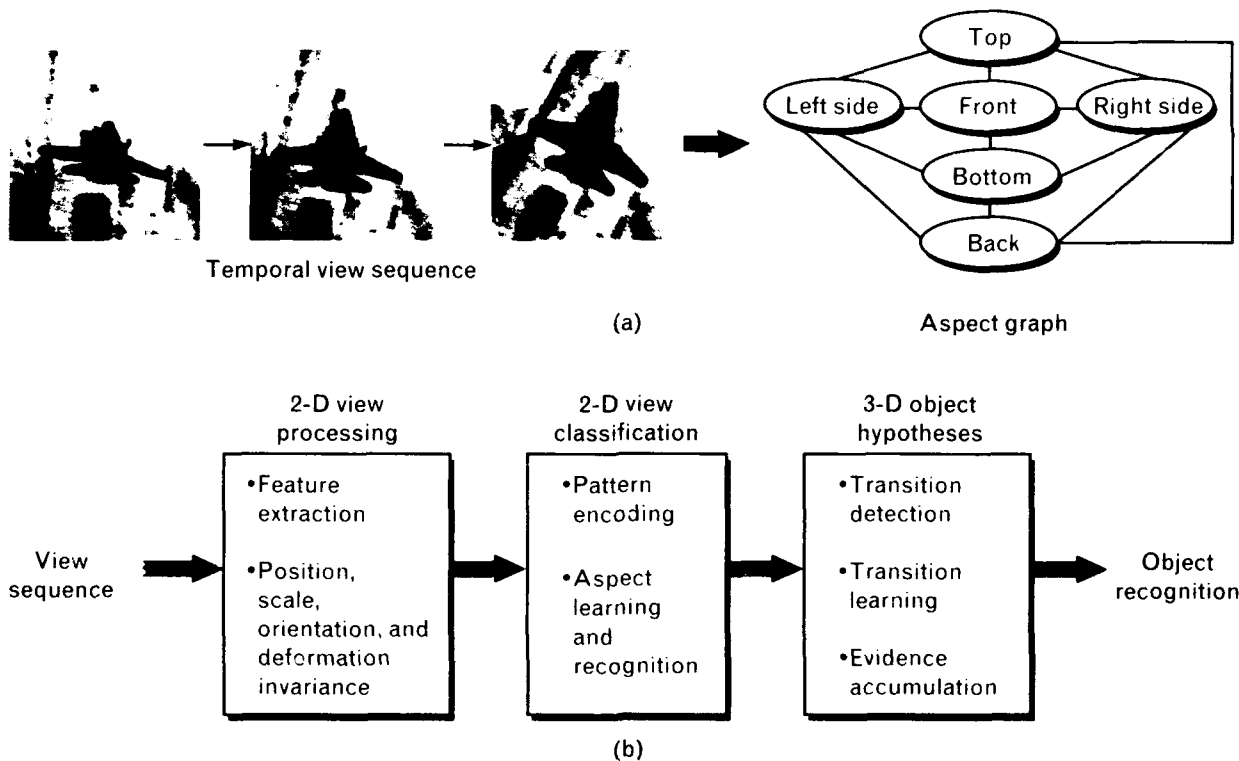
around the line of sight, and small foreshortening deformation. Other cells code transitions between neighboring views that have been exposed by the rotations of a subject's head. A hierarchical combination of the two types of cells allows the construction of view-general cells that are selectively activated by specific heads regardless of the viewing direction. This same strategy for the learning and recognizing of 3-D heads (and, possibly, other objects) can be applied usefully to the design of artificial neural systems for VLR.

### Aircraft Recognition from Visible Image Sequences

We designed our first end-to-end VLR system for the passive visible domain, and applied the system to high-contrast imagery of model F-16, F-18, and F1K-1 (Spruce Goose) aircraft moving against textured backgrounds. (Note: Detailed descriptions of this neural system are contained in several papers by M. Seibert and A.M. Waxman [9-11].)

Figure 3 provides a conceptual overview of the system, in which a temporal view sequence of an object leads to the learning of an *aspect graph* [12] representation of that 3-D object. We can divide the system into three main functional stages, the first of which performs 2-D view processing to extract features (invariant to illumination) from the individual images, group these features to locate object position, and transform the features to render the pattern invariant to scale, orientation, and small deformation. The second stage takes these invariant feature patterns and clusters them into categories of similar views, or *aspects*. This 2-D view classification is done in an unsupervised way; i.e., it is strictly data driven without any category definition by a human. Along with the learning of these aspect categories, a prototype

feature-pattern template is established for each category. The aspect categories correspond to the nodes of an aspect-graph representation of the target; they also play the role of view-specific cells for aircraft. The third stage detects the transitions over time between aspect categories (while the target is tracked in relative motion), learns these transitions, and accumulates evidence for possible targets. The learned transitions are like the arcs that connect the nodes in the aspect-graph concept, and are reminiscent of the STS neurons that are activated by the rotation of the heads between views in Figure 1. The ability to accumulate evidence over time is significant, for there are often cases in which a single view of a target is not sufficient to identify the target unambiguously; moreover, this fusion of evidence leads to a notion of



**FIGURE 3.** Conceptual approach of ATR neural system for passive visible image sequences: (a) temporal view sequence of images and corresponding aspect graph, and (b) functional block diagram of system. As a target moves relative to an observer, qualitatively different views are exposed in a temporal view sequence. The views unfold in an orderly fashion that is represented in the aspect graph. Each image in the sequence is processed by three stages of networks performing feature extraction and invariant mappings, classification of feature maps into aspect categories, and 3-D object evidence accumulation from the recognition of categories and transitions. The learned categories and transitions are analogous to the nodes and arcs, respectively, of an associated aspect graph.



confidence in the recognition decision.

These three processing stages can each be realized with multiple neural networks, and together the networks comprise a neural system architecture, as shown in Figure 4. Here, each module is an individual network that is annotated by the module's functional role in the system. Two processing streams are shown: the gray modules form a *parvocellular stream*, and the red modules form an *attentional stream*.

In the system, images are captured with a conventional CCD camera (which could be replaced by an infrared imaging system) and objects are segmented from the background by using a combination of motion and contrast information. Next, a *shunting center-surround network* enhances the edges of the segmented object, and a *Diffusion-Enhancement Bilayer (DEB)* extracts and dynamically groups the feature points of high edge curvature into a position centroid, as shown in Figure 5. These networks form nonlinear dynamical systems in which individual nodes are governed by (Hodgkin-Huxley-like) cell-membrane equations that resemble the charging dynamics of coupled resistor-capacitor networks. (See Reference 13 by S. Grossberg for a review of his pioneering work on dynamical neural networks, including shunting center-surround networks. Also, see References 14 and 15 for a reformulation of the DEB in terms of coupled dynamical layers of astrocyte glial-like diffusion cells and neural-like contrast-enhancing cells, all inspired by biology and applied to the psychophysical percept of long-range apparent motion.)

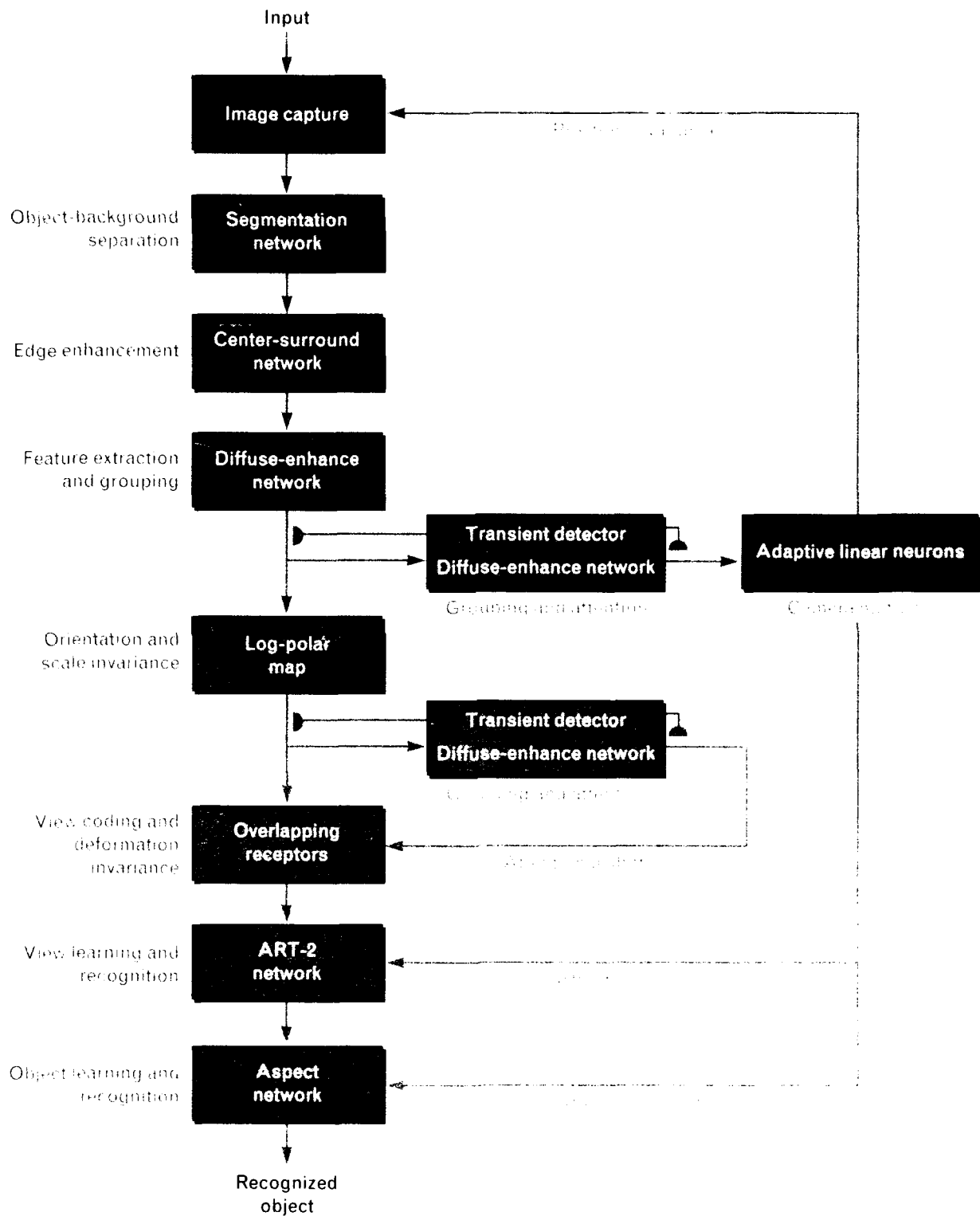
The centroid determined by the DEB network is used to track and fixate the object, and serves as the origin of a log-polar transform of the extracted-feature map. This transformation is very closely approximated by the axonal connections between the lateral geniculate nucleus and the primary visual cortex V1 [16]. In our system the transformation serves to convert changes in 2-D scale and 2-D orientation of the visual-feature map into a translation along new orthogonal axes. These processing steps are illustrated for an F-18 silhouette in Figures 6(a), (b), and (c). The log-polar feature map (periodic in orientation angle  $\theta$ ) is then input to a second DEB to determine a new feature centroid in the transformed coordinates. The spatial pattern of features now represents

the original view of the F-18 invariant to illumination, position, scale, and orientation.

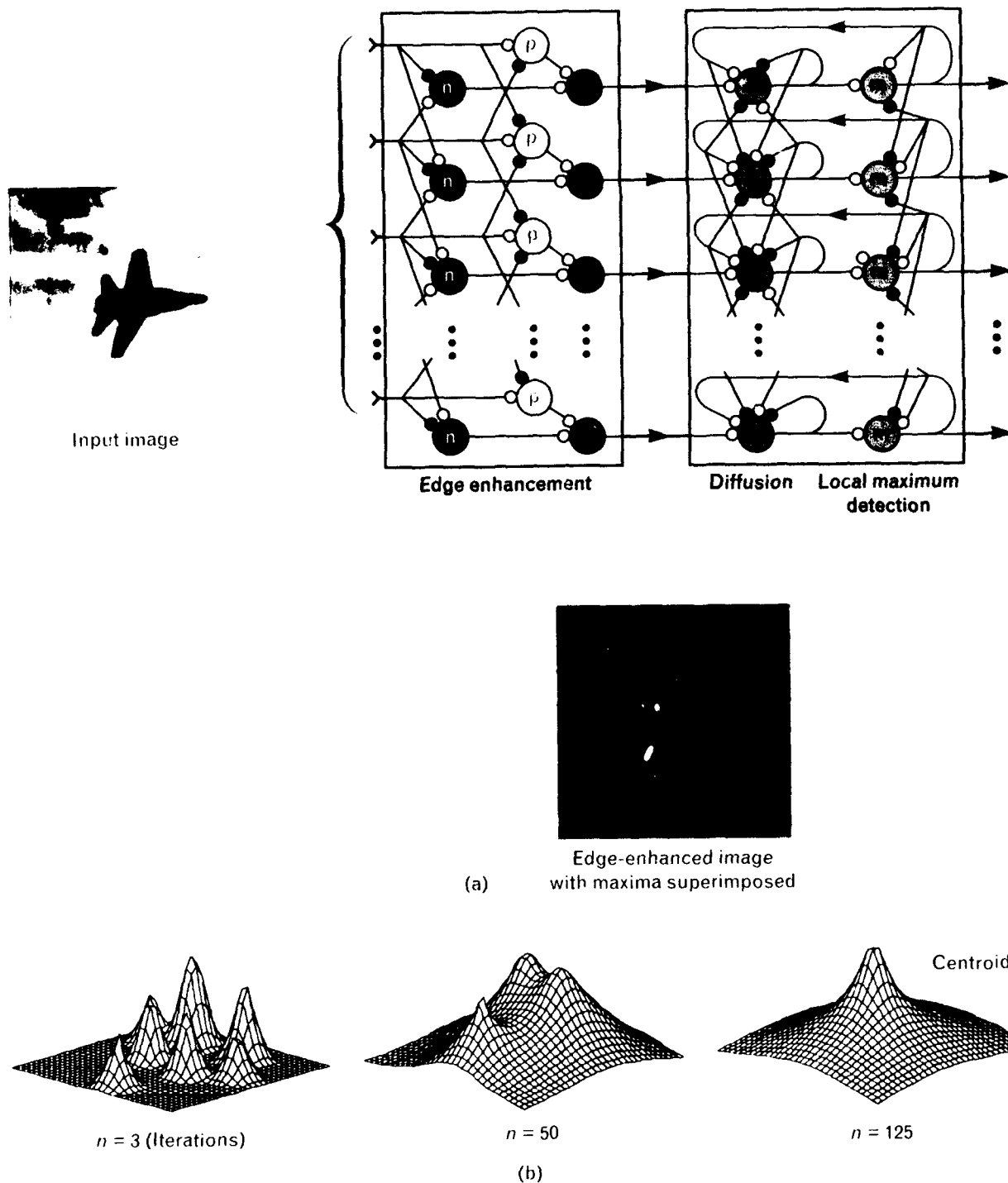
The next layer of processing indicated in Figure 4 consists of *overlapping receptive fields*: the processing is aligned with the centroid that was detected on the log-polar map, and serves to render the feature pattern somewhat insensitive to nonlinear spatial deformation. In the processing (Figure 7), a small array of Gaussian-weighted overlapping receptors are excited by the underlying features in the log-polar map, and the output of the array provides a much compressed code of the spatial feature pattern. (An individual receptor is activated by the feature within the receptor's field that lies closest to the field's center, and the feature's distance is coded according to a Gaussian falloff.) This compressed code is illustrated for the F-18 in Figure 6(d) for the case of a  $5 \times 5$  array of overlapping receptors. In the figure, the sizes of the dots correspond to the receptor activation level: the larger the dot, the greater the activation. This coarse coding of spatial feature patterns simultaneously provides for enormous data reduction from the original target image (compared, for example, with a direct template-matching approach), leads to a tolerance for small deformations due to rotations in depth and inaccurate feature extraction, and yields an input vector for the classification network that forms the next system module.

The later stages of vision support the learning and recognition process. In our system, learning and recognition are realized by two modules consisting of an *Adaptive Resonance Theory* network (cf. several papers on various ART networks in Reference 17) and an *Aspect network* [10, 11].

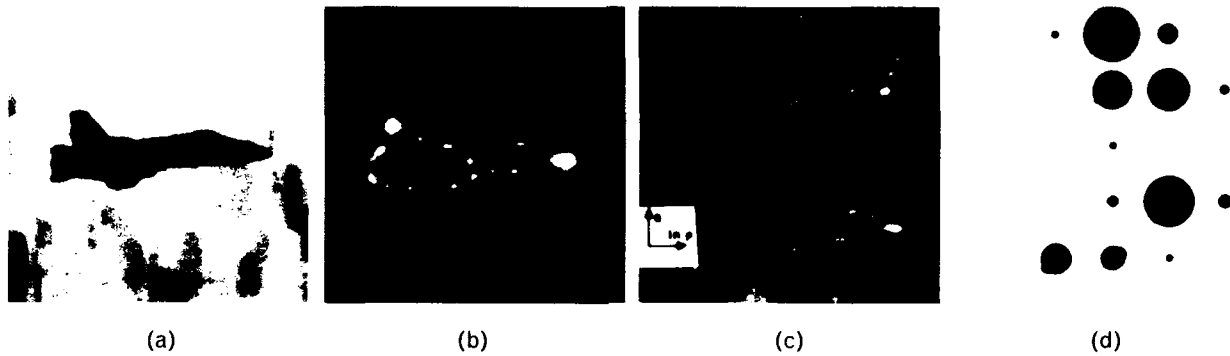
Figure 8 illustrates the ART-2 architecture for unsupervised category learning and recognition. (Note: ART-2 is one implementation of Adaptive Resonance Theory for patterns consisting of real numbers.) The ART-2 network takes an  $N$ -dimensional input vector (in our case, the overlapping receptive field pattern with dimension of order 10 to 100) and first processes it through circuitry that contrast-enhances and normalizes the input as a short-term memory (STM) pattern. ART-2 then passes this pattern through a bottom-up filter (or template) stored in long-term memory (LTM) to excite a field of STM category



**FIGURE 4.** Modular system architecture for the learning and recognition of 3-D targets from visible imagery. The system is organized into two streams of neural network modules: the gray *parvocellular* stream for invariant shape learning and recognition, and the red *attentional* stream. The functional role of each module is indicated along with the type of network.



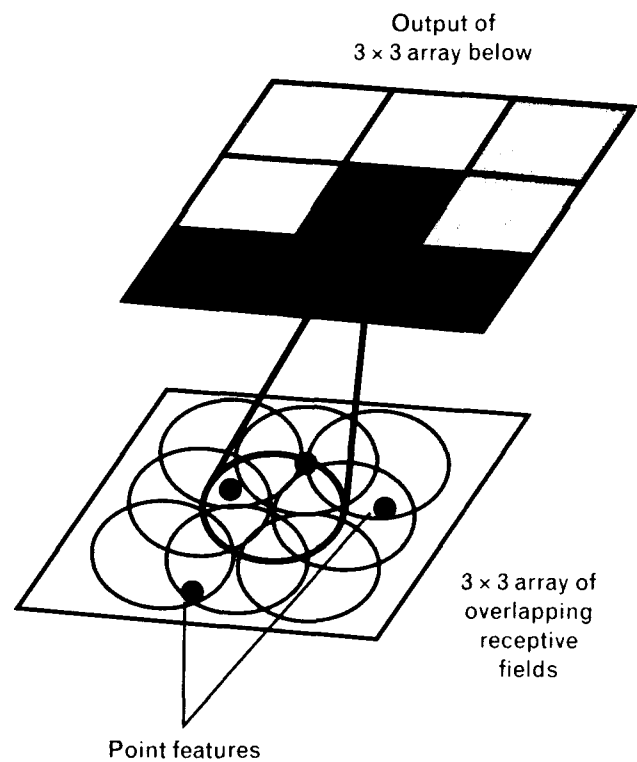
**FIGURE 5.** Diffusion-Enhancement Bilayer (DEB) for feature extraction and grouping: (a) architecture diagram and (b) evolving map of high-curvature points. The first stages of processing are accomplished by center-surround networks to edge-enhance the segmented object, and a diffusion-enhancement network to isolate points of high curvature along the silhouette. These feature points are dynamically grouped into a centroid (providing a focus of attention) by another DEB, which couples a diffusion layer to a contrast-enhancing layer in a feedforward and feedback configuration. (For a detailed description of DEBs, see References 9, 14, and 15.)



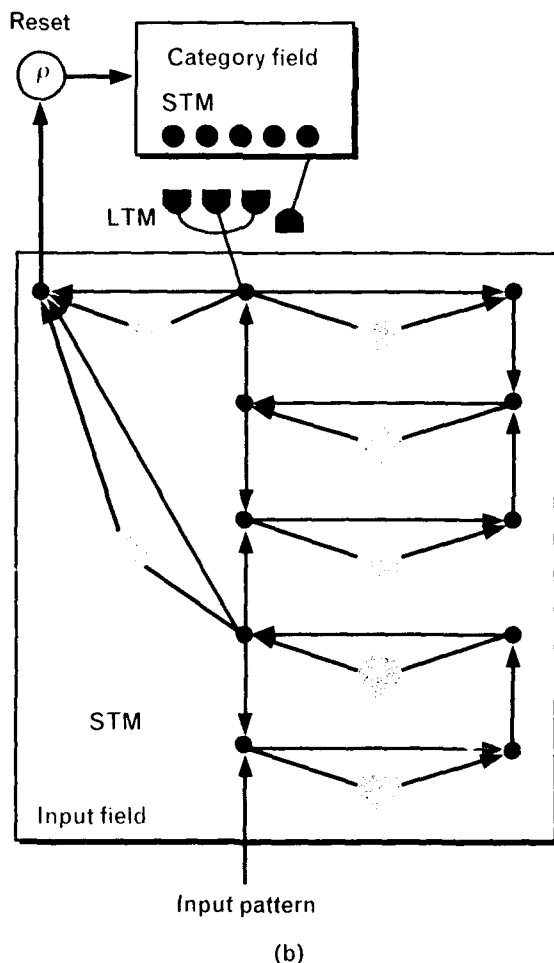
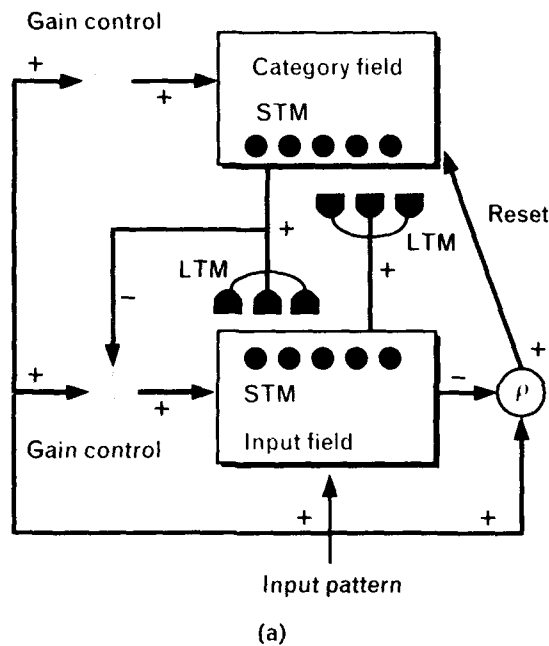
**FIGURE 6.** Stages in the processing of a 2-D view of a model F-18 aircraft: (a) the original image, (b) the edge-enhanced silhouette with DEB features superimposed and the centroid indicated with a "+," (c) log-polar mapping of the image in part b, with the new centroid indicated with a "+," and (d) the resulting output of a  $5 \times 5$  array of overlapping receptive fields (see Figure 7) that forms the pattern fed to the Adaptive Resonance Theory (ART-2) network. In the image in part d, larger dots represent greater activity in the corresponding receptive fields.

nodes (our view-specific cells, or aspect nodes). These category nodes compete among themselves to choose a maximally activated winner, which in turn activates top-down feedback of a learned template also stored in LTM. This feedback represents the network's expectation of a specific input pattern. A vigilance parameter  $\rho$  (in the interval 0 to 1) that is set in advance by the user mediates the matching of the enhanced input pattern with the top-down template. Thus, simply having a best match among already established categories is not enough; rather, the best match must satisfy the established vigilance. When the match does satisfy the vigilance criterion, the network goes into a state of resonant oscillations between layers, and the bottom-up and top-down filters adapt slightly for better representation of the recent input pattern. When the vigilance criterion has not been met, the network generates a reset signal that flips the category field, thus suppressing the recent winner and reactivating the former losers. In this way, an uncommitted category node can establish a new category and a new template can be learned. ART-2 has several important attributes that make it particularly well suited to ATR applications: it supports on-line, real-time, unsupervised, stable category learning and refinement. We have utilized ART-2 successfully in a number of applications.

To present our results for the ART-2 classification of different aircraft, we introduce the concept of a *viewing sphere*, as illustrated in Figure 9. Note that a



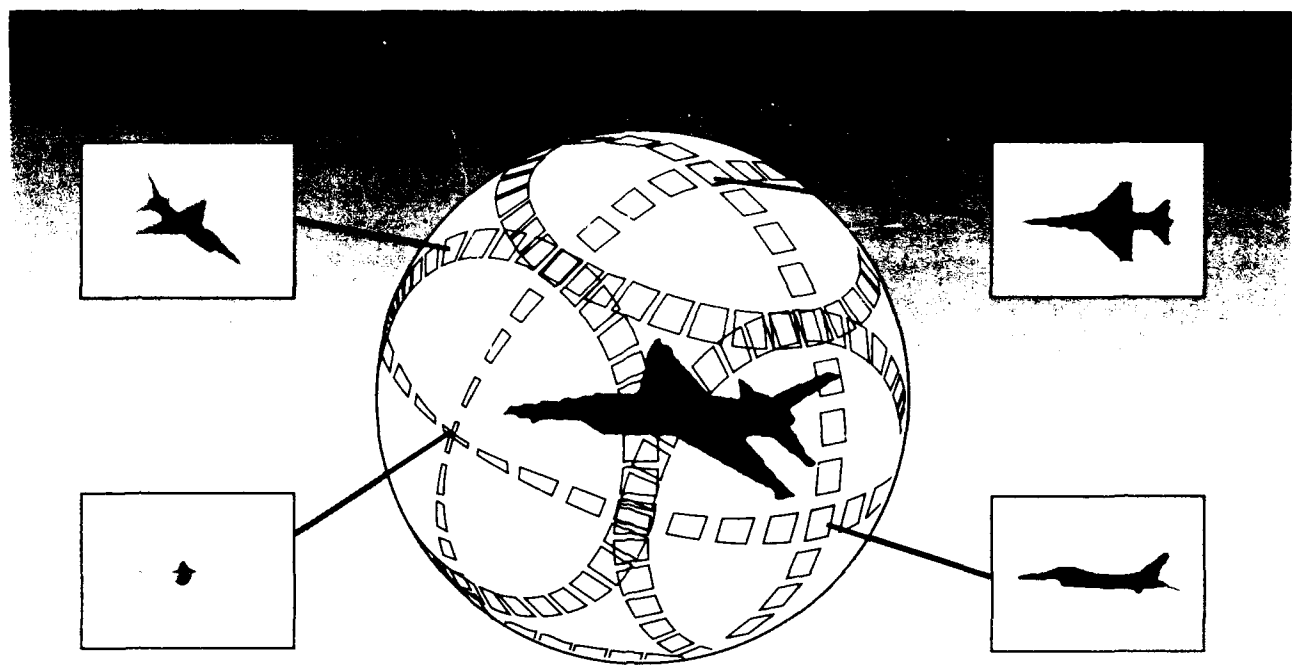
**FIGURE 7.** Spatial coding of features by overlapping receptive fields. Each circular field is activated according to a Gaussian-weighted distance to the point feature that is closest to the receptor center. (Note: Lighter colors in the figure represent closer distances.) These receptors provide enormous data compression, and they code spatial relations of features robustly with respect to deformations due to foreshortening. The fields convert a binary feature map to an analog pattern that is then suited for ART-2 classification.



location on the viewing sphere for an example aircraft corresponds to the view of that aircraft as seen from that particular direction. Using this viewing-sphere concept, Figure 10 summarizes the results of feature extraction, coding, and ART-2 classification for an F-18 model aircraft. With 535 input views of the F-18 and a vigilance  $\rho$  of 0.93, ART-2 generates 12 categories of the aircraft. In Figure 10(a), the categories, or aspects, are shown color coded on an *aspect sphere* with 12 different unrelated colors (i.e., a dark blue has no relation to a light blue). Note that the aspects subtend finite solid angles on the sphere (the target is oriented with its nose to the left). Because of object silhouette symmetry, only one quadrant of the sphere is shown. We can visualize example silhouettes that correspond to the 12 categories by selecting locations on the aspect sphere falling at the centers of each of the established categories, as shown in Figure 10(b). The corresponding silhouettes (numbered 1 through 12 in Figure 10(c)) represent prototype views that the system has created in an unsupervised manner. Notice the variety of silhouettes selected: some prototype views capture the wing shapes, some capture the double tail fins, some capture the dual exhausts, while others emphasize traditional top and side views. Also note the similarity between silhouettes 2 and 5, given the proximity of their corresponding centroids in Figure 10(b). Yet, although similar, silhouettes 2 and 5 do exhibit subtle differences, e.g., the differing slopes of the top portion of the visible tail fin. All of the views in Figure 10(c) were selected automatically. When the vigilance  $\rho$  was increased from 0.93 to 0.95, the ART-2 network generated 24 categories.

In addition to the F-18, we have also investigated

**FIGURE 8.** ART-2 network: (a) architecture and (b) circuit model. ART-2 takes analog input patterns and clusters them into categories by using unsupervised competitive learning. ART-2 can be trained on a dataset, then used to recognize data patterns in the field while continuing to refine its learned category representations (i.e., templates) stored in its adaptive synapses. The vigilance parameter  $\rho$  mediates the matching of the enhanced input pattern stored in short-term memory (STM) with a learned template from long-term memory (LTM). (Adapted from G.A. Carpenter et al. [17], with permission. This reference also contains a detailed description of ART.)

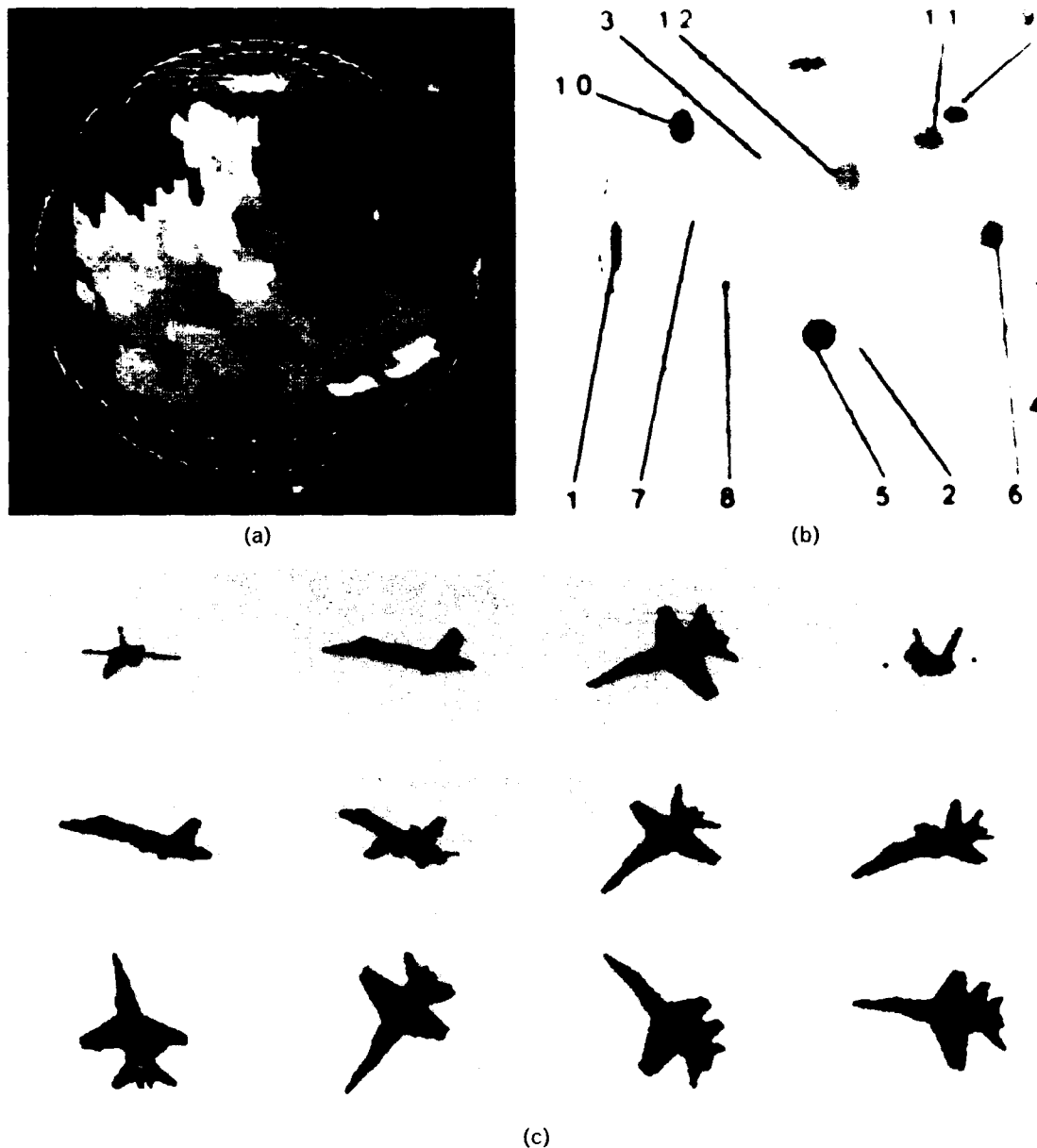


**FIGURE 9.** Example viewing sphere for a fighter aircraft. Note that a location on the sphere corresponds to the view of the aircraft as seen from that particular direction. Silhouettes of the aircraft are shown from different viewing directions. The silhouettes were obtained by applying thresholds to imagery that was captured with a charge-coupled device (CCD) camera and frame grabber. (The jagged contours reflect the finite pixel sizes of the CCD imager.)

ART-2 classification for an F-16 and HK-1 (Spruce Goose) model aircraft, as shown in Figure 11. Approximately 500 views of each aircraft were collected and processed with a single ART-2 module. (The next section, "Tactical Target Recognition in the Synthetic-Aperture Radar [SAR] Spotlight Mode," presents an alternative strategy of using one ART-2 module per target for the SAR application.) All the views together resulted in only 41 independent categories at a vigilance  $\rho$  of 0.93. (Note: Figure 10 investigated the categorization of a single target by using views of just that target. Thus, at the same vigilance setting of 0.93, ART-2 generated only 12 categories, in contrast to the 26 categories of Figure 11[a].) The individual aspect spheres show the similarity in category layout between the two fighter aircraft, and the obvious differences between fighter and transport-like aircraft. The spheres also indicate how certain views of the two fighters are ambiguous, at least in terms of the features extracted by the system. The individual 3-D

targets are represented by roughly 25 categories each. Note in Figure 11 that some of the categories are common to two or more of the targets; i.e., the light yellow in Figure 11(a) corresponds to the same category that is represented by the same light yellow in Figure 11(b).

The aspect spheres in Figure 11 also illustrate the neighbor relations among categories as one rotates or explores a target in 3-D. These neighbor relations correspond to permitted transitions among categories, and are learned and exploited by our *Aspect net work*. Much like the STS cells that code view transitions, and the hierarchical pooling of view specific cells to form view general object specific cells, our Aspect networks self-organize into connections among aspect category nodes that preferentially channel activity into corresponding 3-D object nodes when successive aspects occur in a permitted sequence. The Aspect networks learn these aspect transitions incrementally during controlled training sessions, or im-



**FIGURE 10.** Results of feature extraction, coding, and ART-2 classification of an F-18 model aircraft alone at a vigilance  $\rho$  of 0.93: (a) aspect sphere showing the 12 aspects (color coded) generated by ART-2, (b) centroids of the largest regions of the 12 aspects, or categories, and (c) corresponding example silhouettes of the regions in part b. These views have been selected automatically by the system. (Note: The 12 colors used for the aspect sphere have been selected arbitrarily; i.e., a dark blue has no relation to a light blue.)

tially in the field after the aspect categorization has stabilized (i.e., after repeated exposures to the training data yield the same categorization). Then, during the imaging of a target in motion, multiple viewpoints are experienced, leading to recognition of multiple aspects by the ART-2 network, followed by evidence accumulation by the object nodes in the Aspect net-

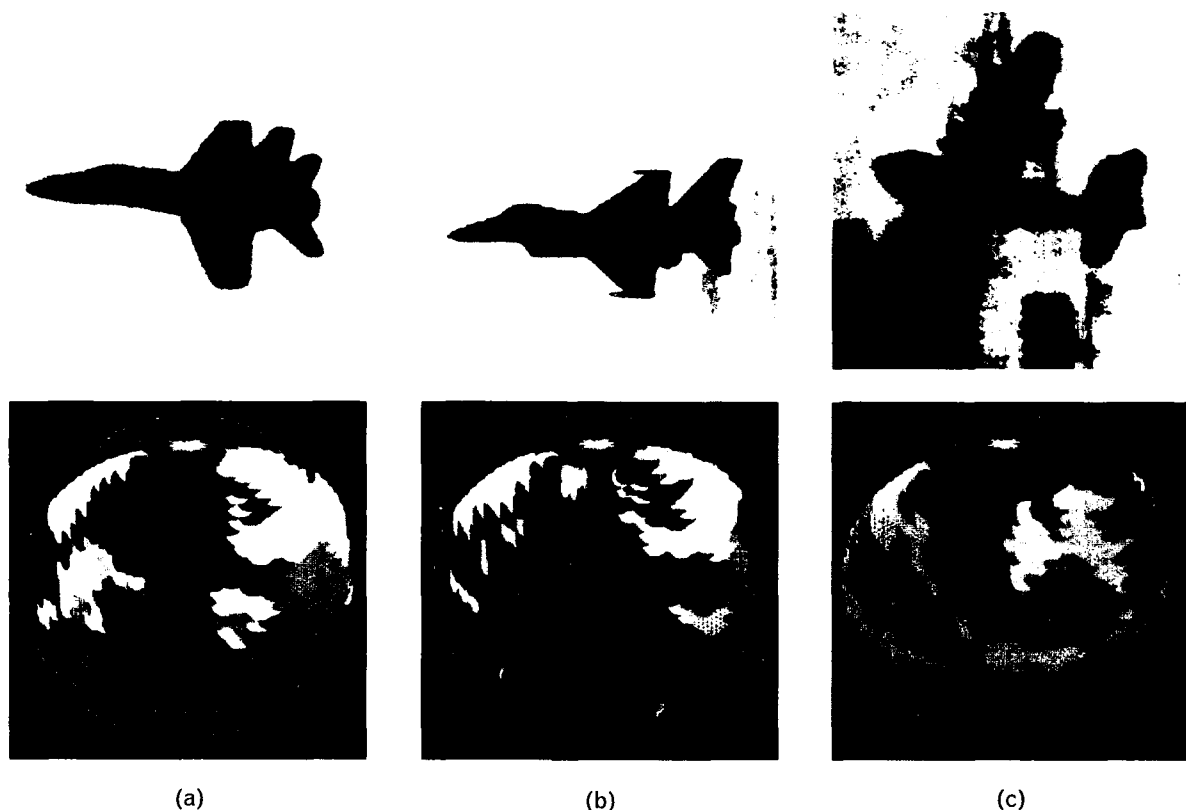
work. Target trajectories are realized as a set of aspect categories linked together by aspect transitions. Much more information becomes available when we consider the aspect transitions among ambiguous views. For example, even if both views of a two-aspect sequence are each ambiguous among potential targets, the additional aspect-transition information is often

sufficient for the preferential activation of the correct target node in the Aspect network.

Figure 12 illustrates an Aspect network for a single object, along with an enlarged view of the network's *adaptive axo-axo-dendritic synapse*. This synapse brings together in close physical proximity projections from pairs of aspect nodes onto a branch of the dendritic tree leading to an object node. When ART-2 categories are excited in temporal succession, the aspect nodes shown charge or discharge exponentially like capacitors, and their temporal overlap of activity supports a Hebbian form of correlational learning on the connecting synapse (cf. Reference 13 for a discussion of modified Hebbian learning with gated decay). The

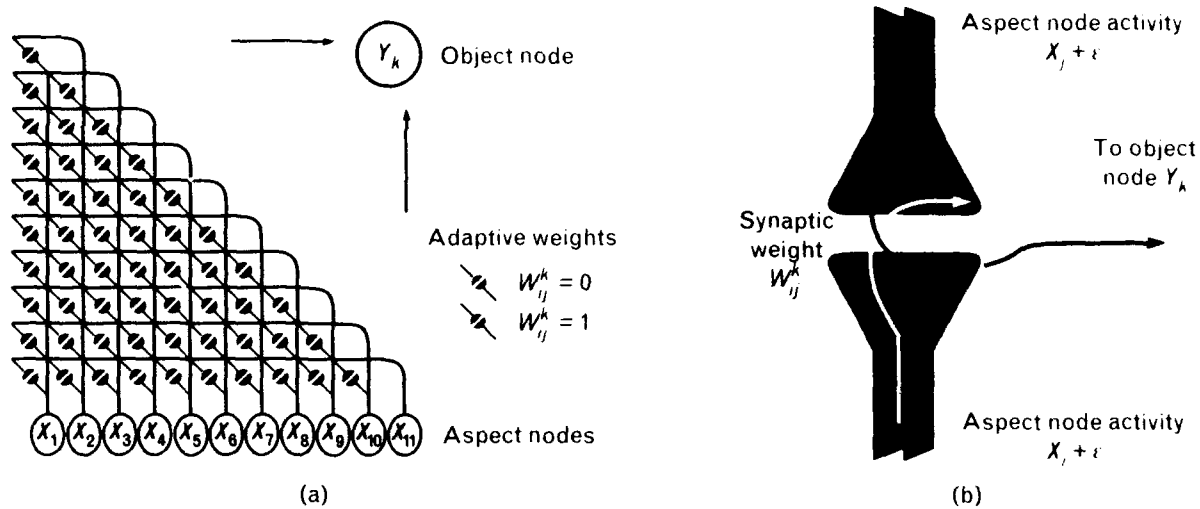
synaptic weights lie in the interval  $[0,1]$ , and, as category transitions are experienced, the weights asymptotically approach the extreme values of 0 (implying no allowed transition between corresponding categories) and 1 (indicating a permitted transition). These values correspond to the absence or presence of an arc in the associated aspect-graph representation. The dendritic tree with its synaptic connections resembles the symmetric state-transition matrices that are commonly used in system modeling techniques.

Extending the Aspect-network concept to multiple targets leads to the network architecture shown in Figure 13. In this design we consider all aspect categories of all targets as belonging to the same ART-2



**FIGURE 11.** Aspect spheres for the (a) F-18, (b) F-16, and (c) HK-1 (Spruce Goose) have been generated from 535, 530, and 423 views, respectively, of each aircraft. Feature extraction, invariant mappings, and ART-2 categorization of all 1488 views generate a total of 41 aspects, or categories, at a vigilance  $\rho$  of 0.93. The number of categories generated for the individual aircraft is 26 for the F-18, 24 for the F-16, and 28 for the HK-1. Note that many categories are common to more than one target; i.e., the light yellow in part a corresponds to the same category that is represented by the same light yellow in part b. Also note the resemblance of the aspect spheres for the two fighter aircraft, in contrast to the HK-1 aspect sphere.





**FIGURE 12.** Aspect network for the single-object case: (a) network and (b) enlarged view of one synapse of the network. The aspect nodes (blue) are each coupled to corresponding categories allocated by the ART-2 network; the nodes charge and decay like capacitors. Axons (wires) emanating from each aspect node cross each other to form a transition matrix, and each crossing has an associated axo-axo-dendritic synapse (red) onto the dendritic tree (orange) of the object node. When two aspect nodes are simultaneously active (during view transitions), they strengthen the synapse (red) via modified Hebbian learning, and conduct activity onto the dendrite toward the object node. Object nodes thus pool activity from aspect nodes, exploiting transition information to amplify this activity, thereby accumulating evidence over time. In the enlarged view, the synapse brings together activity from aspect nodes  $X_j$  and  $X_i$  (as well as a background-noise level  $\epsilon$ ) and channels it onto the dendritic tree. (Note: The box "Aspect Network Learning Dynamics" contains a description of the equations that govern the aspect nodes, object nodes, and synaptic weights. For further details of Hebbian learning and Aspect networks, see References 10, 11, and 13.)

network. The aspect categories of the ART-2 network drive a single set of aspect nodes that fan out to all the synaptic arrays of possible targets. Activity (i.e., evidence) is then channeled into the object nodes, which compete to select the target with the maximum evidence at that moment. The winning object is then able to modify its own transition array. Sudden saccadic eye/camera motions to other locations in a scene initiate a reset of object-node activities to zero; smooth tracking motions do not cause such resetting.

Figure 14 contains an example of aircraft recognition by the Aspect network. In the training sequence, each of the three model aircraft experiences an identical trajectory of 2000 views covering one quadrant of the viewing sphere. Then a test sequence of 50 F-16 images is generated, and evidence is accumulated for each of the three targets as well as for an unlearned other target representing a none-of-the-above category. The graphs shown in the figure illustrate the corresponding category (and transition) sequence, the evi-

dence accumulation and decay for each possible target, and the winning object with the instantaneous maximum evidence. Note that initially the system begins selecting the "other" target until sufficient evidence accumulates to declare the F-16 the winner, and it remains so. Reference 11 contains further details of this experiment.

At this point we have the basic design of a neural ATR system. The system has a number of definite strengths, but it also suffers from a few shortcomings. For example, a difficulty exists in adding new targets once the system has stabilized, because new data may modify the existing ART-2 category templates and lead to the need to retrain the Aspect network. A more efficient design is to assign a separate ART-2 network and (much compressed) Aspect network to each potential target, but allow the unsupervised assignment of aspect categories during the controlled exposure in a training session. By doing so, we can add new targets at a later time by simply adding new

## ASPECT NETWORK LEARNING DYNAMICS

WE DEVELOPED the Aspect network (Figures 12 and 13) as a means to fuse recognition events over time. The network embodies a hierarchical pooling of view-specific aspect categories so as to exploit the additional information associated with permitted category transitions. These transitions are learned by exploring the object.

The dynamics of Aspect networks is in the form of differential equations (shown below) governing the short-term memory activity of the aspect nodes  $X_i$  and object nodes  $Y_k$ , and long-term memory of the adaptive axo-axo-dendritic synapses  $W_{ij}^k$ . Aspect nodes are excited by their corresponding ART-2 category nodes  $I_i$  (with rate constant  $K_X$ ) and passively decay back to their resting

state (with rate constant  $\lambda_X$ ).

Object nodes accumulate evidence for each object by summing the activity (with rate constant  $K_Y$ ) entering from the aspect nodes on the dendritic tree. Activity riding atop background noise  $\epsilon$  enters via the learned synapses corresponding to permitted transitions, and activity is channeled most effectively by paired aspect nodes in a permitted sequence. The function  $\Phi_B(A)$  is a threshold linear function that passes activity levels when  $A > O(B)$ . Similar to the aspect nodes, the object nodes also decay passively to their resting state (with rate constant  $\lambda_Y$ ).

The synaptic weights learn aspect transitions by experiencing correlated activity from two as-

pect nodes, as long as the object node activity is changing (i.e.,  $\dot{Y}_k \neq 0$ ) for the winning object  $Z_k$ . The function  $\Theta_i(C)$  is a binary threshold gate that equals unity when  $C > O(\epsilon)$ . The weights approach asymptotes toward the fixed points of 0 and 1 because of the quadratic shunting terms that modulate the rate constant  $K_W$ . For further details of Aspect networks, see References 1 and 2.

### References

1. M. Seibert and A.M. Waxman, "Learning and Recognizing 3D Objects from Multiple Views in a Neural System," chap. 11.12 in *Neural Networks for Perception*, Vol. 1, ed. H. Wechsler (Academic Press, New York, 1991), pp. 426-444.
2. M. Seibert and A.M. Waxman, "Adaptive 3-D Object Recognition from Multiple Views," *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 107 (1992).

$$\text{Aspect nodes: } \frac{dX_i}{dt} = K_X I_i - \lambda_X X_i$$

$$\text{Object nodes: } \frac{dY_k}{dt} = K_Y \left\{ \sum_i \sum_{j>i} \Phi_i \left[ (X_i + \epsilon) W_{ij}^k (X_j + \epsilon) \right] - \lambda_Y Y_k \right\}$$

$$\text{Synaptic weights: } \frac{dW_{ij}^k}{dt} = K_W W_{ij}^k (1 - W_{ij}^k) \left\{ \Phi_i \left[ (X_i + \epsilon) (X_j + \epsilon) \right] - \lambda_W \right\} \Theta_i(\dot{Y}_k) \Theta_i(Z_k)$$

ART-2 and Aspect networks, without any modification to the existing networks. Moreover, separate ART-2 networks for each target better support the ATR task given only a single view (as opposed to a sequence of views), because each target will have generated its own set of learned templates within its ART-2 module. This design has been adopted for the next application—target recognition from SAR spotlight sequences. For this application we also intro-

duce a measure of *recognition confidence* derived from the accumulated evidence.

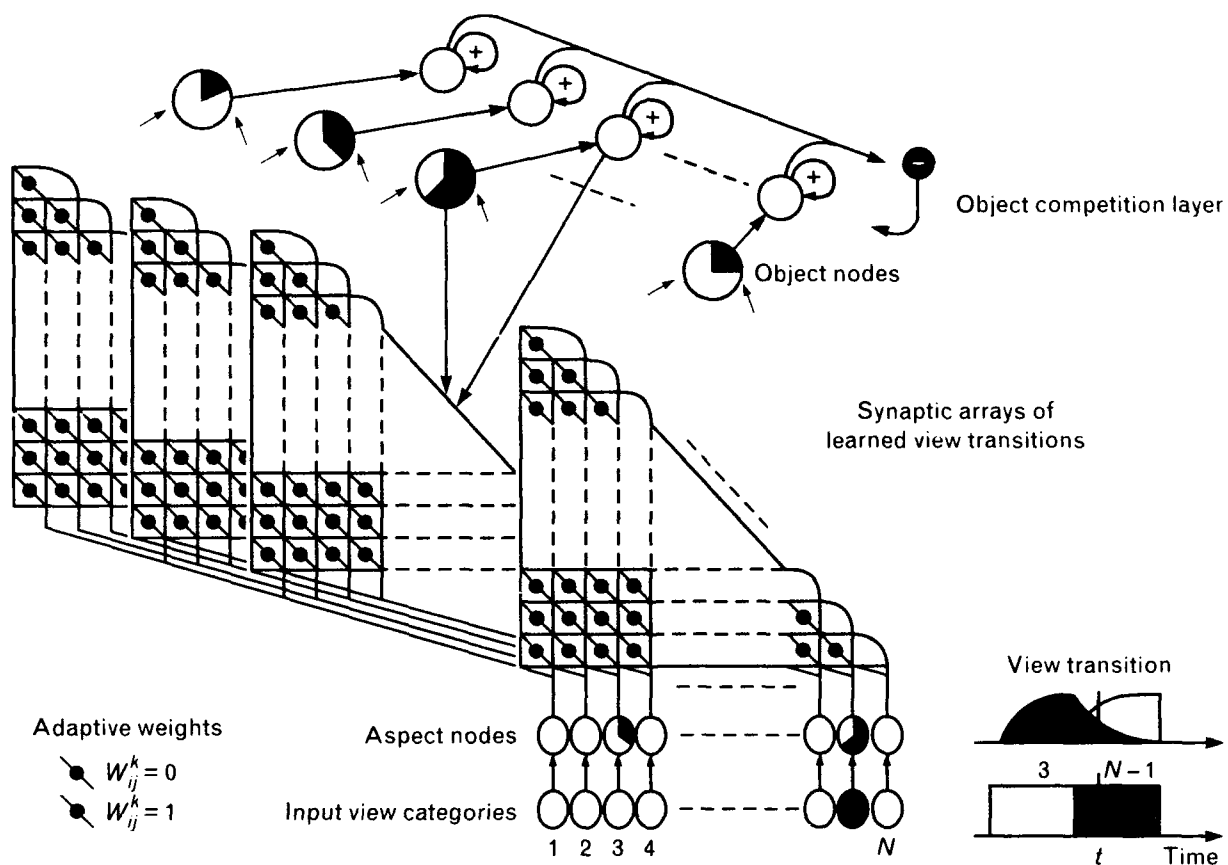
### Tactical Target Recognition in the Synthetic-Aperture Radar (SAR) Spotlight Mode

High-resolution radar imaging of a scene can be accomplished by flying a radar that is transmitting chirp pulses from many closely spaced look angles (Figure 15). The moving radar thus synthesizes a long aper-

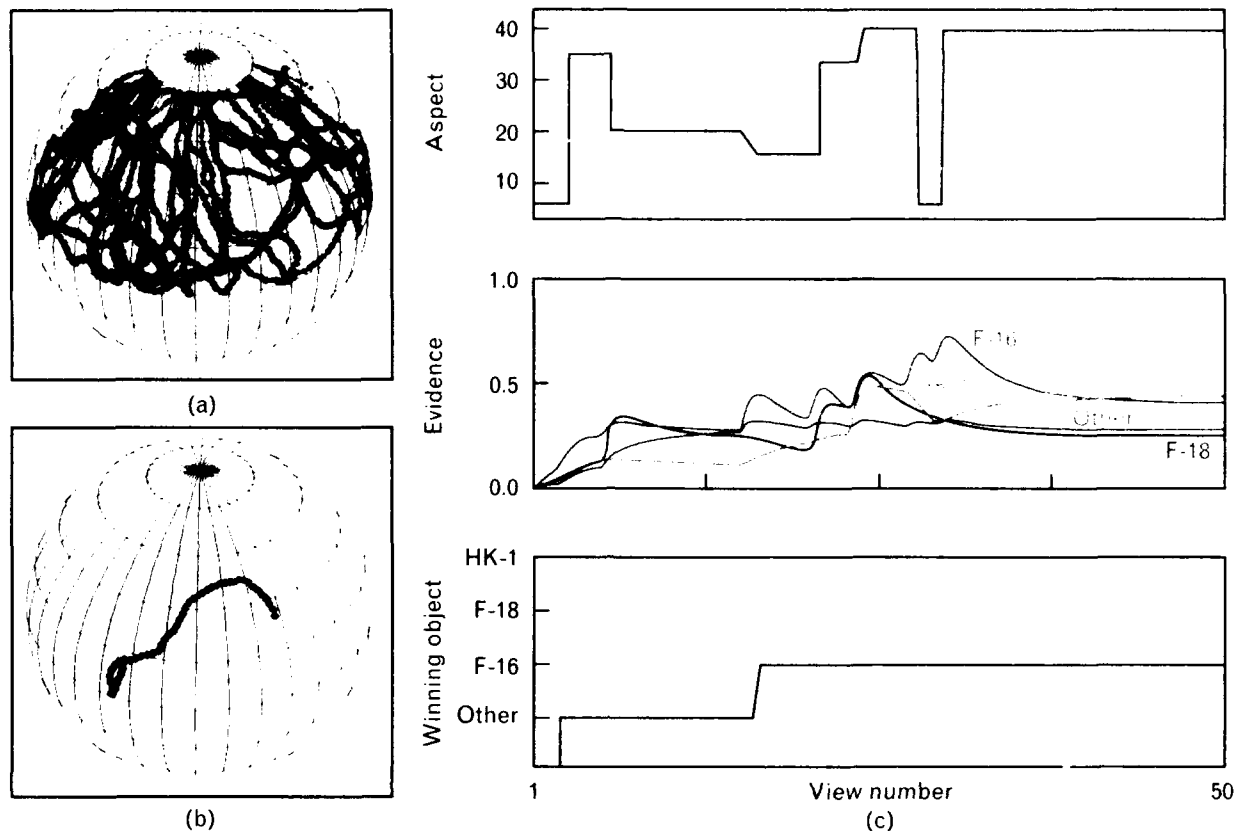
ture, and the return pulses determine a reflectivity image of the scene as projected into the range and cross-range coordinates of the plane formed by the synthetic aperture and the radar line of sight. (This plane is referred to as the *synthetic-aperture radar [SAR] slant plane*.) The range resolution is proportional to the bandwidth of the chirp pulse; the cross-range resolution is proportional to the angle subtended by the synthetic aperture. As the radar moves along the flight path, it can be "squinted" so as to track a fixed location on the ground. Hence, the radar beam *spotlights* a particular scene, and a sequence of SAR images is obtained of that scene from multiple views.

The reader may look ahead to Figure 26(a) to view a typical clutter scene—an overpass that crosses the New York State Thruway—obtained from the Lincoln Laboratory Advanced Detection Technology Sensor (ADTS), a millimeter-wave radar, operating in the SAR mode. (In our work, only single-channel vertical-vertical [VV] polarization imagery is used.) Note that the image is quite speckled, a consequence of the coherent imaging method. Nonetheless, at first glance this scene has a rather natural appearance.

To illustrate here the appearance of objects such as ground vehicles, we refer to the inverse SAR, or ISAR, images shown in Figure 16. Three tactical targets are shown at a radar depression angle (or slant-plane



**FIGURE 13.** Aspect network for the multi-object case. Input aspect categories from a single ART-2 network (coding all aspects of all targets) excite aspect nodes that fan out to all synaptic arrays of learned view transitions, each of which conducts activity (i.e., evidence) to its corresponding object node. A competition layer (created from self-excitation and collective inhibition) determines the target of maximum evidence at any moment, and allows the corresponding synaptic array to be refined. Sudden eye/camera motions can cause the object nodes to reset their evidence to zero. (For a detailed description of Aspect networks, see References 10 and 11.)



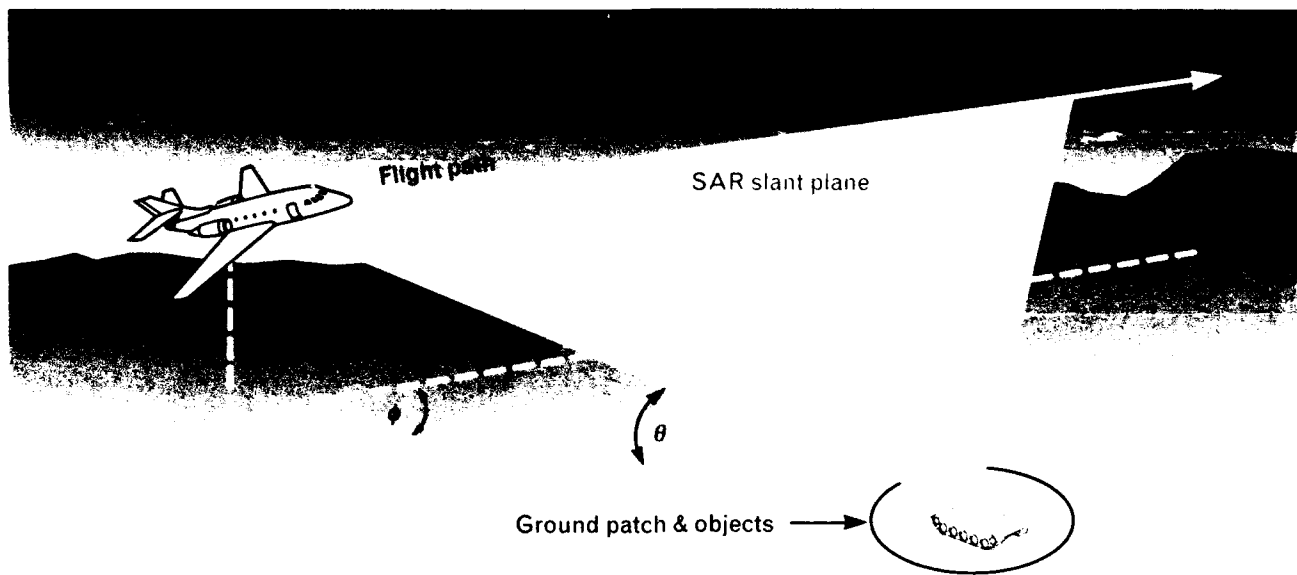
**FIGURE 14.** Example of training and recognition by evidence accumulation: (a) view sphere showing the trajectory from which 2000 views of each aircraft were used for training the system, (b) view sphere showing the trajectory from which 50 views of an F-16 were selected for testing the system, and (c) graphs showing the recognition test results. In part c, the first graph plots the sequence of aspects that were recognized by the system (note the transitions). The second graph shows the activity (i.e., evidence) of the aspect node for each aircraft target, including an unlearned target (referred to as "other"). And the final graph shows the "winning object," or target of maximum evidence at each moment. Note that the system first declares the target as "other," but then generates sufficient evidence to declare it correctly as an F-16, and that correct recognition response is maintained.

slope) of  $15^\circ$  and three azimuthal angles corresponding to front-on, intermediate, and broadside views. The images were obtained by rotating each target on a turntable in front of a stationary radar. Unlike with Figure 26(a), the man-made metallic objects in Figure 16 do not yield radar images that resemble their visible counterparts. The ISAR images are dominated by strong returns from select scattering centers on the target, sidelobe responses, and speckle noise. Both Figures 16 and 26(a) possess 1-ft resolution in range (oriented vertically) and cross-range (oriented horizontally), with the near-range (closest to the radar) located at the top of the image.

To build an ATR system that exploits spotlight-

mode SAR sequences, we can utilize many of the ideas and neural modules developed for the visible imaging domain, as presented in the preceding section. The different sensing modality of radar, however, provides us with direct range and cross-range information, and hence object size, which can be exploited in the grouping process that is used to detect potential targets. On the other hand, our earlier methods of invariant processing must be altered. In particular, the log-polar transform must be discarded because the slant-plane image is not an angle-angle image (as is obtained in passive visible or infrared imaging).

Borrowing heavily from our work in the passive



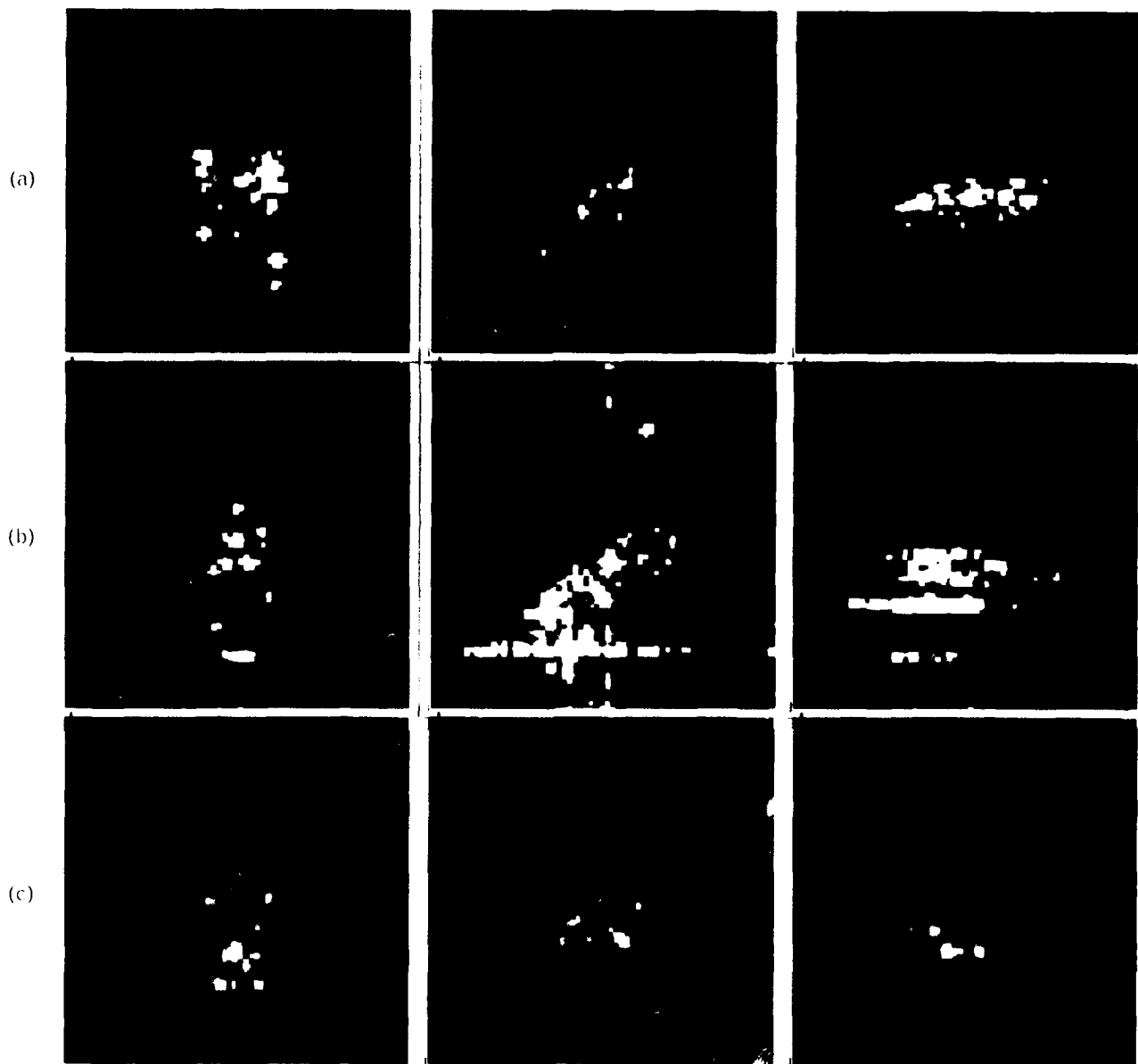
**FIGURE 15.** Imaging geometry for spotlight-mode synthetic-aperture radar (SAR). A radar on board an aircraft illuminates an area of interest on the ground by pointing at a depression angle  $\theta$  and squint angle  $\phi$ . As the aircraft flies along a straight path at altitude  $h$ , the radar transmits chirp pulses from many closely spaced look angles, and the return pulses determine a reflectivity image of the ground patch and objects of interest. Progressing along the flight path, the radar is steered to illuminate the same area of interest, and thus obtains a sequence of SAR images from multiple look angles.

visible domain, the conceptual approach to SAR target learning and recognition is summarized in Figure 17 (compare to Figure 3). Again, each image of the spotlight sequence is processed through three stages. The first stage extracts features, detects potential targets by grouping the features, and estimates the orientation of each potential target. The second stage is again responsible for the adaptive categorization of feature patterns into aspects, or categories (leading to an aspect-sphere representation of the targets). And the third stage detects aspect transitions (analogous to the arcs of a corresponding aspect graph), accumulates evidence over time, and generates a recognition decision as well as a dynamic confidence measure.

Figure 18 shows the end-to-end neural system that we have developed. A quick inspection of the modules, which are organized into three rows representing the three stages of processing of Figure 17, reveals many of the same neural networks that were used for the passive visible domain. It is also evident that we

have learned from our work in that area: we now utilize a separate ART-2 network for each target, and a separate Aspect network connected to each of these ART-2 networks. The following figures illustrate the various processing modules shown in the system diagram.

Each SAR image of the spotlight sequence is processed by the entire chain of neural modules. There are, however, several opportunities to exploit the temporal flow of information inherent in the processed data. The very first module uses shunting center-surround networks, either in isolation or as part of the Boundary Contour System (BCS) and Feature Contour System (FCS) networks for image conditioning. (BCS/FCS networks are discussed in the following section, "SAR Image Conditioning Using BCS/FCS Networks.") Figure 19 provides some details on shunting center-surround networks, as applied to SAR imagery for feature extraction. In this application, the shunting center-surround network converts the slant plane reflectance image into a locally normalized con-



**FIGURE 16.** Examples of inverse SAR, or ISAR, imagery of three tactical ground vehicles: (a) target 1, (b) target 2, and (c) target 3. The three targets are shown at three different orientations: the left, middle, and right columns of images are for azimuth angles of 0° (front-on view), 45° (intermediate view), and 90° (broadside view), respectively. The images are for a radar depression angle of 15° and vertical-vertical (VV) polarization.

trast image. Thresholds are then applied to the value at each pixel of the locally normalized contrast image, and an AND operation is used to combine the resulting image with a low-threshold version of the log reflectance input image to obtain a set of high-contrast feature blobs that can then be projected from the slant plane to the ground plane by using the known radar imaging geometry.

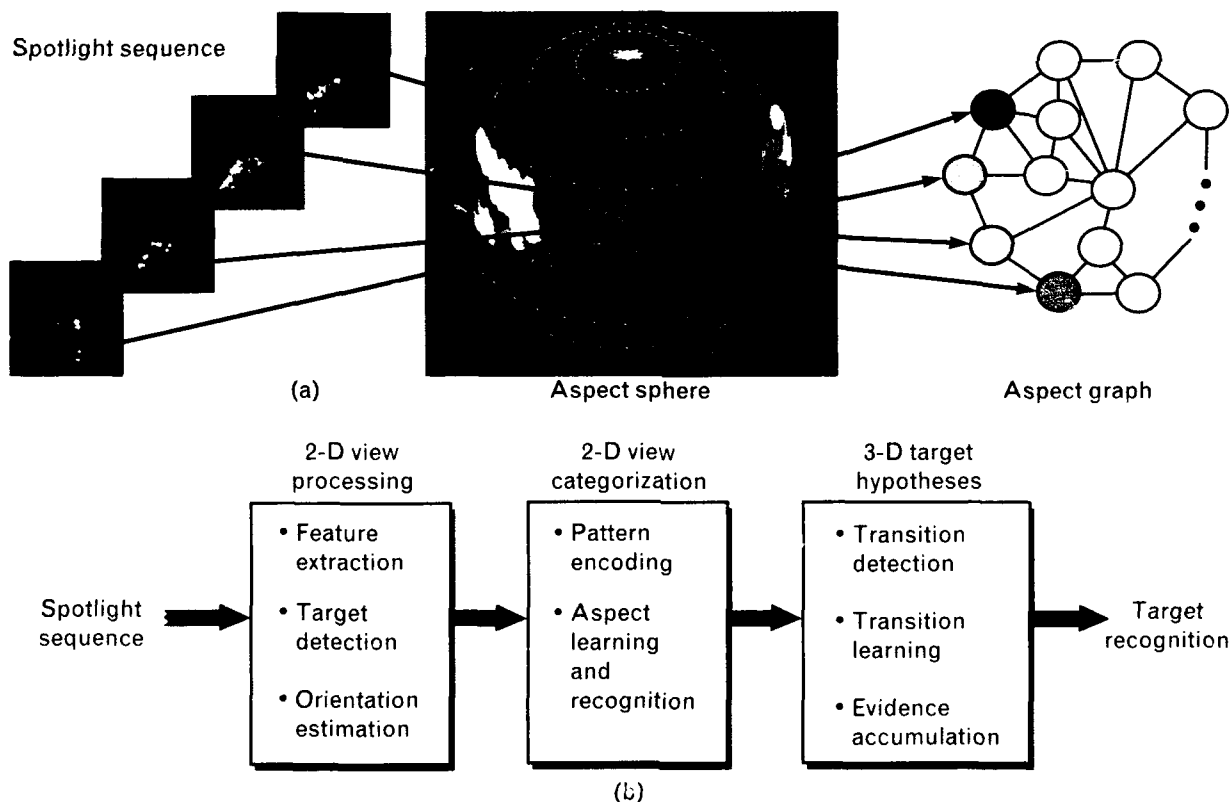
In the locally normalized contrast image, the local contrast is dependent on the choice of spatial scales for the excitatory-center and inhibitory-surround areas of the receptive field, as shown in Figure 19. These scales are chosen so as to capture the texture of scattering centers on a vehicle as compared to the vehicle as a whole. (Note: The network does *not* try to detect bright pixels on the target as compared with the sur-

rounding clutter, as is typical of constant false-alarm rate [CFAR] filtering methods.) Another advantage of using shunting networks here is that they perform an automatic gain-control operation, and, as a result, the large dynamic range of radar reflectances collapses into a predefined range in a locally adaptive fashion. These networks are modeled as dynamic membranes [13] and resemble bipolar and ganglion receptive fields in the retina.

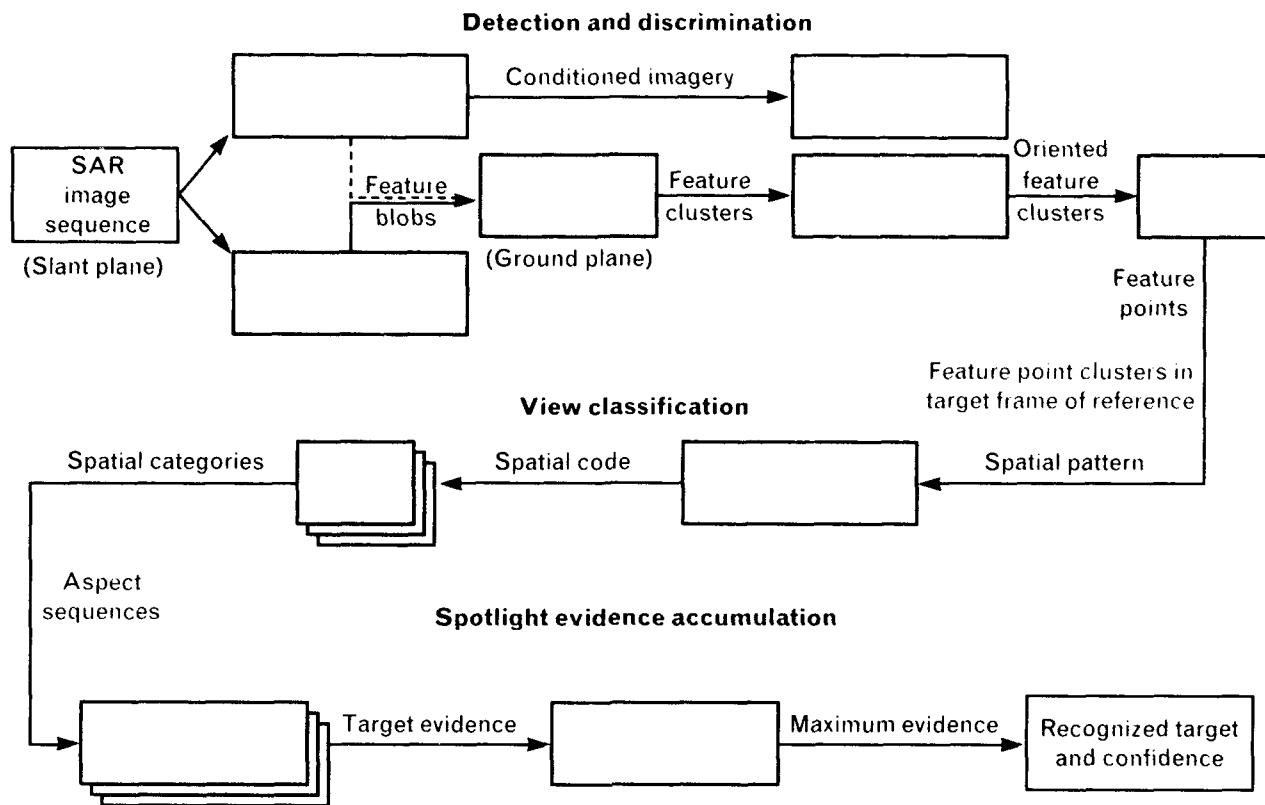
Figure 20 illustrates the four steps involved in processing an ISAR image that contains four targets. The input image is shown in the upper left quadrant, and the result of feature-blob extraction is shown in the upper right. The spatial patterns of the extracted feature blobs show strong resemblance to the scattering patterns obtained from SARTOOL simulations of radar imagery. (SARTOOL decomposes a target object into its principal scatterers and then combines the radar signatures of those scatterers.) Note that we

have discarded the original reflectance values of the feature blobs because, in practice, they can vary considerably from one instance of a target to another. In the more realistic case of targets in clutter, feature blobs generated by nontargets will also be extracted from the clutter. Thus, to simulate clutter, we added 2% random noise to the feature-blob image before proceeding with the processing. (In Figure 20 the feature-blob image is shown without the superimposition of any noise so that we could illustrate clearly the target feature blobs that emerge from the extraction process.)

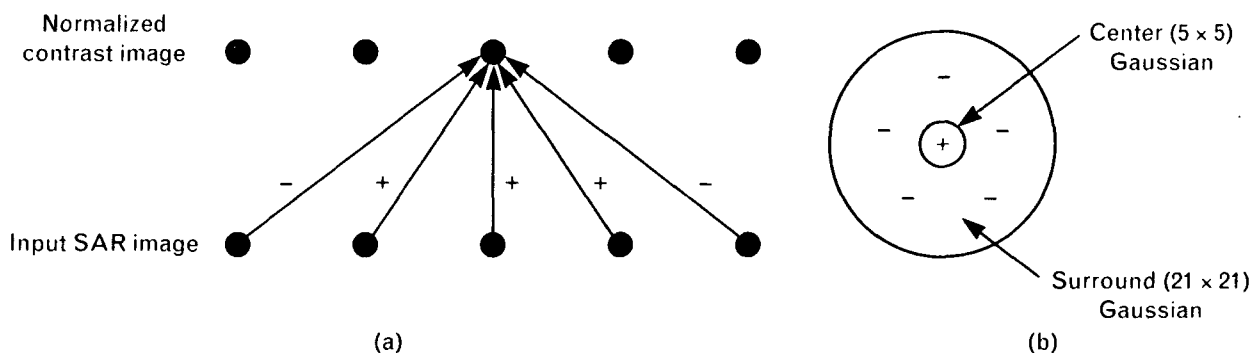
Because the image axes are measured in units of physical size, we can use the images directly to detect potential targets and discriminate them from clutter and nontarget objects by grouping the feature blobs into clusters of approximately the same image size as the targets of interest. This grouping is performed in the ground-plane coordinates, first by using an iso-



**FIGURE 17.** Conceptual approach of ATR neural system for SAR image sequences: (a) spotlight sequence of SAR images and corresponding aspect sphere and aspect graph, and (b) functional block diagram of system. Note that the approach is analogous to the approach for passive visual imagery shown in Figure 3.



**FIGURE 18.** Modular system architecture for the learning and recognition of 3-D targets from SAR imagery. The three rows of modules represent the three stages of processing shown in Figure 17. Each individual module is a neural network that transforms the imagery as indicated. From a sequence of SAR images the recognized targets generate a dynamic measure of confidence.



**FIGURE 19.** Shunting short-term memory model for feature extraction from SAR imagery: (a) center-surround feedforward architecture and (b) center-surround receptive field. The model is implemented as a feedforward dynamical system with an excitatory center/inhibitory-surround receptive field. In equilibrium the resulting image represents locally normalized contrast. The scales of the receptive field— $5 \times 5$  for the center region and  $21 \times 21$  for the surround region—are chosen to capture the contrast between scatterers and target objects. (Note: A description of the equations that govern shunting short-term memory and the equilibrium condition are given in the box "Shunting Short-Term Memory" on page 98. For further details, see Reference 13.)



## SHUNTING SHORT-TERM MEMORY

THE FULL DYNAMIC range radar image  $I_j$  serves as input to a shunting short-term memory (STM) network (Figure 19), as governed by the dynamics of a charging membrane (essentially Ohm's law). Excitatory input from a Gaussian center  $C$ , shunted in-

hibitory input from a Gaussian surround  $S$ , and passive decay (with rate constant  $\lambda_A$ ) yield an equilibrium contrast measure  $A_i$  that is normalized with respect to the local mean amplitude. The Gaussian center is weighted by  $G_{ij}^{\sigma_c}$ , and the Gaussian surround

is weighted by  $G_{ik}^{\sigma_s}$ . More general shunting networks are described in Reference 1.

### Reference

1. S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks* 1, 17 (1988).

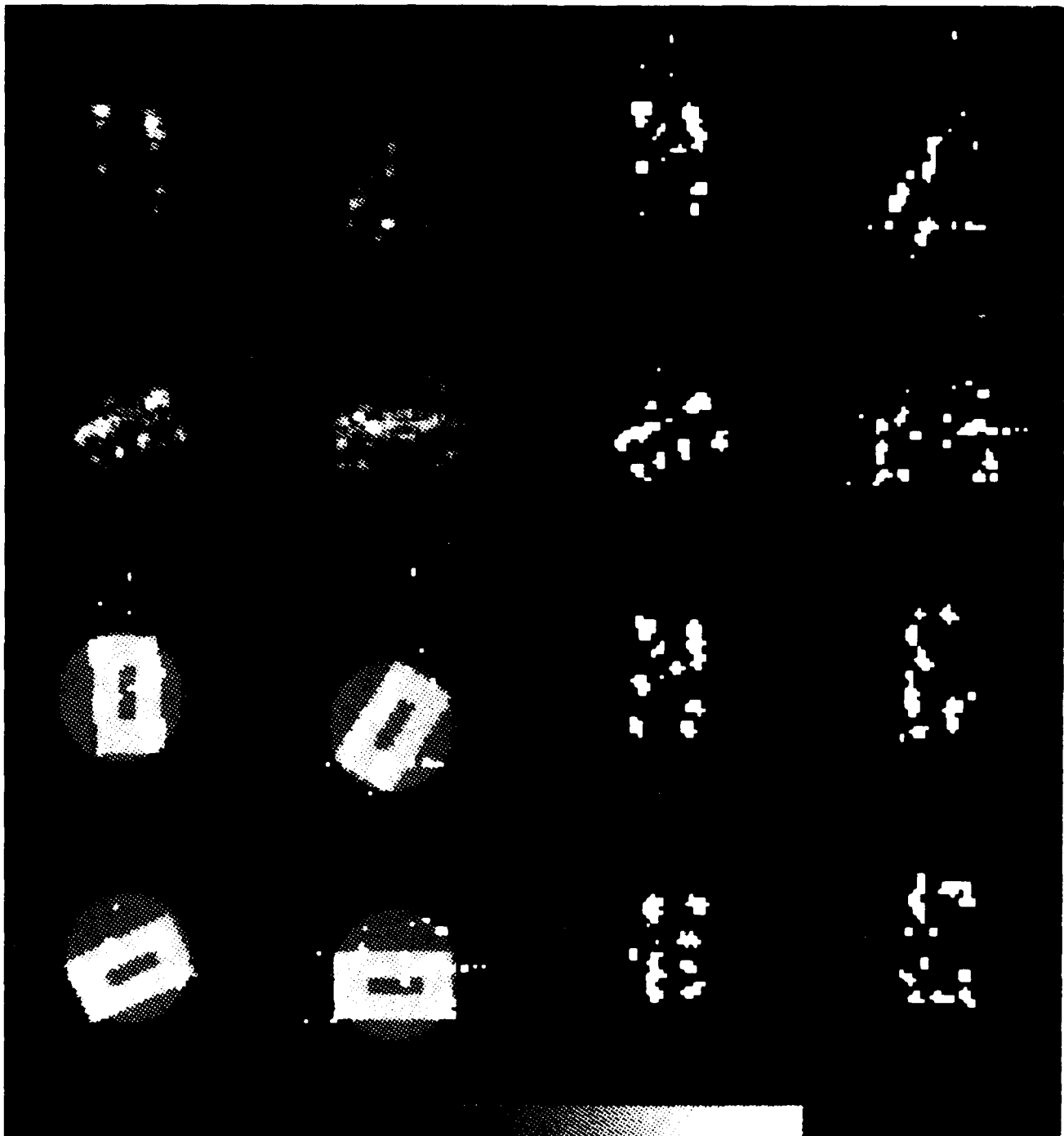
$$\text{Activity dynamics:} \quad \frac{dA_i}{dt} = -\lambda_A A_i + \sum_j G_{ij}^{\sigma_c} I_j - (1 + A_i) \sum_k G_{ik}^{\sigma_s} I_k$$

$$\text{Equilibrium contrast:} \quad A_i = \frac{\sum_j G_{ij}^{\sigma_c} I_j - \sum_k G_{ik}^{\sigma_s} I_k}{\lambda_A + \sum_k G_{ik}^{\sigma_s} I_k} \equiv \frac{C - S}{\lambda_A + S}$$

tropic receptive field (shown as circular areas in the lower left quadrant of Figure 20), and then by using oriented rectangles in the vicinity of the isotropic groupings. The rectangles are constructed from inhibitory-center/excitatory-surround receptive fields, motivated by the scatterer distributions that are typical of the targets of interest. Given a view sequence in which targets may be considered stationary as compared to the moving and squinting radar, Adaptive Linear Neurons (ADALINES) [18] performing a recursive least-squares estimation from the measurements can be used to estimate and refine the target locations and orientations. After a target has been detected and localized, it can then be segmented from the scene, rotated into a reference frame aligned with the target (with an ambiguity between whether the target is facing forward or backward), and processed by the remaining modules. The oriented feature blobs for each detected target are shown in the lower right quadrant of Figure 20. Note that sidelobe responses outside the targets have been discarded.

Figure 21 illustrates the processing of an individual

target. The input slant-plane imagery is shown in the upper left quadrant of the figure, and the localized target feature blobs are shown in the upper right. After the features are reoriented in a frame of reference with respect to the target, a DEB network is used to reduce the features to points, as shown in the lower left quadrant. This oriented spatial pattern of feature points covers an extent of approximately  $20 \times 30$  pixels at 1-ft resolution. Moreover, as the target orientation and radar depression angle change, this pattern must change quickly, too. Of equal importance is the deformation of this feature pattern with varying radar squint angle (given an identical depression angle and target orientation with respect to the radar). For these reasons, a template constructed from this feature pattern (no less a template that incorporates the original reflectance values) is both memory intensive and fraught with difficulties. Thus we can again utilize the large overlapping receptive fields (cf. Figure 7) to reduce this binary feature pattern to a  $9 \times 9$  array of analog numbers that code the spatial distribution of features in a compressed manner that is robust to



**FIGURE 20.** Multitarget localization and orientation estimation is illustrated for the case of four tactical targets. The upper left quadrant shows the input slant plane imagery for the composite of four targets at a depression angle of  $15^\circ$  (Note: In each of the four quadrants, the four targets are arranged with target 1 in the upper left, target 2 in the upper right, target 3 in the lower left, and a modified version of target 1 in the lower right.) The upper right quadrant contains the feature blobs that have been extracted and projected in the ground plane. Each target is then grouped with a circular mask, and the target's orientation is estimated with an inhibitory center-excitatory surround oriented rectangular mask, as shown in the lower left quadrant. This processing allows the detected targets to be reoriented in a target frame of reference, as has been done in the lower right quadrant. The color bar at the bottom of the figure denotes increasing (from black to white) reflectivity for the imagery in the upper left quadrant.

spatial deformation. Such a coding is illustrated in the lower right quadrant of Figure 21.

The  $9 \times 9$  array of 81 numbers forms the input to an ART-2 network that is dedicated to the learning of a particular target. The target is learned during a training session in which the target exposure is controlled. In a testing session, the system is run in recognition mode, and the 81-member spatial code vector is fed to all ART-2 networks representing all targets of interest. Figure 22 illustrates the results of training independent ART-2 networks on each of the three ISAR targets of Figure 16. The resulting aspect categories that were established are shown color coded on aspect spheres seen both from the side with the targets facing left, and from above (compare to Figure



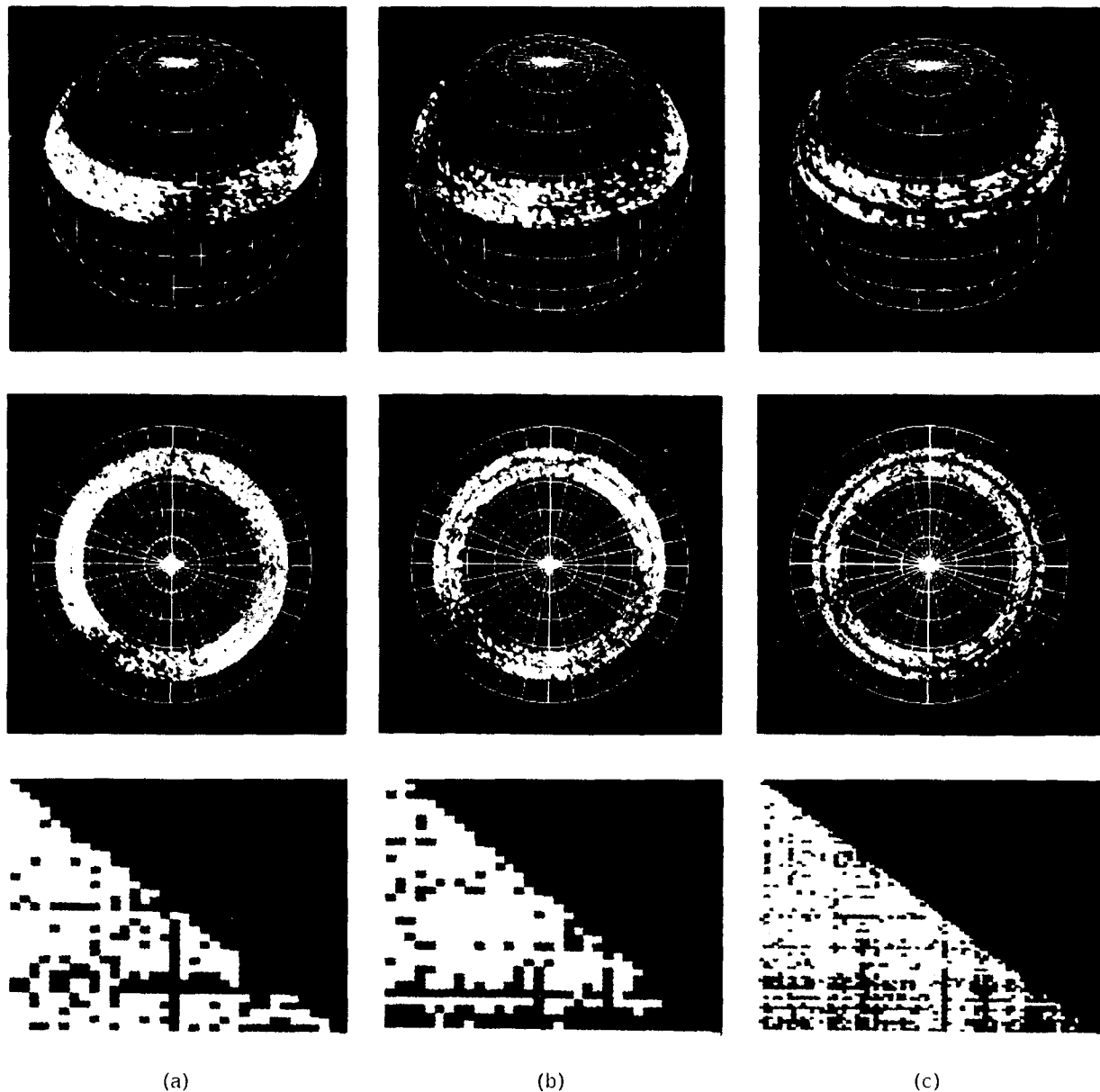
**FIGURE 21.** Target feature extraction and spatial coding for a single target. The upper left quadrant shows the input slant-plane image of a target at a depression angle of  $15^\circ$ . In the upper right quadrant, feature blobs have been extracted from the input image and projected in the ground plane. After the feature blobs are reoriented in the target frame of reference, a DEB network is used to reduce the blobs to points, as shown in the lower left quadrant. The feature points are then coarsely coded by a  $9 \times 9$  array of overlapping receptive fields (cf. Figure 7), as shown in the lower right quadrant. The color bar at the bottom of the figure denotes increasing (from black to white) reflectivity for the upper left image, and increasing receptive-field activity for the lower right image.

11). The data consisted of ISAR images created at all even azimuths  $360^\circ$  around each target, for radar depression angles between  $15^\circ$  and  $32^\circ$ , comprising approximately 3000 views of each target. (Note the missing data at a few intermediate depression angles for targets 2 and 3.) The resulting unsupervised classification generated islands of common category that extended over large azimuthal extents and many depression angles. We purposely changed the vigilance parameter setting between targets to emphasize the user's control over the fineness of categorization. With a finer categorization, more details survive the learning process. The roughly 3000 views of each target have been compressed into only 34 categories for targets 1 and 2, and 75 categories for target 3, for which we used the highest vigilance setting.

Associated with each category allocated by each ART-2 network is a template of the prototype  $9 \times 9$  array (contrast enhanced and normalized) that was learned by the synapses (the adaptive LTM sites) in the network, as shown in Figure 8. Eight of the principal templates for target 1 are illustrated in Figure 23, along with sample slant-plane images from the corresponding viewing directions. Note that the learned templates include two broadside views, frontal and end-on views, and the characteristic L-shapes near the four corner views. These color patterns code prototype spatial feature patterns (*not* reflectance patterns).

To complete the learned description of each target, the permitted transitions among aspect categories must be detected and imposed on the synapses of each Aspect network. The result of this process is contained in the last row of photographs in Figure 22. The photographs show the transition matrices for each target (cf. Figure 13). In the matrices, a red pixel corresponds to a permitted transition while a green pixel codes the absence of such a transition.

Figure 24 contains an example of our ATR system running in recognition mode. We used an ISAR image sequence that consisted of 45 views of target 1 (only the odd azimuths in the interval  $67^\circ$  to  $157^\circ$ ) at a depression angle of  $21^\circ$ . (Note: Although this dataset was not part of the original training set, it was admittedly not very different from the training data. This lack of adequate training and test datasets is a prob-



**FIGURE 22.** Aspect spheres and transition matrices for (a) target 1, (b) target 2, and (c) target 3. The spheres were generated with independent ART-2 networks. For each target, approximately 3000 views, collected for even azimuths  $360^\circ$  around and depression angles from  $15^\circ$  to  $32^\circ$ , have been compressed into 34, 34, and 75 categories for targets 1, 2, and 3, respectively. The categories, or aspects, have been assigned colors and are shown on a viewing sphere from the side with the target facing left (top row of photographs), and from above (center row of photographs). Note the category islands that emerge over large viewing extents, particularly for target 1. The fineness of categorization is controlled by the vigilance parameter of the ART-2 network, and can be chosen to be more or less sensitive to variations in the feature patterns. The vigilance  $\rho$  is 0.97, 0.98, and 0.99 for targets 1, 2, and 3, respectively. The last row of photographs shows the category transitions that were learned for each target by independent Aspect networks coupled to each ART-2 network. Each transition matrix codes possible category transitions in red, while green denotes the absence of such a transition (cf. Figure 13). Note that the transition patterns are quite different among the targets. Detected transitions between categories contribute to the evidence accumulation during the recognition process.

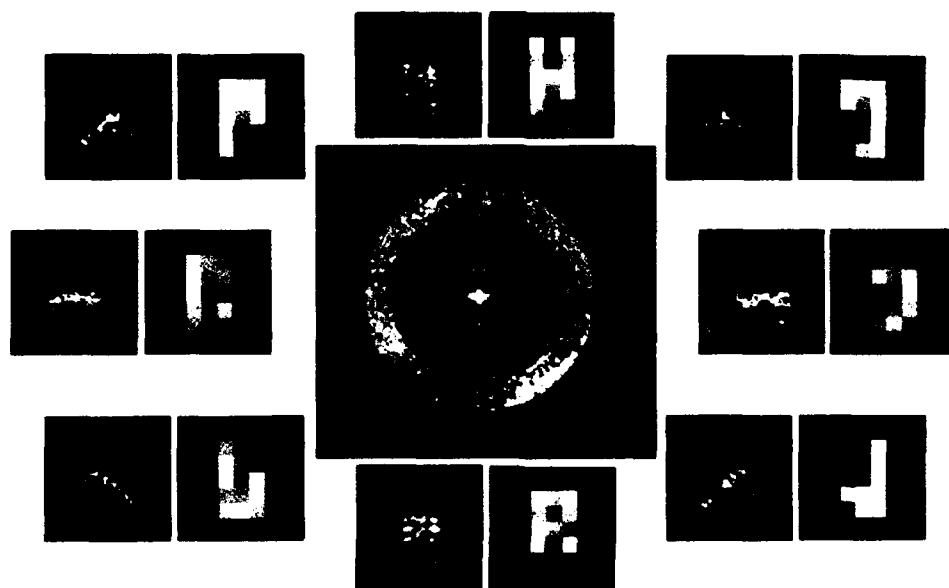
lem with many ATR studies.) The test imagery was passed through the early modules of our system, and then to four ART-2 networks (with learning turned off) coupled to four Aspect networks corresponding to the training targets 1, 2, and 3, in addition to a fourth unlearned target (referred to as "other") that was represented by random synaptic weights. Each ART-2 network determined the best matching aspect category for the test target, and each ART-2 network activated its corresponding Aspect network to accumulate evidence over the test view sequence. A competition between the object nodes in the different Aspect networks then selected the target with the instantaneous maximum evidence.

In Figure 24, the category sequence recognized by the ART-2 network for target 1 is illustrated both on an aspect sphere seen from above, and as a graph of category versus view number. The second graph in Figure 24(b) plots the accumulation of evidence for each target, while the third graph indicates the selected target with the maximum evidence accumulated at each view. Although the selected target is target 1, we can see that target 3 also accumulates a

significant amount of evidence. Thus selecting target 1 solely on the basis of maximum evidence can be risky because the evidence for target 1 may exceed that for target 3 by only a slight amount. This possibility suggests looking at the *differential evidence* between the two targets of highest accumulated evidence, as illustrated in the fourth graph. The differential evidence may be small for some views, but it too can be integrated along the temporal view sequence, giving rise to a *dynamic confidence measure*. As shown in the bottom graph of Figure 24, the confidence measure increases monotonically along the view sequence in this example. It is a matter of preference to select the threshold level of confidence that the system should use in declaring a target as recognized. Clearly, the number of views required to reach this confidence threshold will depend on the target itself, as well as the starting view in a sequence.

### SAR Image Conditioning Using BCS/FCS Networks

We have already noted that single-channel SAR imagery is characterized by a very large dynamic range and



**FIGURE 23.** Aspect sphere, example typical views, and corresponding learned templates for target 1 of Figure 22(a). The learned templates include two broadside views, frontal and end-on views, and the characteristic L-shapes near the four corner views. Note the ability of the ART-2 network to quantize the viewing space around a target in an unsupervised fashion. The learned templates are then used for the recognition process.

excessive speckle noise, and man-made objects possess rather broken signatures that vary rapidly with small changes in viewing angle. To a great extent, we can alleviate these problems by first conditioning the imagery with the *Boundary Contour System* and *Feature Contour System* (BCS/FCS) network paradigm developed by S. Grossberg, E. Mingolla, and D. Todorović, (see chapters 1 to 4 in Reference 19). This neural processing architecture is strongly motivated by the known anatomy and physiology of the early visual processing stages, including that of the retina, LGN, V1, V2, and V4. The architecture, which essentially incorporates a general theory of preattentive vision, has also been quite successful in explaining a very

large body of psychophysical perceptual data.

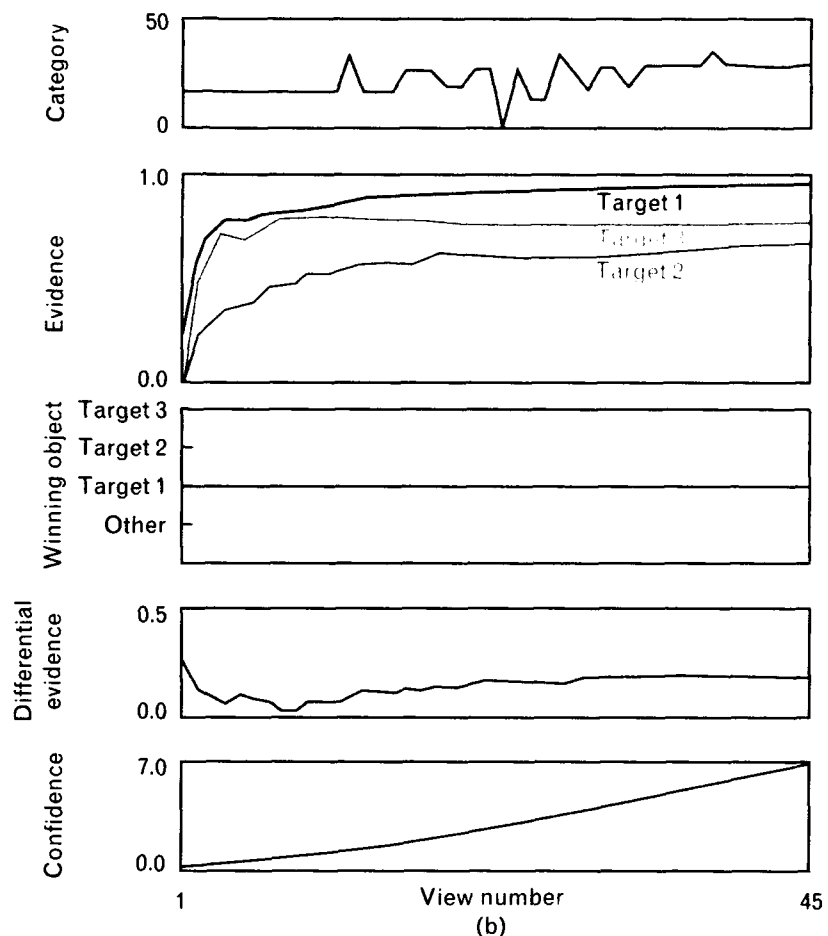
The BCS/FCS networks are shown as an alternative first module in the ATR system of Figure 18. Our preliminary work indicates that the initial processing of SAR imagery with BCS/FCS networks, in lieu of a shunting center-surround network, improves the target detection (and false-alarm rejection) process.

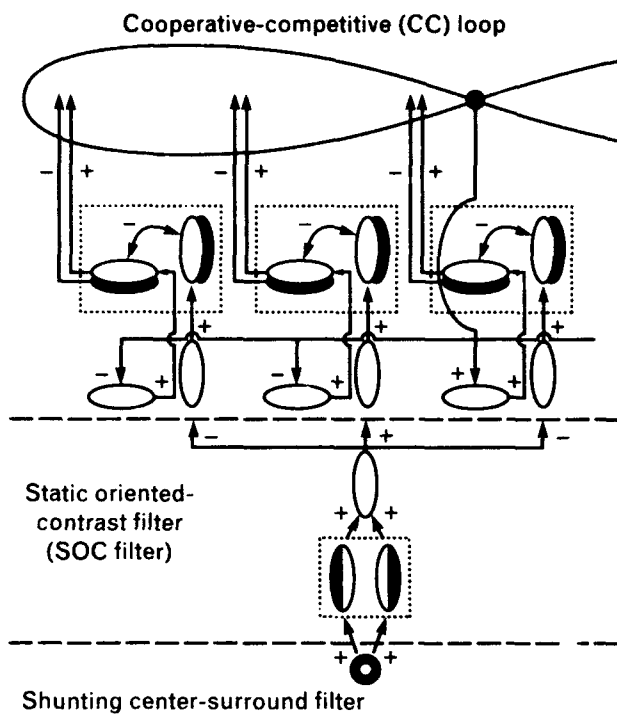
Preattentive vision, in its simplest form, is a computational process in which contours are contextually established and the perceived brightness (and color) is generated primarily from local-contrast information. In BCS/FCS theory, the role of the BCS network is to establish such contours in the context of local fields of



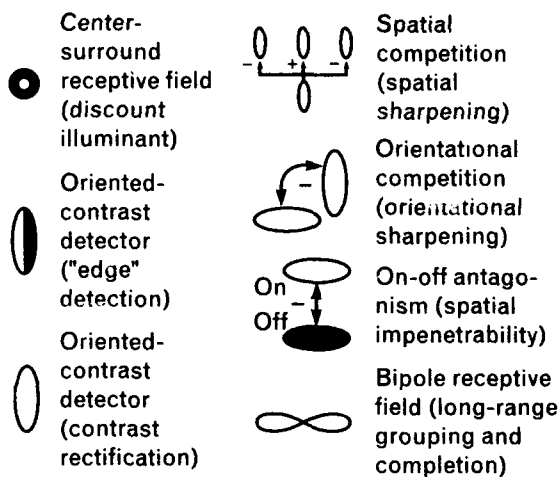
(a)

**FIGURE 24.** Example of target recognition by evidence accumulation: (a) aspect sphere and (b) recognition results. The ISAR image sequence used consists of 45 views of target 1 (only the odd azimuths in the interval  $67^\circ$  to  $157^\circ$ ) at a depression angle of  $21^\circ$ . The category sequence recognized by the ART-2 network for target 1 is represented both on an aspect sphere seen from above in part a, and as a plot of category versus view number, as shown in the first graph of part b. The next graph shows the evidence generated by the resulting category matches for the training targets 1, 2, and 3. The third graph indicates the "winning object," i.e., the selected target with the maximum evidence accumulated at each view. The target with the instantaneous maximum evidence is consistently target 1, although target 3 also has a strong response. The differential evidence between those two targets is plotted in the fourth graph, and integrated across view numbers in the fifth graph to generate a monotonically increasing confidence measure.

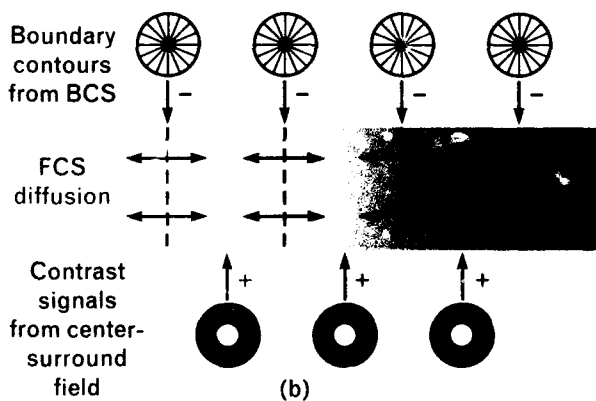




Legend



(a)



edge fragments, or oriented contrast. Although such *boundary contours* are themselves invisible, they modulate the dynamics of a diffusive filling-in process in the FCS network whereby local contrast and brightness information mix and spread within such boundaries to create a smoothly shaded figure.

The architecture of the BCS network is illustrated in Figure 25(a). Beginning with monocular preprocessing in the form of shunting center-surround receptive fields, local measures of normalized isotropic contrast are made. An oriented-contrast filter then derives evidence for local *edge fragments*, which are then used as input to a *cooperative-competitive (CC)* feedback loop. The CC loop performs long-range completion of contours in the context of the local edge statistics. We have found that one pass through the CC loop is typically sufficient for our purposes. The boundary contours obtained from the BCS network provide input to the FCS filling-in network, along with the center-surround contrast signals, as shown in Figure 25(b). Essentially, the contrast signals try to spread diffusively to neighboring nodes, but the BCS signals modulate the local diffusivity

**FIGURE 25.** Architecture of the (a) Boundary Contour System (BCS) of S. Grossberg and E. Mingolla and the (b) Feature Contour System (FCS) of Grossberg and D. Todorović. The BCS architecture in part a models the neurodynamics of preattentive visual processing in the LGN, V1, and V2 visual areas of the brain. Shunting center-surround receptive fields provide input to oriented-contrast, or edge, detectors that compete across position and orientation. The resulting local edge fragments are grouped over large distances by oriented bipole receptive fields that feed back to the oriented-contrast detectors to complete broken boundaries comprising the edge fragments. The boundary contours obtained from the BCS network provide input to the FCS network, which uses the information to modulate the local diffusivity between compartments, as shown in part b. The local diffusivity between compartments affects the FCS diffusion layer, where the contrast signals from the shunting center-surround network spread laterally in two dimensions. In the diffusion layer, strong boundary contours inhibit diffusion across the boundaries. The FCS architecture models hypothesized filling-in interactions in the V4 visual area of the brain. (Adapted from Figures 15 and 17 of chapter 1 in Reference 19, with permission. This reference also contains a detailed description of BCS/FCS networks.)

such that strong boundary contours inhibit diffusion across the boundaries. Thus the boundary contours impede the spreading contrast signals. In the presence of a dense web of boundary contours of varying strength, the FCS diffusion process results in smoothly shaded images while retaining sharp transitions in brightness.

In applying BCS/FCS processing to SAR imagery, various parameters in the governing dynamical system need to be selected so that the pixel values are not discounted completely (because the original SAR image brightnesses are actually reflectance measures). We can then preserve the ordering of the resulting brightnesses in fairly uniform areas so as to mimic the ordering of the initial reflectance values. In nonuniform areas, however, the resulting signals indicate a mixture of reflectance and local contrast. The overall effect is SAR imagery with significantly less speckle noise, darkened and sharpened shadows, and more smoothly shaded signatures. Figure 26(a), obtained with the Lincoln Laboratory ADTS SAR, illustrates a clutter scene of trees, roads, and an overpass that crosses the New York State Thruway. The image was obtained with single-channel VV polarization at 1-ft resolution. Because of the large dynamic range, the scene is displayed as a log-amplitude image. Figure 26(b) shows the same scene after BCS/FCS processing of the full-dynamic-range SAR image. Note the dramatic reduction in speckle, the darkening of shadows, the sharpening of shadow contours, and the smooth shading of the treetops, roads, and grass.

Figure 27 illustrates the various stages of processing for the three ISAR targets oriented at a  $45^\circ$  azimuth and a  $15^\circ$  radar depression angle. The log-amplitude ISAR imagery is shown in the first column, the contrast-enhanced output of the shunting center-surround network is contained in the second column, the boundary contours derived from the BCS network are given in the third column, and the smoothly shaded signatures obtained from the FCS network are in the fourth column. An important attribute of the FCS filled-in signatures is that they are quite stable with respect to small changes in target orientation.

For ATR applications involving SAR imagery, BCS/FCS processing is a useful image-conditioning proce-



(a)



(b)

**FIGURE 26.** SAR image conditioning with BCS/FCS networks: (a) original SAR image of an overpass that crosses the New York State Thruway and (b) image after BCS/FCS processing. The original single-channel VV-polarization image (shown as log amplitude of the reflectance) is corrupted by speckle noise, which results in many false alarms making target detection difficult. In the BCS/FCS-processed image, note the reduction of speckle noise, the darkening of shadow areas, and the crispness of the shadow contours. Such image conditioning improves target detection while reducing false alarms. The SAR image was obtained with the Advanced Detection Technology Sensor (ADTS), a Lincoln Laboratory millimeter-wave radar.



ture. We expect it to improve both the target detection and recognition stages of our ATR system.

### **Reentry-Vehicle Recognition from ISAR Sequences**

We have also applied our SAR target-recognition system to the identification of reentry vehicles imaged by a ground radar while the vehicles were spinning and traveling along a trajectory. The resulting reentry-vehicle images are thus ISAR imagery, although they are in general simpler than that obtained with tactical targets in clutter. Radar processing is typically done in the range-Doppler domain to extract peaks corresponding to isolated scattering centers on the vehicle's shroud. We can then apply to these data the same three stages of processing that we applied to the ISAR tactical-target imagery: the range-Doppler peaks can be used as point feature patterns, the patterns can be encoded by overlapping receptive fields followed by classification with ART-2, and evidence and confidence can be accumulated with the Aspect network. (Note: An alternative approach to the learning and recognition of reentry vehicles has recently been reported by A.M. Aull et al. [20].)

With this approach in mind, we have constructed ISAR imagery of point scatterers for three reentry vehicles over several rotations at a single angle of attack (i.e., a single depression angle). The vehicles are designated as RV-1, RV-2, and RV-3. Figure 28 illustrates the result of coding and ART-2 category learning for vehicle RV-2 at a vigilance setting of 0.95. Aspect categorization over multiple rotations are shown on an aspect sphere, along with the learned templates and typical feature patterns for the six categories that ART-2 established. Figure 29 shows the results of separate ART-2 categorizations for all three reentry vehicles over multiple rotations, as well as the learned transitions among aspect categories used by the Aspect network. The three vehicles differ in their complexity, which is reflected by the number of categories required by ART-2 to cluster the data: 3, 6, and 16 categories for RV-1, RV-2, and RV-3, respectively.

The results of a recognition experiment are shown in Figure 30 (compare to Figure 24). In the experiment, a sequence of views of vehicle RV-3 was input

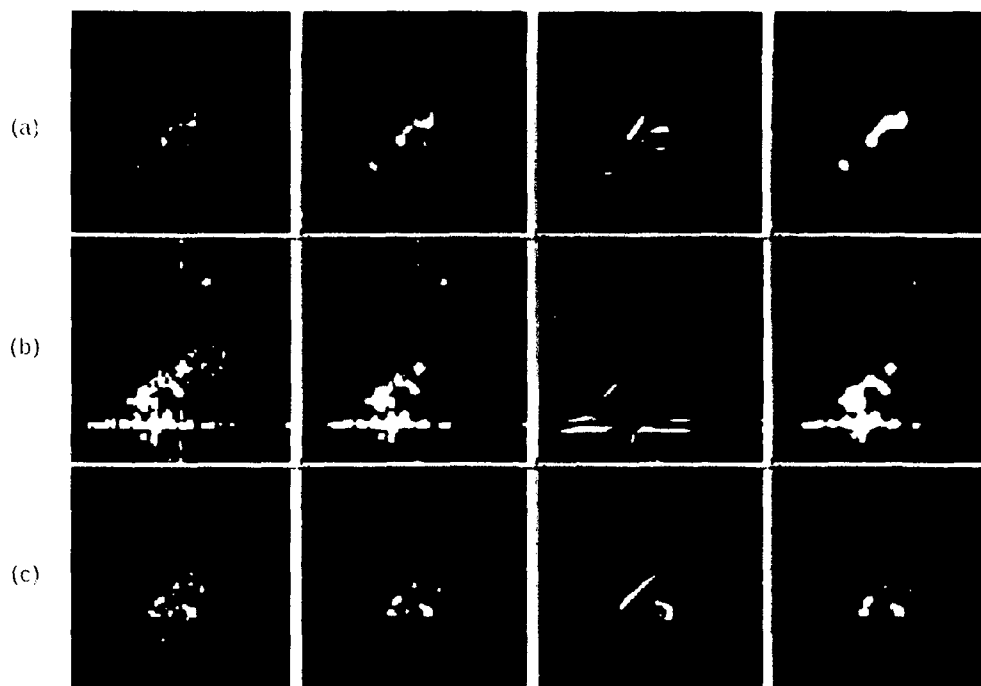
to the system. (Note: This sequence was not part of the data used for training the system.) Again, evidence was accumulated for all three targets in addition to an unlearned target, the target of maximum evidence was chosen, differential evidence was computed from the two targets of highest evidence, and the difference was integrated along the view sequence to generate a confidence measure. In Figure 30 we see an example of the system changing its selection. The system first (correctly) chooses RV-3 although the confidence is still relatively low, then the system gets confused and switches between the other two vehicles. The switching resets the confidence to zero, and it remains very small due to the small differential evidence generated. Finally, the system locks back onto the correct decision, and confidence builds monotonically.

Table 1 (page 109) summarizes the results of preliminary recognition experiments on these three reentry vehicles. In each case the test sequence consisted of 90 images starting at randomly selected azimuths. In all cases the correct vehicle was recognized, and fewer than 25 images were required in each sequence to converge to a high-confidence correct decision. By converting this result to the fraction of each vehicle's rotation cycle that is required to achieve such recognition, we find that fewer than two revolutions were required in each case.

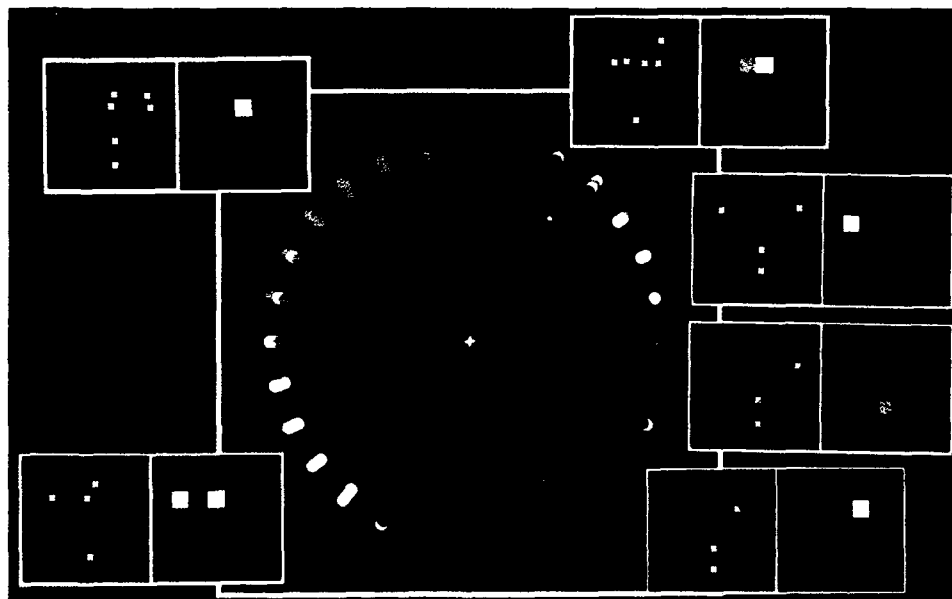
### **Learning and Recognition Using Salient Object Parts**

The 3-D object learning and recognition system described thus far processes the views of objects as a whole. But this approach can lead to a decline in recognition ability when an object is partially occluded or disguised, or when a part of the object is articulated or variable (removed or replaced). To deal with these situations, we return to biology for guidance.

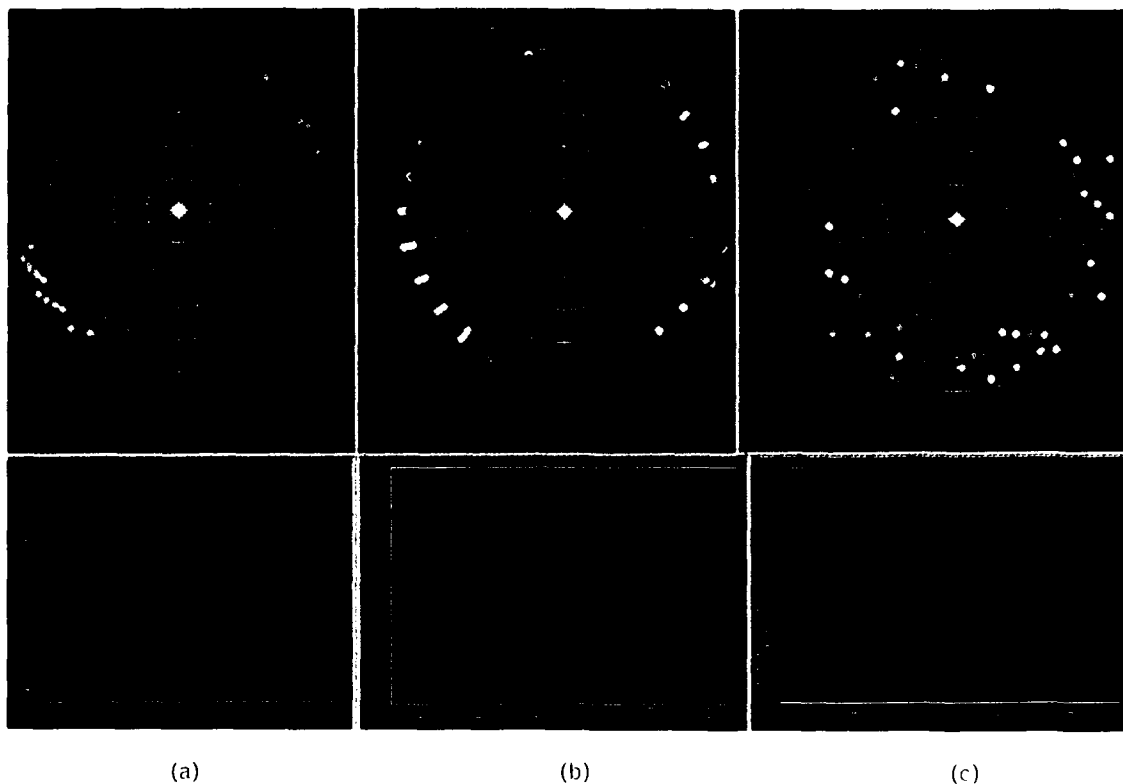
The brain processes information by using a *principle of contrast*. Many operations seem to be cast in terms of differences, or in terms of the detection of novelties or transitions in space, time, or patterns. Mechanisms exist that detect novel changes, as reflected by the peak in EEG measurements that occurs 300 msec after the introduction of an unexpected



**FIGURE 27.** BCS/FCS processing applied to the ISAR images of (a) target 1, (b) target 2, and (c) target 3. From left to right, the columns show different stages of the BCS/FCS processing. The ISAR imagery (first column) is contrast enhanced by a shunting center-surround network (second column) and boundary contours are extracted (third column). The contrast-enhanced imagery diffuses within the boundary contours to produce filled-in target signatures (fourth column). All three targets are oriented at a 45° azimuth and a 15° radar depression angle.



**FIGURE 28.** Learned categories for reentry vehicle RV-2 are plotted on an aspect sphere over four rotations of the target. (For convenience, the data for each of the four rotations have been plotted on the aspect sphere at different shifted depression angles. Note the four rounds of colored dots on the sphere.) During the learning process, ART-2 generated only six categories (at a vigilance setting of 0.95). The learned templates along with the representative scatterer patterns for the six categories are shown. From the upper right corner of the overall figure, the corresponding colors for the learned templates are dark brown, dark blue, green, light blue, white, and light brown.



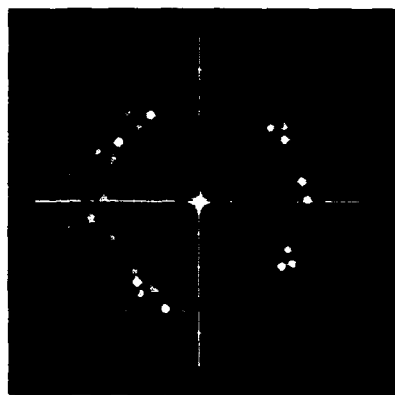
**FIGURE 29.** Learned categories (aspect spheres) and transition matrices for the (a) RV-1, (b) RV-2, and (c) RV-3 reentry vehicles over multiple rotation cycles. The vigilance setting is 0.95, 0.95, and 0.96 for the three reentry vehicles, respectively, and the resulting number of categories established is 3, 6, and 16, respectively. (The difference in the number of categories reflects differences in the complexities of the three vehicles.) In the transition matrices, the possible category transitions are coded in blue, while red denotes the absence of such a transition.

new stimulus. Other mechanisms are responsible for suppressing information that is not changing. For instance, stabilized retinal images fade away in about one second. In fact, all sensory systems become habituated to constant or repetitive input patterns. Indeed, human vigilance decreases after long periods of waiting, and we become bored. This principle of contrast has been exploited earlier in our system in the form of center-surround receptive fields, edge detectors, competitive learning, view-transition detection, and confidence estimation via differential evidence. We now use the principle again, this time as a foundation for *Saliency Maps*, hierarchical object-part representations, and caricature-based recognition [21].

Visual attention not only focuses processing power on an object in a scene, it often isolates only a part of the object for closer inspection. (Note: Evi-

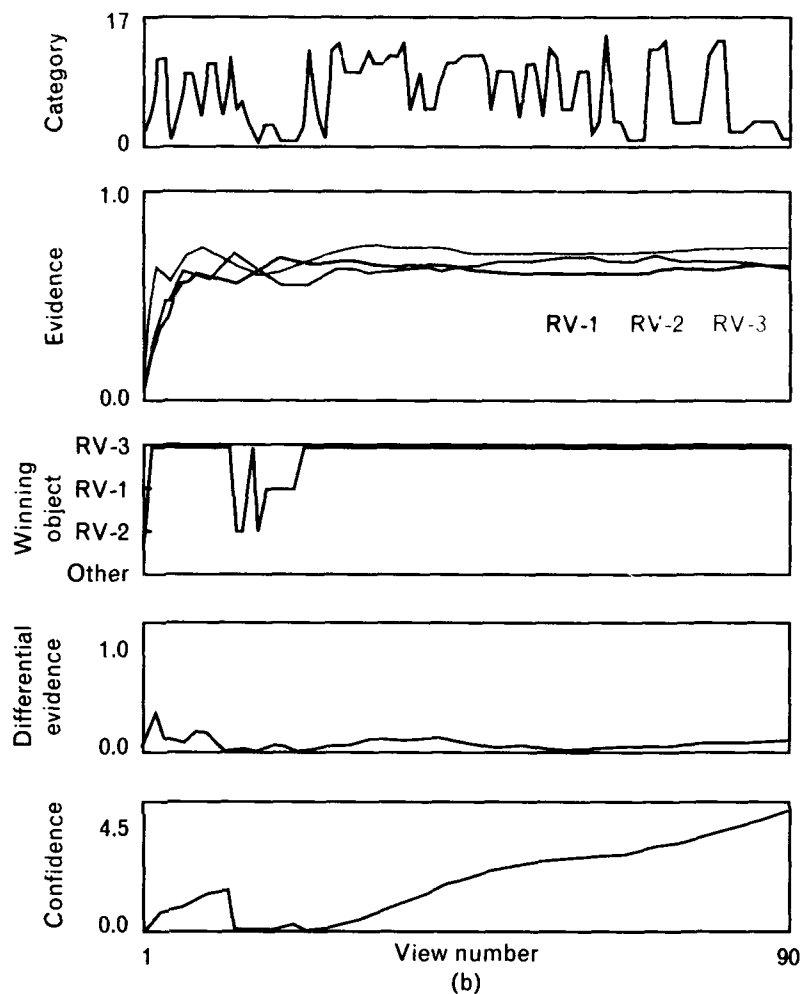
dence for such a finely tuned attentional mechanism has been found in psychological studies in which subjects are demonstrably unaware of stimuli external to the attended visual area.) A serial examination of the object takes place in which the examination is focused on the different *parts* of the stimuli, which may or may not correspond to different parts of the object. But what actually constitutes an *object part*?

For a specific recognition task, some parts may carry more information than other parts, and determining those key parts and the amount of information they carry depends on the specific task. For example, human faces typically have two eyes, so that particular piece of information is not very useful in discriminating between different people, although it would be useful in differentiating human faces from clock faces. In a tactical military application, we need



(a)

**FIGURE 30.** Example of recognition by evidence accumulation: (a) aspect sphere and (b) recognition results (cf. Figure 24). The category sequence recognized by the ART-2 network for reentry vehicle RV-3 of Figure 29(c) is represented both on an aspect sphere seen from above in part a, and as a plot of category versus view number, as shown in the first graph of part b. The next graph shows the evidence generated for the three different reentry vehicles, the following graph shows the selected target with the maximum evidence, and the last two graphs show the differential evidence between the two highest scoring targets and this differential evidence integrated along the view sequence to give a measure of confidence. In this case, the system initially identifies the target correctly and confidence grows, although the differential evidence remains small. But the system then changes its decision, causing the confidence to be reset to zero. Finally, the system reverts to the correct identification and locks in on that decision, and confidence grows monotonically.



(b)

**Table 1. Reentry-Vehicle Recognition Results**

Test Vehicle	Number of Images in Test Sequence	Correct Recognition	Number of Images Required for Convergence	Fraction of Rotation Cycle Required for Convergence
RV-1	90	Yes	4	0.15
RV-2	90	Yes	9	0.33
RV-3	90	Yes	23	1.58

to determine what information is useful in recognizing the differences between the various types of vehicles that are being sought.

In our research, we use *differentencing* to generate expectation-driven part segmentation cues. As with the 3-D object learning and recognition system described earlier, in Figure 31 the best-match aspect category for tank 1 can be located on the tank-1 aspect sphere. The category carries with it a learned template of the invariant appearance of the object. For the extension to part-based representations, the category must also carry a more complete description to include characteristic attributes of the object such as scale, orientation, context, and other information. (Recall the *what* and *where* visual pathways, and their interaction, mentioned earlier.) In addition to a description of specific views of specific objects, the system also requires a description of the views of generic (i.e., average) objects of a class. The generic-object description is necessary to represent efficiently the hierarchical descriptions that have been learned, as well as to navigate quickly through the representation during the recognition phase, as described below. A generic-object description subsumes the descriptions of all the specific objects that are associated with it. For instance, a generic cannon-tank side view is the average of the side views of all tanks that have cannons. Thus the generic view is a generalized composite representation.

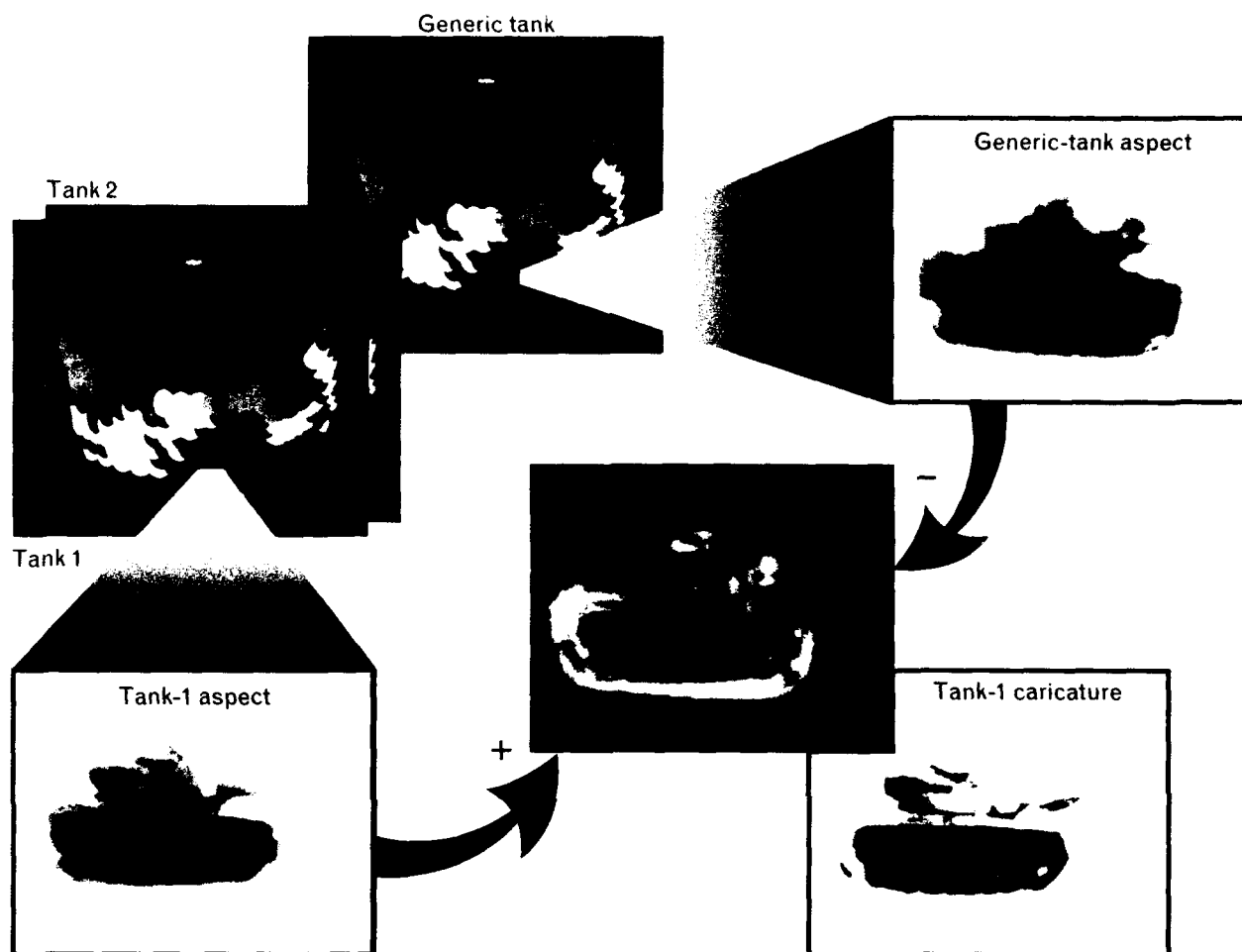
After an ART-2 category is activated, the next step in the object-part process is to compare the description associated with the activated category node of tank 1 with the corresponding previously learned description for the generic tank. The differences between the two descriptions are reported in the form of a visual map called the *Saliency Map*. If all tanks have exactly the same treads, turret, and cannon, then these parts are not salient to the recognition or discrimination tasks, and they will not appear in the Saliency Map. On the other hand, if the gun is longer for tank 1 than for other tanks, then this difference will be evident in the Saliency Map, and the degree to which it is highlighted is used to prioritize the serial attentional examination strategy.

Of course, an input image can activate (to various degrees) the category nodes in many different tank

ART-2 networks. Each of these category nodes has a corresponding view-description template with other associated information, including its own Saliency Map. Each tank's Saliency Map indicates which parts are most salient to discriminating that particular tank, and the saliencies predict which parts should be in the image from that vantage point, if the object in question is indeed that particular tank. The predictions become expectation-driven attentional cues for segmenting the most salient parts of the image, as shown in Figure 32. With Saliency Maps, we not only know *what* parts to look for, but we also know *where* to look for those parts relative to other parts and to the object as a whole. As each expectation is investigated, it either confirms or contradicts the hypothesized description, and evidence is accumulated or dissipated for each potential model target.

A Saliency Map can be obtained for a particular object by computing the difference between the description of the characteristic view of that object and the corresponding description of the characteristic view of the class of objects to which that particular object belongs, i.e., the generic object. Figure 31 illustrates this process. (Note: For simplicity, the Saliency Map shown in Figure 31 was derived from the original gray-scale imagery. In a complete implementation, however, the Map should be obtained from an invariant description of a view, such as a log-polar mapped image with the illuminant discounted in the case of passive visible sensors.) As an example, we might have a generic class of objects that are tanks with cannons, turrets, and treads, and included within that class we might have M48 and M60 tanks. Then the Saliency Map for the front view of an M60 tank represents the differences between the front view of the M60 and the front view of a similar class of tanks in general. Such differences are referred to as "activity"—the greater the difference in a particular area of the Saliency Map, then the greater the activity in that part. Areas in which there are no differences (i.e., no activity) are ignored in the scheduling of attentional shifts.

Using Saliency Maps, we can organize a hierarchical representation of the learned objects. Figure 33 illustrates an example hierarchy of tanks. Beginning at the upper left are the descriptions of a generic tank

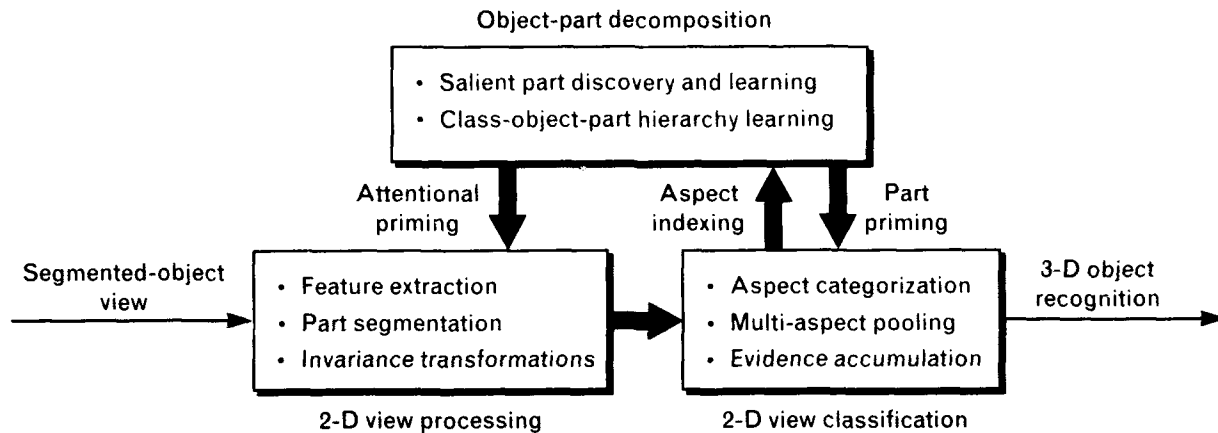


**FIGURE 31.** Construction of a Saliency Map and corresponding caricature image for the side view of tank 1, an M60 tank. The Saliency Map is created by taking differences between the side view of tank 1 and the side view of a similar class of tanks in general. (This class of tanks is collectively called a *generic tank*). The caricature image emphasizes the salient parts of tank 1 with respect to the generic tank. The salient parts in a Saliency Map are used to generate attentional cues for the recognition and discrimination of a particular object among similar objects, and the use of caricatures increases the efficiency of this process.

from various aspects. If all we desire is to discriminate between tanks and aircraft, then this level of description may be adequate. If, instead, we desire to discriminate between a flamethrower tank and a cannon tank, then more detailed descriptions indicating the information-carrying attributes of both types of tanks are needed. The Saliency Maps described earlier naturally contain this information, so that if an object has been determined to be a tank, the Saliency Maps indicate exactly what must be investigated to make a more refined decision about which specific tank that object is. Once the tank has been recognized as a cannon tank, either an M48 or an M60 in this ex-

ample, additional Saliency Maps indicate the differences between these two types of tanks and the generic cannon tank. Although Figure 33 shows only 2-way branching, the branching often is *N*-way.

Caricatures of the object descriptions can be used to increase the efficiency of the recognition process. For the recognition of human faces, there are many different possible facial caricatures that can be used, depending on what qualities are emphasized. A caricaturist might emphasize age, sex, beauty, or simply the differences evidenced between a particular face and a corresponding generic age-matched, sex-matched face. P.J. Benson and D.I. Perrett [8] have demon-



**FIGURE 32.** Conceptual approach to the learning and recognition of class-object-part hierarchies. Again, views are quantized into aspects through the use of unsupervised learning, but objects of a class are averaged together to form generic-object representations. Differences between specific objects and the generic object of that class are highlighted on a *Saliency Map* (Figure 31), which is then used to focus attention on salient parts during the recognition process. Recognized aspect categories for salient parts generate evidence for targets. In addition, the categories prime the system with expectations for other parts at certain locations.

strated a reduction in reaction time in the recognition task for subjects who are shown a caricatured face versus a non-caricatured face.

Caricaturing occurs naturally in the class-object-part hierarchical representations of Figure 33. Computing a difference map between a description of an input target image and a previously learned generic description leads to the detection of differences between the two descriptions. With that information, the differences can then be emphasized, resulting in a caricature of the input description. Because certain parts in the caricatured map have been exaggerated, they stand out even more strongly, and, because the non-differences have been suppressed, attention can be focused more quickly on the parts of the input image description that are most unusual and therefore most likely to carry discrimination information. Figure 31 contains an example caricature image of a tank.

### Visual Navigation by MAVIN

The AIR system design described in the section "Aircraft Recognition from Visible Image Sequences" has been implemented at Lincoln Laboratory on a mobile robot called the Mobile Adaptive Visual Navigator (MAVIN). Shown in Figure 34, MAVIN can be pro-

grammed to travel a reconnaissance path, detect and track objects as it moves, and recognize objects it has learned. Arrays of light bulbs, such as the ones shown in Figure 34, have been used for the target objects. Currently, MAVIN is also able to recognize silhouettes of objects that can be segmented easily from the background. Equipped with binocular cameras, MAVIN operates in real time, with feature extraction running on a PIPE video-rate parallel-processing computer, and all other neural network computations running on SUN computers. (Capable of 1-billion 8-bit integer operations per second, PIPE was developed for robotic vision at the National Institute of Standards and Technology [NIST] and manufactured by Aspex Corp. of New York.)

Our past investigations have incorporated the visual learning and recognition system into a neural architecture that is capable of supporting various Pavlovian behavioral-conditioning paradigms based on learned associations and expectations, including excitatory conditioning, inhibitory conditioning, secondary conditioning, and the extinction of conditioned excitors [22, 23]. We have recently extended the MAVIN system to incorporate the learning and recognition of environments that are defined by the layout of visual landmarks observed during explora-

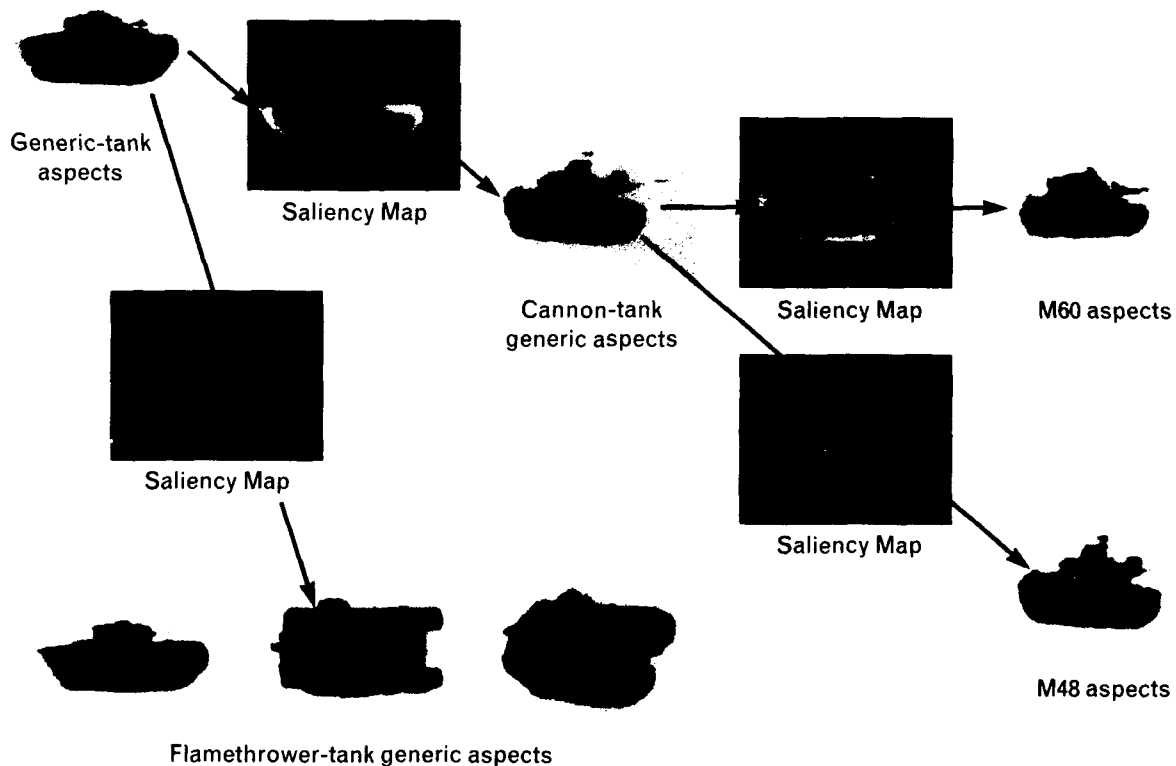
tion [24, 25]. Associative learning methods similar to those used for learning 2-D feature patterns have been applied to spatial patterns of recognized visible landmarks to establish *place cells*, which qualitatively map an environment based on its visual surroundings. We are currently incorporating *displace cells* into the architecture to code *place field* transitions that are induced by robot motions. (A place field corresponds to an area in the environment where recognized target landmarks possess a similar spatial layout.) These concepts for the qualitative mapping and navigation of space are based on behavioral experiments with rats, and on the physiological measurements of neurons in the rat hippocampus.

An important motivation for developing MAVIN

has been to demonstrate in the laboratory the system's ability to recognize in real time both fixed landmarks and mobile targets from a sensor platform that can navigate through, explore, and map an environment, viewing the scene from a variety of vantage points. Indeed, MAVIN has proven to be an excellent experimental domain to test the ATR systems that we have developed.

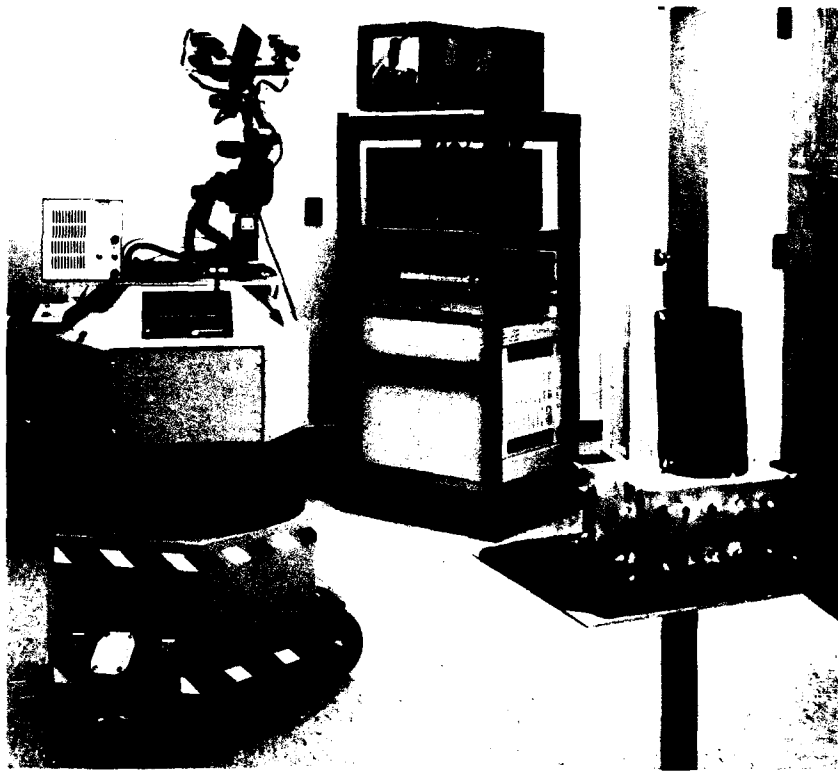
### Conclusion

Our strategy of using the unsupervised learning of view-based, invariant representations in conjunction with evidence accumulation that exploits view transitions has proven effective in several sensory domains, and is relevant to both automatic target recognition



**FIGURE 33.** Hierarchical object representations are a natural consequence of the Saliency Map approach. The Saliency Maps direct a branching down from generic object to specific target, which may be unique because of some specific part. Because of this hierarchy, Saliency Maps can be used in the recognition process to guide a rapid search among learned categories.





**FIGURE 34.** The Mobile Adaptive Visual Navigator (MAVIN) developed at Lincoln Laboratory. MAVIN, a mobile robot with binocular cameras, provides a testbed for a passive-vision ATR system in which the concepts that underlie 3-D object learning and recognition have been extended to the learning of representations for environments that are defined by distributions of visual landmarks. This extension supports the ability for an autonomous sensor platform to explore, map (in a qualitative fashion), and navigate through environments consisting of fixed landmarks and moving targets. The neural architecture being developed is based on studies of the rat hippocampus.

(ATR) and environment navigation. But perhaps the most important lesson we have learned is that many valuable insights can be gained from serious study of the brain and behavior. Anatomical, physiological, and psychophysical studies have all helped shape the computational theories and system architectures used in our work. We believe that such studies will continue to enable rapid progress in the ATR field.

#### **Acknowledgments**

We wish to thank the members of the Surveillance Systems Group at Lincoln Laboratory for providing us with the ADTS imagery of clutter and tactical targets, as well as the radar phase history data for the

ISAR targets. We are grateful to Jacques Verly and Carol Lazott of the Machine Intelligence Technology Group for constructing the ISAR target imagery from the radar phase histories. We are also indebted to the Signature Studies and Analysis Group for providing us with the range-Doppler peak data. This work on reentry-vehicle recognition was done in conjunction with Bob Gabel of the Machine Intelligence Technology Group. We also wish to acknowledge our ongoing collaboration with Professors Stephen Grossberg and Ennio Mingolla of Boston University's Department of Cognitive and Neural Systems.

This work has been supported by the U.S. Department of the Air Force and the Office of Naval Research.

## REFERENCES

1. E.A. DeYoe and D.C. Van Essen, "Concurrent Processing Streams in Monkey Visual Cortex," *Trends in Neuroscience TINS-11*, 219 (1988).
2. S. Zeki, "The Visual Image in Mind and Brain," *Scientific American* **267**, 68 (Sept. 1992).
3. D.A. Fay and A.M. Waxman, "Neurodynamics of Real-Time Image Velocity Extraction," chap. 9 in *Neural Networks for Vision and Image Processing*, eds. G.A. Carpenter and S. Grossberg (MIT Press, Cambridge, MA, 1992), pp. 221-246.
4. M. Mishkin, L.G. Ungerleider, and K.A. Macko, "Object Vision and Spatial Vision: Two Cortical Pathways," *Trends in Neuroscience TINS-6*, 414 (1983).
5. D.I. Perrett, A.J. Mistlin, and A.J. Chitty, "Visual Neurones Responsive to Faces," *Trends in Neuroscience TINS-10*, 358 (1987).
6. D.I. Perrett, M.H. Harries, R. Bevan, S. Thomas, P.J. Benson, A.J. Mistlin, A.J. Chitty, J.K. Hietanen, and J.E. Ortega, "Frameworks of Analysis for the Neural Representation of Animate Objects and Actions," *Journal of Experimental Biology* **146**, 87 (1989).
7. D.I. Perrett, M.W. Oram, M.H. Harries, R. Bevan, J.K. Hietanen, P.J. Benson, and S. Thomas, "Viewer-Centred and Object-Centred Coding of Heads in the Macaque Temporal Cortex," *Experimental Brain Research* **86**, 159 (1991).
8. P.J. Benson and D.I. Perrett, "Perception and Recognition of Photographic Quality Facial Caricatures: Implications for the Recognition of Natural Images," *European Journal of Cognitive Psychology* **3**, 105 (1991).
9. M. Seibert and A.M. Waxman, "Spreading Activation Layers, Visual Saccades, and Invariant Representations for Neural Pattern Recognition Systems," *Neural Networks* **2**, 9 (1989).
10. M. Seibert and A.M. Waxman, "Learning and Recognizing 3D Objects from Multiple Views in a Neural System," chap. II.12 in *Neural Networks for Perception*, vol. 1, ed. H. Wechsler (Academic Press, New York, 1991), pp. 426-444.
11. M. Seibert and A.M. Waxman, "Adaptive 3-D Object Recognition from Multiple Views," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 107 (1992).
12. J.J. Koenderink and A.J. van Doorn, "The Internal Representation of Solid Shape with Respect to Vision," *Biological Cybernetics* **32**, 211 (1979).
13. S. Grossberg, "Nonlinear Neural Networks: Principles, Mechanisms, and Architectures," *Neural Networks* **1**, 17 (1988).
14. R.K. Cunningham and A.M. Waxman, "Astroglial-Neural Networks, Diffusion-Enhancement Bilayers, and Spatio-Temporal Grouping Dynamics," *SPIE* **1611**, 411 (1991).
15. R.K. Cunningham and A.M. Waxman, "Parametric Study of Diffusion-Enhancement Networks for Spatiotemporal Grouping in Real-Time Artificial Vision," *Technical Report No. ESC-TR-92-207*, MIT Lincoln Laboratory (6 Apr. 1993).
16. E.L. Schwartz, "Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding," *Vision Research* **20**, 645 (1980).
17. G.A. Carpenter and S. Grossberg, *Pattern Recognition by Self-Organizing Neural Networks* (MIT Press, Cambridge, MA, 1991), chaps. 9-15.
18. B. Widrow and S. Stearns, *Adaptive Signal Processing* (Prentice-Hall, Englewood Cliffs, NJ, 1985).
19. S. Grossberg, *Neural Networks and Natural Intelligence* (MIT Press, Cambridge, MA, 1988), chaps. 1-4.
20. A.M. Aull, R.A. Gabel, and T.J. Goblack, "Real-Time Radar Image Understanding: A Machine-Intelligence Approach," *Lincoln Lab. J.* **5**, 195 (1992).
21. M. Seibert and A.M. Waxman, "An Approach to Face Recognition Using Saliency Maps and Caricatures," *Proc. World Congress on Neural Networks*, Portland, OR (to be published in July 1993).
22. A.A. Baloch and A.M. Waxman, "Visual Learning, Adaptive Expectations, and Behavioral Conditioning of the Mobile Robot MAVIN," *Neural Networks* **4**, 271 (1991).
23. A.A. Baloch and A.M. Waxman, "Behavioral Conditioning of the Mobile Robot MAVIN," chap. 6 in *Neural Networks: Concepts, Applications, and Implementations*, vol. IV, eds. P. Antognetti and V. Milutinovic (Prentice-Hall, Englewood Cliffs, NJ, 1991), pp. 162-200.
24. I.A. Bachelder and A.M. Waxman, "Neural Networks for Mobile Robot Visual Exploration," *SPIE* **1831**, 107 (1993).
25. I.A. Bachelder, A.M. Waxman, and M. Seibert, "A Neural System for Mobile Robot Visual Place Learning and Recognition," *Proc. World Congress on Neural Networks*, Portland, OR (to be published in July 1993).



**ALLEN M. WAXMAN** is a senior staff member in the Machine Intelligence Technology Group, where his focus on research has been in vision processing, neural networks, mobile robots, and electronic aids for the visually impaired. Allen also currently holds a joint appointment with the Center for Adaptive Systems at Boston University. Before joining Lincoln Laboratory four years ago, he was with the Department of Electrical, Computer, and Systems Engineering at B.U. He received a B.S. degree in physics from the City College of New York, and a Ph.D. degree in astrophysics from the University of Chicago. In 1992, he was the corecipient (with Michael Seibert) of the Outstanding Research Award from the International Neural Network Society.



**MICHAEL SEIBERT** received a B.S. and an M.S. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, and a Ph.D. degree in computer engineering from Boston University. He has been with Lincoln Laboratory for six years; he is currently a staff member in the Machine Intelligence Technology Group. Michael's focus on research has been in vision and neural networks, and in 1992 he was the corecipient (with Allen M. Waxman) of the Outstanding Research Award from the International Neural Network Society.



**ANN MARIE BERNARDON** is a staff member in the Machine Intelligence Technology Group, where her research has been on machine intelligence, neural networks, and signal processing. Before joining Lincoln Laboratory seven years ago, she worked for Voice Processing Inc. She received the following degrees in electrical engineering: a B.S. from Purdue University and an S.M. from MIT.



**DAVID A. FAY** received a B.S. degree in computer engineering and an M.A. degree in cognitive and neural systems from Boston University. While pursuing his graduate studies at B.U., Dave joined Lincoln Laboratory in 1989, and is currently a staff member in the Machine Intelligence Technology Group. His research has been on the development of neural network systems for enhancing radar imagery.

# Multidimensional Automatic Target Recognition System Evaluation

Paul J. Kolodzy

■ We are developing an evaluation facility that includes an electronic terrain board (ETB) to provide an effective test environment for automatic target recognition (ATR) systems. The input to the ETB, which is a high-performance computer graphics workstation, is very high-resolution data (15 cm in 3-D) taken with pixel registration in the modalities of interest (laser radar, passive IR, and visible). The ETB contains sensor and target models so that measured imagery can be modified for sensitivity analyses. In addition, the evaluation facility contains a reconfigurable suite of ATR algorithms that can be interfaced to real and synthetic data for developing and testing ATR modules.

A first-generation hybrid-architecture (statistical, model based, and neural network) ATR system is currently operating on multidimensional (laser radar range, intensity and passive IR) sensor, synthetic, and hybrid databases to provide performance and validation results. A recent study determined the sensor requirements necessary for target classification and identification of eight vehicles under various view aspects, resolutions, and signal strengths.

This article presents a description of the infrared airborne radar used to gather sensor data, a discussion of sensor fusion and the hybrid ATR measurement system, and a review of the ATR evaluation facility. This article also discusses the computer manipulation and generation of laser-radar and passive-IR sensor imagery and the processing modules used for target detection and recognition. We give results of processing real and synthetic imagery with the ATR system, with an emphasis on interpreting results with respect to sensor design.

**T**HE BATTLEFIELD SCENARIO continues to grow in complexity as the use of high-resolution sensors and precision strike weapons has forced the increased use of concealment and camouflage technology to improve vehicle survivability. The advent of multidimensional sensors that trade individual sensor performance for aggregate system performance and automatic target recognition (ATR) systems that can assist in or automatically identify targets also are a threat to vehicle survivability. The understanding of multidimensional sensors, the algorithms that are used

to process their data, and the manner in which they are evaluated is necessary to determine their suitability for military applications.

Unfortunately, the testing and acceptance of ATR systems for military applications has proven elusive. On one hand, many researchers are concerned that not enough information exists in one sensor modality to build an ATR system that performs effectively against targets in natural and man-made clutter. On the other hand, the use of multisensor information to solve this vexing problem is relatively recent, and the

results are limited. Although we have strong indications that several sensor modalities are better than one for target identification, no convincing database of evidence exists.

At Lincoln Laboratory we have constructed a flyable multisensor measurement system to evaluate the use of single and multiple sensor modalities for search-and-identification applications. This article describes the measurement system, which includes a forward-looking suite of sensors, a down-looking suite of sensors, and an MMW sensor. We also describe an ATR system for processing laser radar range and intensity imagery as well as other sensor modalities.

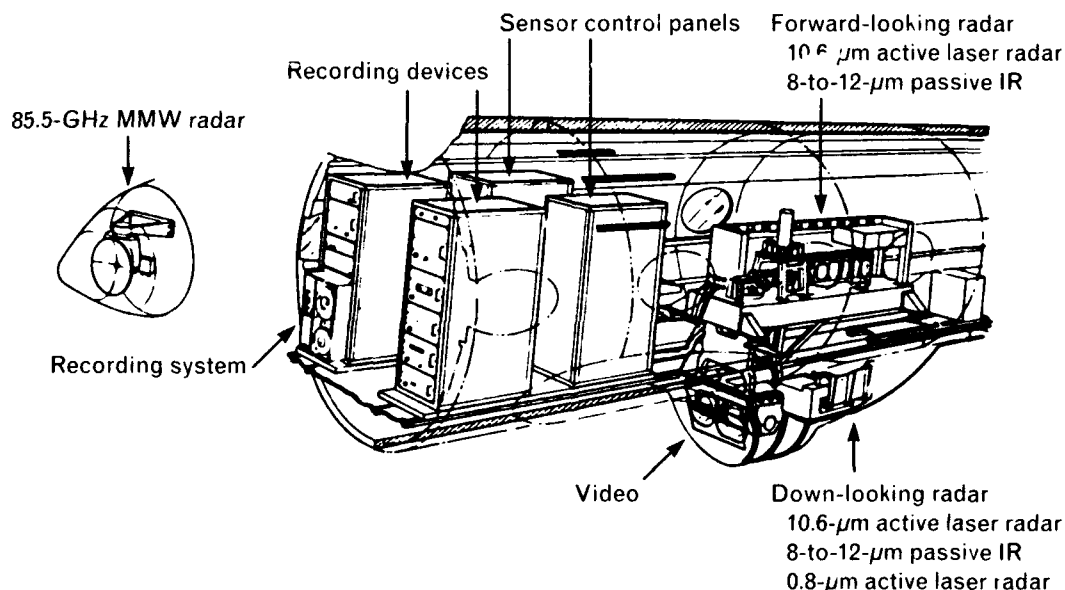
Testing the ATR system to quantify the performance limits of the multisensor measurement system is an important step in the development of useful benchmarks and the definition of radar requirements. This article examines the performance tests we have developed and provides a summary of test results for spatial extent, image quality, and 3-D recognition requirements. An ATR evaluation facility is currently under development to provide an effective test envi-

ronment for ATR systems. The inputs to the facility, which is a high-performance computer graphics workstation and data-processing engine, are very high-resolution data (15 cm in 3-D) taken with pixel registration in the modalities of interest (laser range, intensity passive IR, and visible) and stored in databases. An electronic terrain board (ETB) combines the databases with sensor and target models to modify the measured imagery for ATR sensitivity analyses.

### The Infrared Airborne Radar

The Infrared Airborne Radar (IRAR) is a flyable multisensor measurement system that consists of a set of active and passive infrared (IR) and active millimeter-wave (MMW) sensors. This system is installed in a Gulfstream G-1 twin turboprop test aircraft used by Lincoln Laboratory; Figure 1 illustrates the locations of these sensors in the aircraft. We are especially interested in the ability of the multisensor measurement system to detect targets autonomously (i.e., without human interaction with the measurement system).

In the forward-looking sensor suite, the active laser



**FIGURE 1.** Schematic diagram of the multisensor measurement system on the Gulfstream G-1 aircraft, showing the location of each individual sensor system. The two sensor suites—forward-looking and down-looking—are located in the aft section of the aircraft, the recording system and electronic racks are located in the midsection, and the antenna for the MMW radar is located in the nose. The forward-looking sensor suite is mounted on an optical table and then relayed through a pod on the fuselage. The down-looking sensor suite is housed entirely in the pod aft of the forward-looking sensor system.

radar sensor measures absolute range with a precision of 1 m while the passive-IR sensor measures the thermal intensity of the target and scene in the 8-to-12- $\mu\text{m}$  band. The down-looking sensor suite, which is a multispectral active-passive sensor, has the ability to measure relative range with a precision of 15 cm, as well as the ability to measure passive-IR thermal intensity. In addition, an MMW real-aperture measurement system developed by General Dynamics of Pomona, California, is installed in the aircraft. This MMW sensor measures absolute range with a resolution of 0.5 m, and is slaved to cover the same search area as the forward-looking sensor.

All the IRAR sensors reside on board the aircraft platform. The heart of the IRAR system is located in the center section of the aircraft. A radome extends down from the center of the aircraft, allowing the laser beam of the forward-looking sensor to exit through a germanium window on the left side of the radome. An additional window immediately to the right of the germanium window is used by the measurement system's boresighted color television camera, which is used to point the laser beam manually and to record a live sequence of the measured scene.

The radome was modified so that the down-looking sensor could be placed immediately behind the forward-looking laser-radar pointing-mirror assembly and look straight down; the scan direction of the down-looking sensor is therefore always perpendicular to the longitudinal axis of the aircraft. The MMW system is sufficiently small so that the 1-ft diameter radar dish and the gimbal mount are totally enclosed within the nose cone of the aircraft.

#### *Forward-Looking Laser Radar*

The transmitter in the forward-looking sensor is an RF-excited, water-cooled,  $\text{CO}_2$  waveguide laser operating at 10.6  $\mu\text{m}$ . In the pulsed mode, the transmitter laser provides a nominal 25-nsec pulsewidth at approximately 3-W average power at a pulse-repetition frequency of 20 kHz. In CW operation, the laser can provide power in excess of 30 W.

A 5-in diameter, afocal, Ritchey-Chretien telescope functions both as the transmit and receive aperture of the sensor to produce a 200- $\mu\text{rad}$  diameter beam (100- $\mu\text{rad}$  resolution). The sensor uses two linear 12-

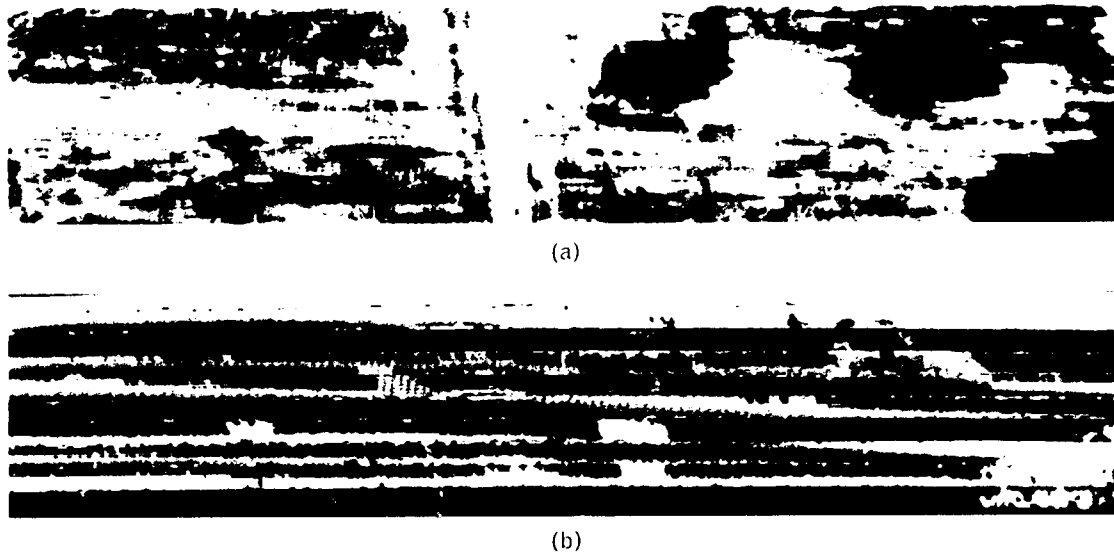
element arrays of HgCdTe photovoltaic detectors: one array for the active measurements and one for the passive measurements. Registration of the active and passive measurements is always assured because both arrays share the common telescope.

In the present configuration, the two arrays are oriented vertically to provide a 10° azimuthal coverage at 2.5 scans/sec in linescan mode. In a separate framing mode (25.6 mrad by 12.0 mrad), the scanning mirrors operate at 20 frames/sec; when the passive channel is enabled, however, the recording rate is reduced to 10 frames/sec because of recorder limitations. Television images from the boresighted TV camera are digitized and stored on computer tapes. Table 1 shows selected system parameters for the forward-looking sensor.

Figure 2 is an example of a laser radar range image and a passive-IR image made simultaneously by the forward-looking sensor. Two features in these images are particularly interesting with respect to data fusion and scene understanding: (1) the road that traverses vertically in the center of the scene is clearly visible in the passive-IR image in Figure 2(b) but invisible in the range image in Figure 2(a) because the road is at the same elevation as the local ground plane, and (2) although a tank (at the center left of the scene) has a negative passive-IR contrast with the background, it has positive range contrast in the active laser radar image. We can overcome the measurement limita-

**Table 1. Forward-Looking Laser Radar System Parameters**

<b><math>\text{CO}_2</math> laser</b>	
Wavelength	10.6 $\mu\text{m}$
Nominal power, CW	30 W
Pulsed, average	3 W
Number of detectors	12
Telescope aperture	13 cm
Instantaneous field of view	0.2 mrad
Range sampling interval	1.1 m



**FIGURE 2.** (a) Passive-IR imagery and (b) laser radar range imagery taken simultaneously at Stockbridge, New York, by the forward-looking sensor. The passive-IR image in part *a* is coded by thermal intensity, so that warmer objects such as vehicles are brighter than cold objects. The range image in part *b* is coded by color to distinguish objects at different distances from the viewer.

tions of each individual sensor by fusing the information from the two sensors to provide enhanced detection capability.

#### *Millimeter-Wave Radar*

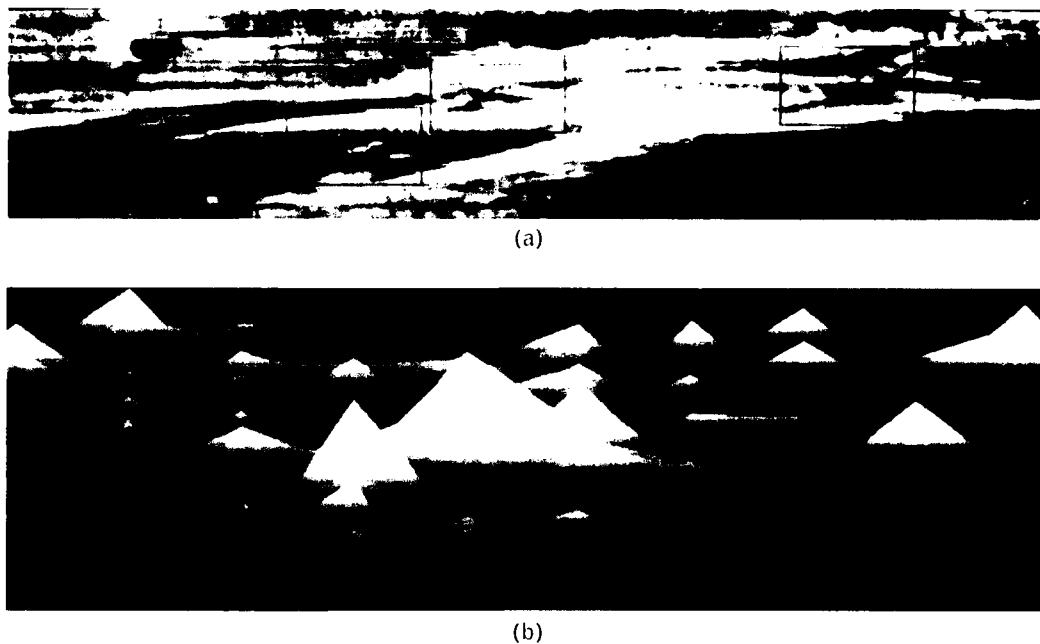
To investigate the advantages of combining the output of two or more diverse sensors, we added the General Dynamics 85.5-GHz real-aperture MMW radar to the forward-looking sensor suite. This radar has low cross-range resolution and high line-of-sight resolution, and operates at 3.5 mm. Table 2 lists the operating characteristics of this radar. The MMW antenna is mounted in the nose section of the aircraft and is boresighted to the IRAR sensor suite during pointing-mode operation.

The cross-range resolution of the MMW radar is such that a 10° azimuthal field of regard is stored as 15 intensity-versus-range profiles on each scan. The oversampling that occurs in the down-range dimension is then used to enhance the processing statistics for detection. The modulation characteristics of the sensor are such that the line-of-sight range resolution is 1.7 ft, while data are sampled at approximately half this value, thus providing the potential for excellent range resolution on the target.

Figure 3 illustrates the range resolution of the MMW radar in combination with a passive-IR imaging sensor. Three logging trucks in the passive-IR image in Figure 3(a) are each highlighted by a box. The environmental conditions at the time the data were taken are responsible for the low passive-IR con-

**Table 2. Millimeter-Wave Radar System Parameters**

Operating frequency	85.5 GHz
Transmitter power	15 mW
Modulation format	FMCW
Antenna diameter	12 in
Antenna beamwidth	0.76°, one way 0.57°, two way
Range resolution	1.67 ft
PRF	1600 Hz
Noise figure	20 dB, including system losses



**FIGURE 3.** (a) Passive-IR imagery and (b) boresighted MMW radar imagery. The MMW radar data are displayed as a 3-D plot of down-range, cross-range, and thermal-intensity values. The three logging trucks indicated by boxes in the passive-IR image correspond to four of the five highest MMW radar intensity peaks. Part b shows two peaks for the one truck in the center of part a because distinct returns were obtained from both the truck cab and the truck bed.

trast. If the MMW radar signal is displayed as a 3-D image (cross-range, down-range, and intensity), however, as shown in Figure 3(b), then four of the five highest intensity peaks shown in the figure correspond to radar returns from target locations. Two peaks are determined in Figure 3(b) for the truck in the center of Figure 3(a) because we obtained strong distinct returns from both the truck cab and the truck bed.

#### *Down-Looking Laser Radar*

The multispectral active-passive down-looking sensor is a compact multiple-channel system that employs two lasers for active detection and a single passive detection channel. This sensor is configured with a 10.6- $\mu\text{m}$  amplitude-modulated continuous wave (AMCW) CO<sub>2</sub> laser and a 0.8- $\mu\text{m}$  AMCW AlGaAs diode laser for the two active channels, which are coregistered with an 8-to-12- $\mu\text{m}$  passive detection channel.

The system was designed with 1-mrad angular resolution to provide a 15-cm cube on the target from an

optimal measurement height of 150 m. The active-channel lasers are modulated at 15 MHz to provide an AMCW waveform that translates to a 10-m range ambiguity but provides 15-cm precision (i.e., the range values are produced from 0 to 10 m in 15-cm increments and they fold over at the range boundaries). Thus these measurements are relative range measurements with 15-cm precision, as compared with the absolute range measurements of the forward-looking sensor. Table 3 lists selected parameters of the multispectral active-passive down-looking sensor, and Figure 4 shows five separate images produced by this sensor during a flyover of the USS *Connole*.

The multispectral down-looking sensor has two characteristics of interest for the development and testing of ATR systems: (1) the viewing aspect allows the imaging of objects in clutter that are not generally seen by forward-looking sensors, and (2) the highly precise range imagery gives us the capability to transform the observed scene to a variety of viewing aspects. Figure 5 illustrates this process. Figure 5(a) contains a photograph of a truck that is camouflaged



**Table 3. Down-Looking Laser Radar System Parameters**

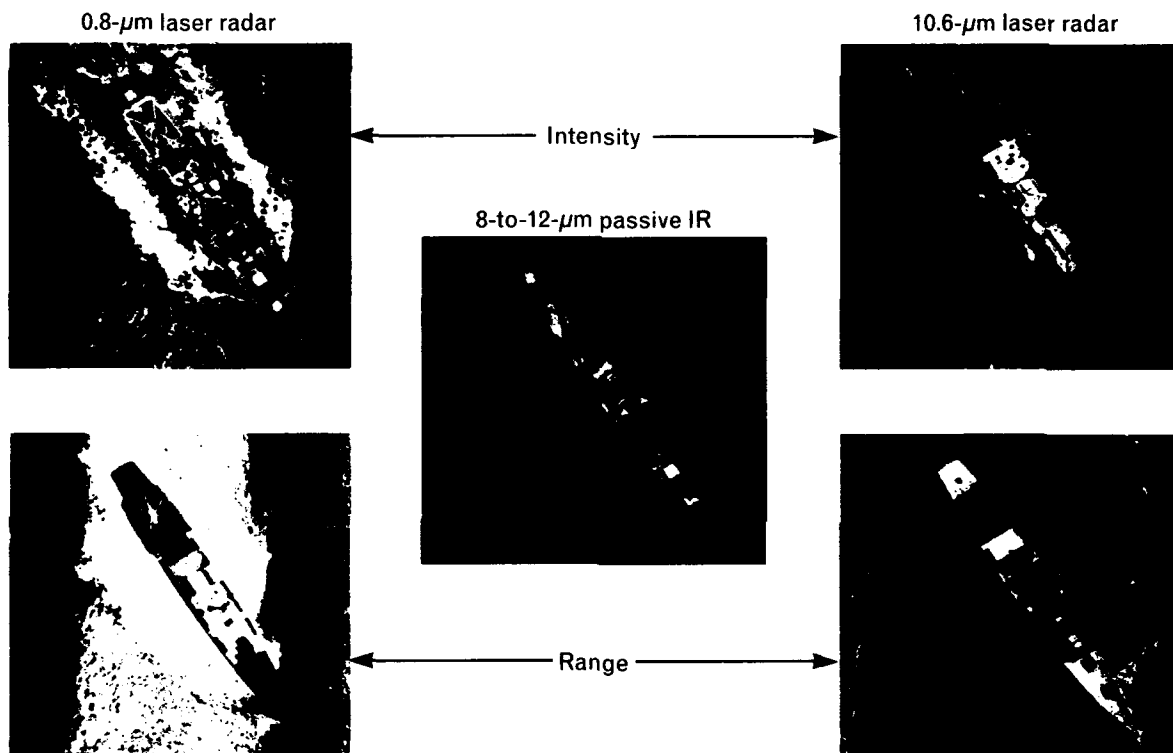
Angular resolution	0.5 mrad, x and y axes
Range precision	15 cm
Range ambiguity interval	10 m
Altitude range	400 ft to 1300 ft
Ground coverage	2000 ft at 1000 ft

by netting and parked on a dirt road in a forest. Figure 5(b), which is the down-looking range image, clearly shows the road and the truck, with the height of the truck above the road encoded in color. Figure 5(c) is a computer-transformed forward-looking range

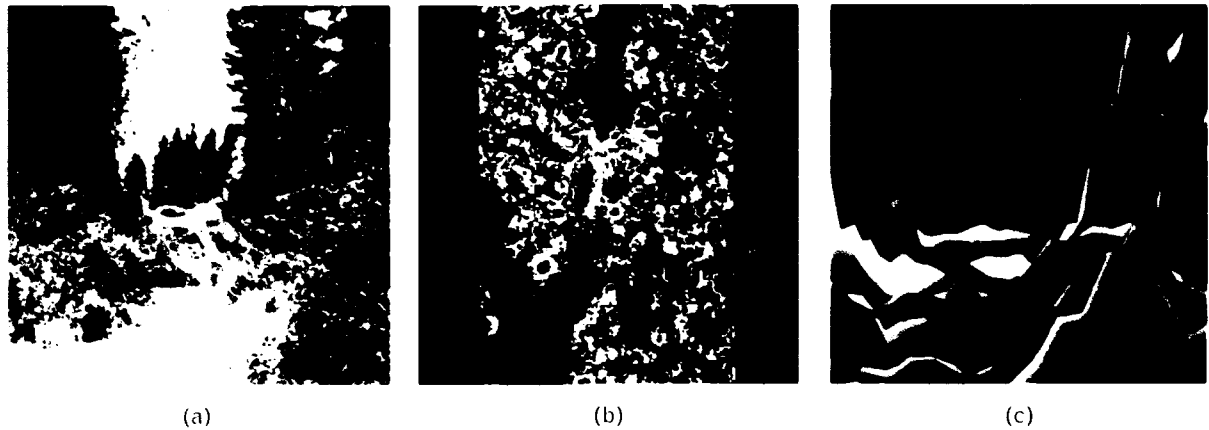
image of the camouflaged truck from a viewpoint that is just above the road. In this way, a down-looking view can be used to develop or test algorithms for a forward-looking or near-forward-looking sensor through the use of coordinate transformations. A more detailed description of how down-looking data can be utilized for a variety of ATR evaluation tasks is given in the section entitled "The ATR Evaluation Facility."

### Sensor Fusion

Figure 3 illustrates the possible benefits of fusing MMW radar imagery and passive-IR imagery. This figure demonstrates that the MMW radar image can be used to indicate areas of interest in a coregistered passive-IR image. Other techniques that incorporate the detection lists from both sensors usually fuse the lists by an OR or AND procedure; i.e., the target



**FIGURE 4.** Example of imagery produced by the multispectral active-passive down-looking sensor during a flyover of the USS *Connole*. This sensor produces coregistered laser radar range and laser intensity images for wavelengths of 0.8  $\mu\text{m}$  and 10.6  $\mu\text{m}$ , as well as an 8-to-12- $\mu\text{m}$  passive-IR thermal-intensity image. Note the parked helicopter near the stern of the ship in each of the sensor domains as well as the depiction of the ship's wake.



**FIGURE 5.** (a) Optical photograph of a truck covered with camouflage netting on a road in a forest. (b) The relative range image of the truck as determined by the multispectral down-looking sensor. (c) The 3-D spatial transformed image illustrates the relative range image data in part *b* as viewed from a depression angle similar to that of the optical photograph in part *a*.

must be detected at least on one list (OR) or on all lists (AND). The OR procedure produces a higher likelihood of detection at the expense of a high false-alarm rate. On the other hand, the AND procedure has a low false-alarm rate at the expense of a lower likelihood of detection. The next section describes an AND procedure that fuses sensor data to create a range-passive histogram, and the following section describes a maximum-likelihood fusion estimate for object detection.

#### Range-Passive-IR Histogram

Target cueing and detection can be accomplished with range data alone, with a range-only histogram, or with a range passive-IR histogram (which is created by using an AND operation to fuse range and passive-IR data registered at the pixel level) [1]. The range-only histogram is a 3-D mapping of the number of occurrences of a range value plotted in a coordinate system of cross-range versus down-range. The histogram is calculated by scanning the range image pixel by pixel and adding one count to the histogram bin that corresponds to the pixel azimuth and the pixel down-range value:

$$H_{RR}(az, rng) = \sum_{cl} U(az, cl),$$

where

$$U(az, cl) = \begin{cases} 1 & \text{if } R(az, cl) = rng \\ 0 & \text{otherwise} \end{cases}$$

and where  $R(az, cl)$  is the absolute range of that pixel, and  $rng$  is the specific range value. Peaks in the 3-D range-only histogram indicate regions of significant vertical extent, or *verticality*, in the image, and the magnitude of the peak represents the vertical surface area of an object in the image. The property of verticality is effective for finding targets in open terrain; it produces a large number of false alarms, however, when applied in wooded areas.

The passive-IR thermal intensity can be used as a discriminant to separate trees from man-made targets that have a significant positive thermal signature. Pixel-level fusion of the range image data and the passive-IR image data is possible because each pixel of the range and passive-IR images is collocated. Each passive-IR pixel can be registered, according to its associated range value, to compute what we define as a *range-passive-IR histogram*.

The range-passive-IR histogram is a 3-D mapping of the sum of the passive-IR intensities plotted in cross-range versus down-range coordinates derived from the pixel-registered range image. Figure 6 shows an example of a range-passive-IR histogram. In Figure 6(a), a passive-IR intensity histogram is calculated for

each column, which corresponds to a particular cross-range value that uses both the range image to provide the coordinates for the histogram and the pixel-registered passive-IR image for the intensity values. An azimuth value is selected, and then we scan the range image pixel by pixel along that azimuth column, where the range value for each pixel selects the histogram range bin. The corresponding passive-IR intensity value in Figure 6(b) is then added to that histogram

bin. In this way, a three-dimensional histogram (cross-range, range, passive-IR intensity) is created, as shown in Figure 6(c). Peaks in the histogram indicate objects with vertical extent (i.e., trees, buildings, and vehicles) and with sufficient thermal contrast with respect to the background (i.e., running engines, heated buildings).

This calculation is written as

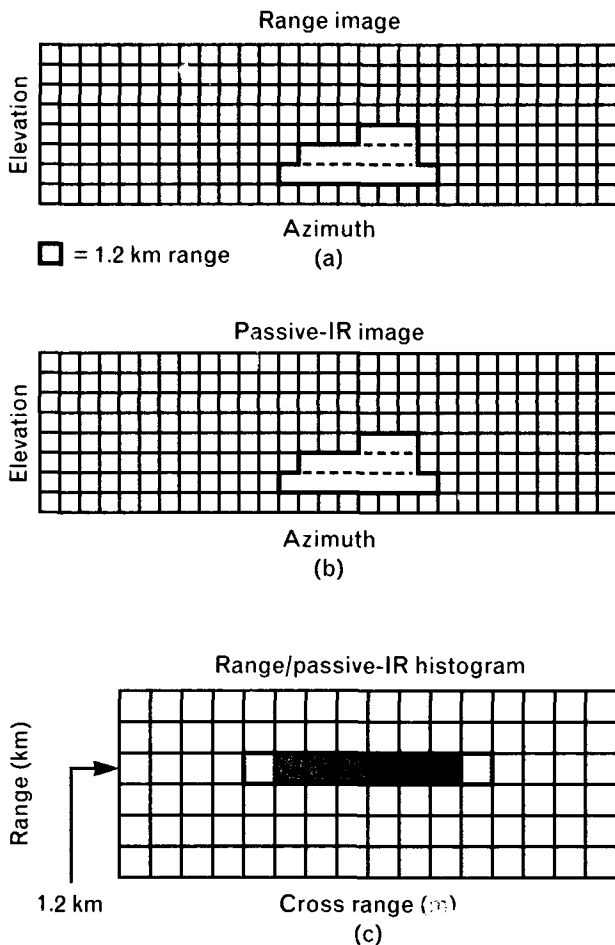
$$H_{RP}(az, rng) = \sum_{el} P(az, el) \times U(az, el),$$

where  $U(az, el)$  is as defined previously and  $P(az, el)$  is its processed passive-IR intensity. Peaks in the range-passive histogram indicate regions of vertical extent that have positive thermal contrast.

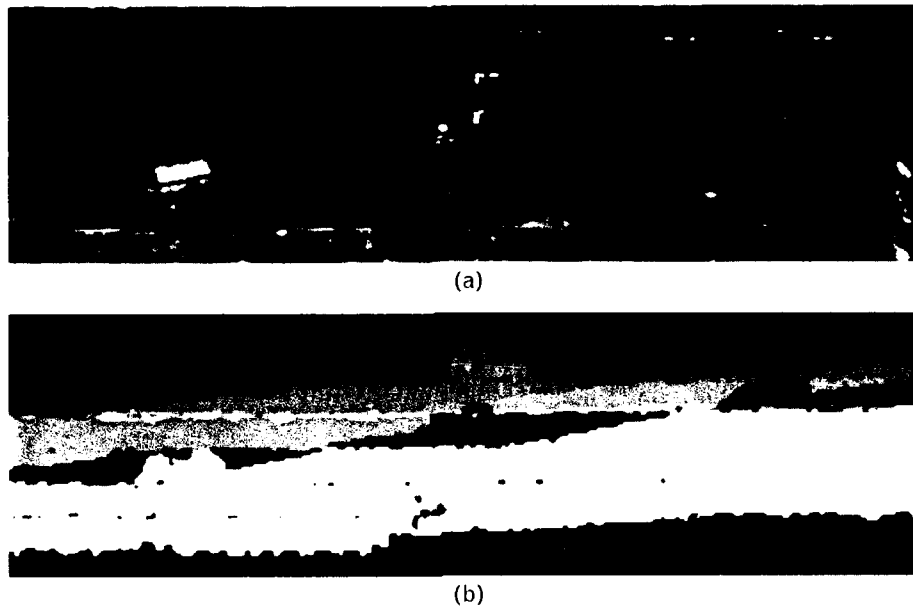
Figure 7 shows how the range-passive histogram algorithm was applied to an IRAR linescan scene taken at Fort Devens, Massachusetts. The linescan scene contains the passive-IR image and laser radar range image of three trucks and a motor generator set. The vehicles were not in operation; their thermal signature is due entirely to solar heating. Figure 8 shows the resulting range-passive histogram. The three largest peaks correspond to the three trucks in the scene. For each peak, the truck position is now localized in cross-range and down-range. This example clearly shows the value of fusing multiple sensor domains at the pixel level with an AND operation, which improves the probability of detection and lowers the probability of false alarms.

#### *Theoretical Study of Active-Passive Detection of Multipixel Targets*

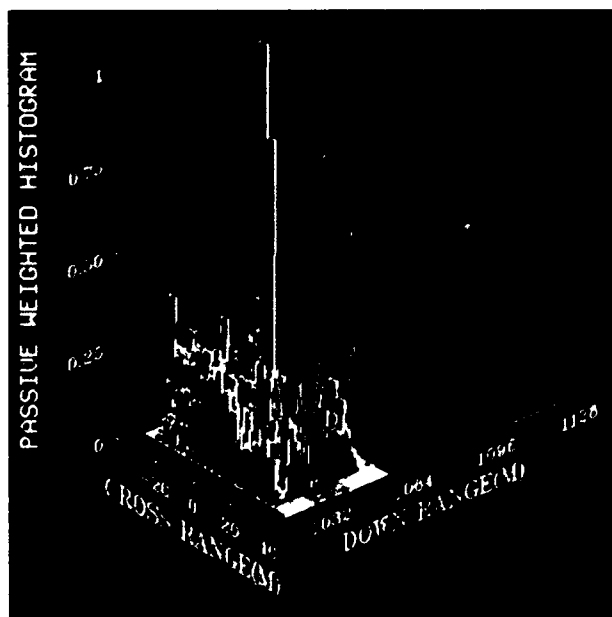
Research into the development of a quasi-optimal, single-sensor detection processor for multipixel laser radar was done by M. Mark [2] and resulted in the generation of receiver operating-characteristic curves for this processor. Mark used a generalized-likelihood ratio test to estimate unknown parameters for a maximum-likelihood estimate. Computer simulations with benign synthetic scenes, generated with uniform laser intensity, range, and passive-IR values for target and background, were used to provide performance measures. Recent extensions of this work to multiple sensor modalities (laser radar range and laser intensity,



**FIGURE 6.** Schematic diagram of how the range image and passive-IR image are mapped into a range-passive-IR histogram. (a) An azimuth value is selected, and the range image is scanned pixel by pixel along that azimuth column; the range value for each pixel selects the histogram range bin. (b) The passive-IR intensity value from the corresponding passive-IR image column is then added to the histogram bin. (c) In this way a three-dimensional range-passive-IR histogram (cross-range, range, passive-IR intensity) is created.



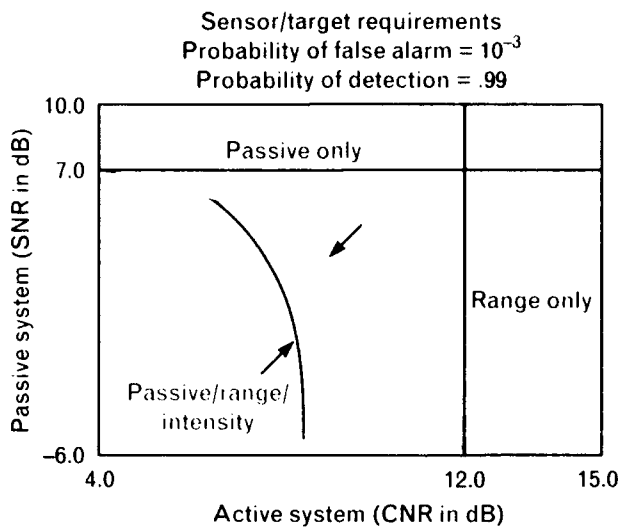
**FIGURE 7.** A scene containing three trucks and a motor generator as imaged by (a) the passive-IR sensor and (b) the laser radar range sensor in the forward-looking sensor suite. The trucks and generator are clearly visible in the center of the passive-IR image. The laser radar range image depicts the objects as silhouettes standing out of the sloping terrain and in the same location as in the passive-IR image.



**FIGURE 8.** The result of processing the data in Figure 7 with the range-passive-IR histogram. The down-range values are color coded in the same manner as the laser radar range image in Figure 7. The four highest peaks correspond to the three trucks in the scene. These peaks would cue a classification processor to a region of interest.

passive-IR thermal intensity) were accomplished by S. Hannon and J. Shapiro [3]. The results of these computer simulations, which were later confirmed by experimental data, indicate that for a specified operating power such as the probability of detection and the probability of false alarm, the required sensor signal-to-noise ratios were relaxed for a multisensor measurement system over a single sensor system.

Figure 9 depicts the sensor/target requirements for a 10-pixel target (2 pixels by 5 pixels) on a 1000-pixel image (20 pixels by 50 pixels). The target size can be scaled to simulate a tank-sized vehicle at a distance of approximately 5 km with a sensor field of view (given a 6° depression angle) of 15,000 m<sup>2</sup>. Figure 9 indicates the sensor requirements for detecting 99% of tank-sized vehicles at 5 km with a false-alarm rate of 10<sup>-3</sup>, or of 0.1 km<sup>2</sup>. Because the simulations were done on idealized scenes, however, the results are not directly transferable to a specific sensor design. The trends still indicate a reduction for either passive-IR signal-to-noise ratios (SNR) or laser radar carrier-to-noise ratios (CNR) when a combination of two sensors is employed.



**FIGURE 9.** Sensor/target requirements for multipixel target detection using the generalized likelihood ratio test for single and multiple sensor modalities to detect a tank-size target at 5 km with a probability of detection of 0.99 and a false-alarm rate of  $10^{-3}$  per image or  $0.1 \text{ km}^{-2}$  [3]. A 7-dB SNR is required for a passive-only sensor system, and a 12-dB CNR is required for a laser radar range-only sensor system. The SNR and CNR requirements are relaxed for a combined passive-range sensor system or a passive-range-intensity sensor system.

### Hybrid ATR System

We have developed ATR processing modules for the primary sensor groups described previously; these groups are laser radar intensity, range, passive-IR thermal intensity, and MMW. Although the individual processing modules can vary among sensor groups, the general processing structure has the same sequence of stages: cleanup, detection, segmentation, feature extraction, invariant mapping, and classification. The general ATR system was originally developed to operate on laser radar range and intensity imagery, and the results presented in this article are based on this imagery. Figure 10 illustrates the processing modules for the range-imagery recognition system; this system is described in more detail below.

### Modular ATR System Concept

The unambiguous range image is first processed by the cleanup stage to reduce data anomalies and enhance the image. The cleanup stage attempts to re-

construct the most probable input image that would produce the measured sensor image. This reconstruction clarifies the image appearance, and makes the returns from the various objects in the scene appear more continuous and complete by reducing sensor and scene artifacts such as dropouts and anomalies.

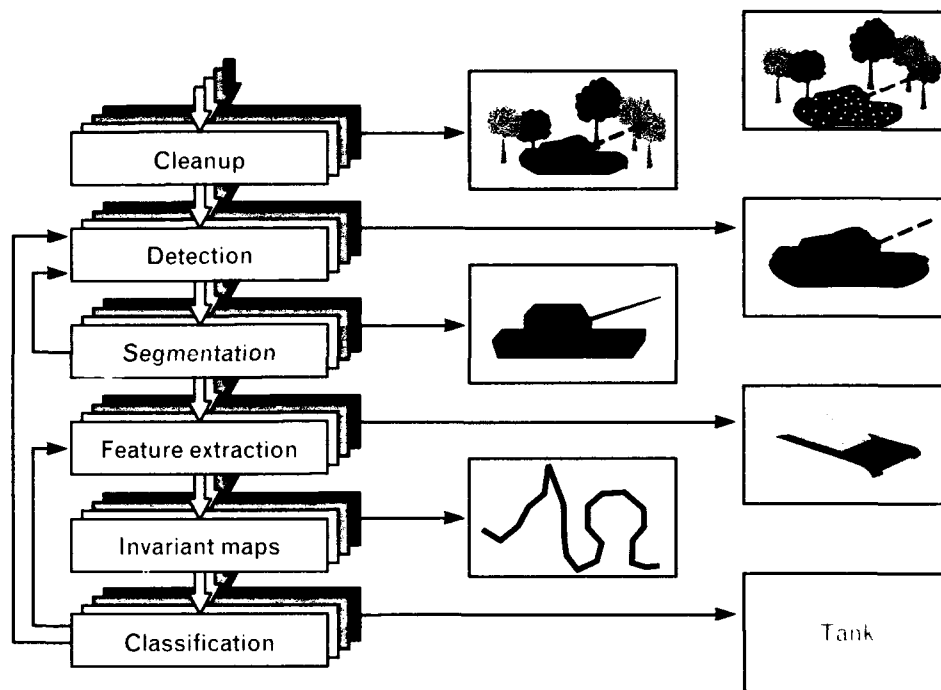
Next, the enhanced image is processed by the detection stage to identify correctly sized regions of constant range as potential targets. The detection stage extracts these regions from the background clutter and removes the ground plane. The detected target at the output of this stage is a silhouette consisting of multiple fragments and rough boundaries.

The multiple fragments are combined by the segmentation stage into a complete, smooth, filled silhouette. The completed silhouette is then separated by the feature extraction stage into feature regions (e.g., barrel, turret, body, and tread for a tank). For this article, the entire target silhouette is considered the single feature. The silhouette is then mapped by the invariant-mapping stage into an abstract pattern that is invariant to translation, rotation, and scale within the sensor field of view. This invariant pattern is processed by the classification stage, which initially learns to cluster the invariant maps into groups and then, after the training cycle, classifies the input data with respect to its learned categories.

### Image Cleanup

To provide adequate recognition performance in a noisy environment, the cleanup stage must be capable of using prior knowledge to restore measured images. We present here an image-restoration model that quantitatively incorporates prior knowledge of the measurement process and scene. The model is based on a Bayesian formulation using Markov random fields, as introduced by S. Geman and D. Geman [4]. The processing is massively parallel because the Markov-random-field assumption allows the image to be decoupled into a large number of connected local neighborhoods, each of which can be processed independently. The local-neighbor information is spread out in time such that a global image restoration is effected when the image-restoration system reaches a steady state.

Real-time image restoration is possible by using



**FIGURE 10.** The six processing modules of the range imagery-recognition system: cleanup of sensor artifacts, detection of potential targets, segmenting targets to improve image characteristics, extraction of relevant features, invariant mapping of features to remove translation and rotation effects, and the classification of features into target categories.

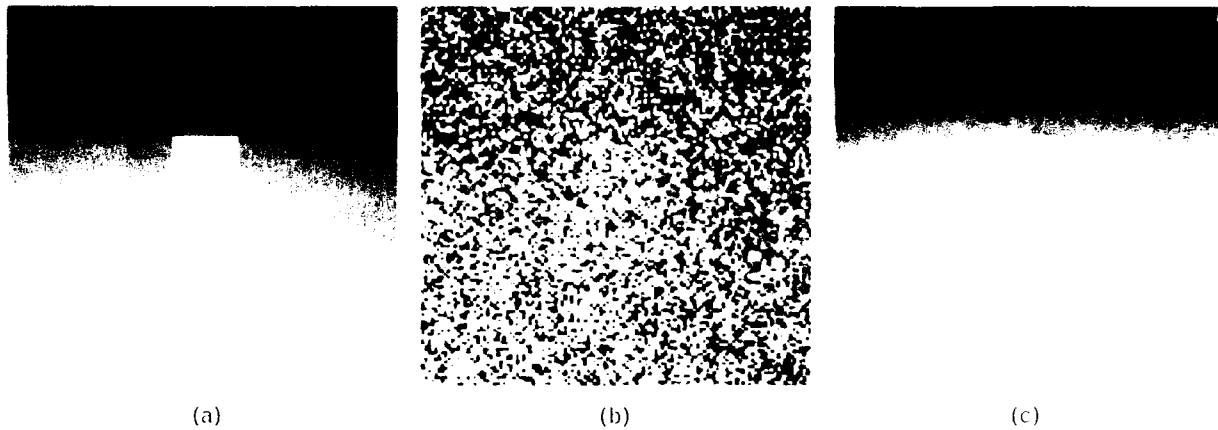
the model with a massively parallel single-instruction multiple-data (SIMD) computer such as the Connection Machine or a direct hardware implementation on a custom microprocessor. A more detailed description of the image-cleanup process is given in this issue in the article by Murali M. Menon entitled "An Efficient MRF Image-Restoration Technique Using Deterministic Scale-Based Optimization."

We applied the image-cleanup process to a simple synthetic image corrupted with noise according to a measurement model described in the literature [3]. The noise does not have a Gaussian distribution and is based on realistic sensor measurements. The original noise-free synthetic image has a simple geometric shape at a constant pixel value, with a background that linearly increases in pixel value from the top of the image to the bottom. Figure 11(a) shows the uncorrupted image, while Figure 11(b) shows the image with 70% of the pixels corrupted with noise. The original image has 256 gray levels, and the noise

spans the entire range of possible pixel values. Except for a few discrepancies at the boundary, the restoration shown in Figure 11(c) is nearly perfect, especially the recovery of the sloping background.

#### *Target Detection*

The detection stage of the ATR processing system extracts target-like regions from the enhanced range image produced in the cleanup stage. The process occurs in three phases: (1) regions of interest are selected, (2) target-like objects are detected, and (3) objects are extracted from the scene. Regions of interest are located by using range-only or range-passive-IR histograms, as previously described in the article. The peaks of these histograms indicate regions of significant vertical extent (i.e., constant range with varying elevation), or a significant thermal signature with some vertical extent. The selected regions are searched for areas of constant range that have range contrast with their neighbors and are similar in size



**FIGURE 11.** The effect of processing a synthetic laser radar range image of a geometrical object with the image-cleanup neural network. (a) The original noise-free image, (b) the image with 70% of the pixels corrupted with noise, and (c) the restoration of the original image from the corrupted image with only a few discrepancies at the image boundary.

(both in absolute height and width) to a target of interest. The target-like region is then separated from the background by selecting only the pixels with that range value. The object is then extracted from this selected image by computing and removing the ground plane.

Figure 12(a) shows the initial range image of an M48 tank at 700 m and the subsequent detection result that was formed by using the previously described range-only histogram and removing the ground plane. Figure 12(b) shows the M48 tank after the cleanup stage and detection stage of processing.

#### *Segmentation*

The segmentation stage of the MTR system smooths the boundaries and completes the fragments of the detected potential target. The boundary-contour system (BCS), a subsystem of a visual processing theory developed by S. Grossberg and E. Mingolla [5], is used to generate the perceived segmentation of the potential target, with respect to illuminance contrasts. The BCS system consists of two stages: an oriented-contrast (OC) filter and a cooperative-competitive (CC) loop. The OC filter measures local luminance



**FIGURE 12.** (a) The initial range image of an M48 tank; (b) the subsequent detection result that was formed by using the range-only histogram and removing the ground plane.

differences, or edges, within an image at a number of different orientations. This filter models the orientation-selective cells discovered by D. Hubel and T. Wiesel [6] in the human visual system. These oriented edge strengths are then allowed to compete and cooperate with one another in the CC loop to generate the perceived boundary contours.

The four layers of the CC loop consist of two competitive layers, one cooperative layer, and a feedback layer. The first competitive layer thins and sharpens boundaries within the image by allowing competition for dominance in the final boundary segmentation between neighboring edge strengths of the same orientation. The second competitive layer straightens jagged or noisy boundaries by allowing competition between edge-contrast strengths with differing orientations at the same location. The cooperative layer completes and connects boundaries by allowing edges of like orientation to cooperate over a distance in the image. The feedback layer introduces into the system any new boundaries formed by the cooperative layer.

The OC filter is implemented by convolving a set of orientationally tuned digital filters with the input image. The CC loop is modeled by using a set of four coupled nonlinear differential equations for each orientation and location within an image. Input to the CC loop is static; therefore, the boundary is complete when the system of differential equations is in equilibrium.

The BCS algorithm has been previously applied to laser radar imagery as reported by Kolodzy et al. [7] and by E. Van Allen [8, 9]. Figure 13 shows an example of BCS processing on the range image of an M48 tank. The input range silhouette in the upper left of the figure is the input to the segmentation stage. The range-silhouette image is sampled to obtain the oriented contrast strengths by using the OC filter in each of twelve orientations. These oriented contrast strengths are then processed by the CC loop to produce twelve new images, which are compressed into a single image by using one of two methods: (1) compute the maximum contrast strength of any orientation at each pixel location (upper right of Figure 13), or (2) sum the contrast strengths across all the orientations at each pixel location (lower left of Figure

13). The compressed image is then filled to form a completed and smoothed silhouette of the potential target for classification. The specific filled image shown in the lower right of Figure 13 is the result of using the summing method for compressing the results of the CC loop.

### *Feature Extraction*

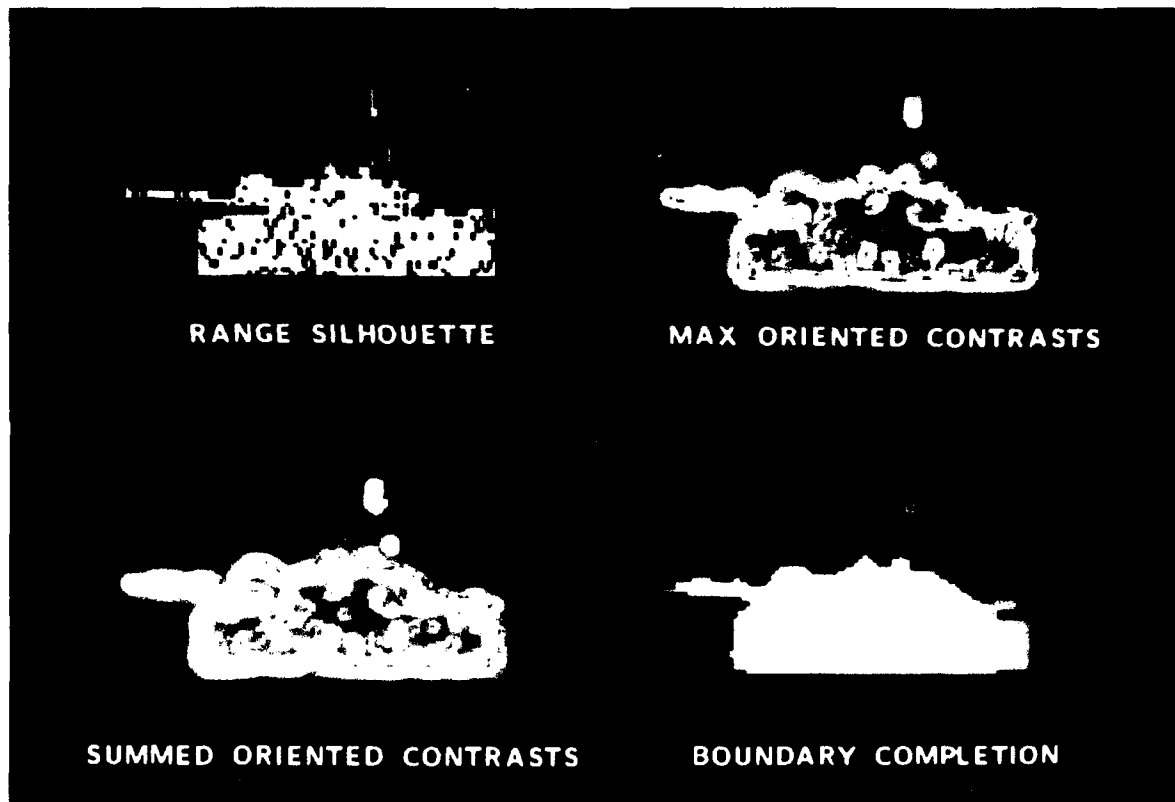
In general, the filled and smoothed image provided by the segmentation stage of the ATR system is then used to extract relevant features for classification. Many different feature domains (images or vectors) such as image geometry, object parts, fractal dimensions, distance of hot spots from the central locations, and the Hough transform can be used and are part of ongoing research.

In particular, model-based systems have been developed at Lincoln Laboratory to parse images into geometric features (such as circles, squares, rectangles) and subsequently classify those features into target features (such as barrel, hull, turret). These model-based systems are discussed in the article by J. Verly et al. entitled "Machine Intelligence Technology for Automatic Target Recognition" [10]. The use of model-based systems for feature extraction in this ATR system has been evaluated previously by D. Dudgeon et al. [11], and they are not discussed further in this article. For our purposes, the pixel image provided by the segmentation stage is used as the feature for classification.

### *Invariance Mapping*

The segmented silhouette is spatially mapped to eliminate translation, rotation, and scale variations prior to classification. An invariant silhouette is therefore built directly into the classifier memory to form a single compact representation for the target. This invariance reduces the number of stored patterns from one pattern for each of several terrain angles and ranges to a single stored pattern, which reduces memory requirements and search times and improves efficiency. Invariance can be obtained by using the following process: (1) locate the segmented silhouette in the field of view, (2) detect the silhouette edges, and (3) spatially map the silhouette edges. The resultant abstract pattern has the desired invariance and is processed di-





**FIGURE 13.** Segmentation of a laser radar image of an M48 tank with the boundary-contour system (BCS) showing smoothed boundaries and connected segments. The image in the upper left is the original segmented range silhouette. The result of applying the BCS using the maximum-contrast edge-strength method is shown in the upper right and the result using the summed-contrast edge-strength method is shown in the lower left. The summed-contrast edge-strength result was then filled and is shown in the lower right.

rectly by the classifier in the next stage.

The target silhouette is located in the plane of the field of view by calculating a position weighted sum, or *centroid*, of its pixel intensities. The centroid of the segmented silhouette is then used as the origin in the spatial mapping that follows.

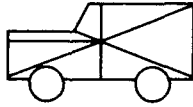
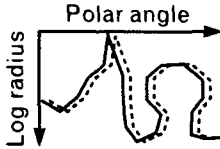
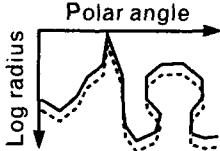
The silhouette is next detected for edge strengths. The edge detection algorithm uses contrast sensitive oriented elliptical receptive fields. In this approach, the receptive fields are passed over the image to sum the pixel energy present in the area centered around each pixel. The major axes of the elliptical receptive fields are oriented in as many as twelve directions to calculate edge strength as a function of orientation. The output at each pixel in the edge image is the output value of the strongest receptive field orientation at that pixel. Because edge detection by receptive field processing is computationally intensive, there is a trade-off between orientation accuracy and process-

ing time. The work described in this article was satisfactorily accomplished by using four orientations for the receptive fields.

The spatial mapping function provides target rotation and scale invariance within the plane of the field of view. The function used in this work is a log polar mapping of the edge strength image about its centroid, as shown in Table 4. The log polar mapping is biologically inspired by the visual field mapping of the human visual cortex, as demonstrated by L. Schwartz [12].

The log polar mapping transforms the edge detected range slice to a log radius, polar angle coordinate system by using the centroid of the silhouette as its reference point. Determining the location of the image centroid is important because the accuracy of the log polar mapping is often highly sensitive to errors in the position of the centroid. Tests using noisy simulated imagery, however, have shown the

**Table 4. Invariant Mapping of Silhouette Edge Strengths for Translation, Rotation, and Scale Invariance in the Sensor Field of View**

Translation	Pixel centroid locates origin for mapping log radius and polar angle	
Rotation	Rotation in field of view is mapped to shift in polar angle	
Scale	Scale (or range) is mapped to shift in log radius	

centroid calculation and log-polar mapping to have robust behavior.

The log-polar mapping around the centroid of the target maps rotation in the field of view to a shift in polar angle, while it maps range to the target as a shift in log radius. This mapping is insensitive to rotation and scale variations; cross-correlation with an unrotated, unscaled log-polar mapping gives an estimate of the amount of rotation and scaling present in the detected silhouette.

Another method for making the log-polar mapping invariant to rotation and scale in the field of view is to calculate the magnitude of its Fourier transform. The shift property of the Fourier transform eliminates the rotation and scaling shifts of the log-polar mapping by treating the mapping as a periodic function, as reported for laser radar range imagery by Kolodzy et al. [7]. While this method has merit, it will not be discussed further here; it is the subject of other research [13].

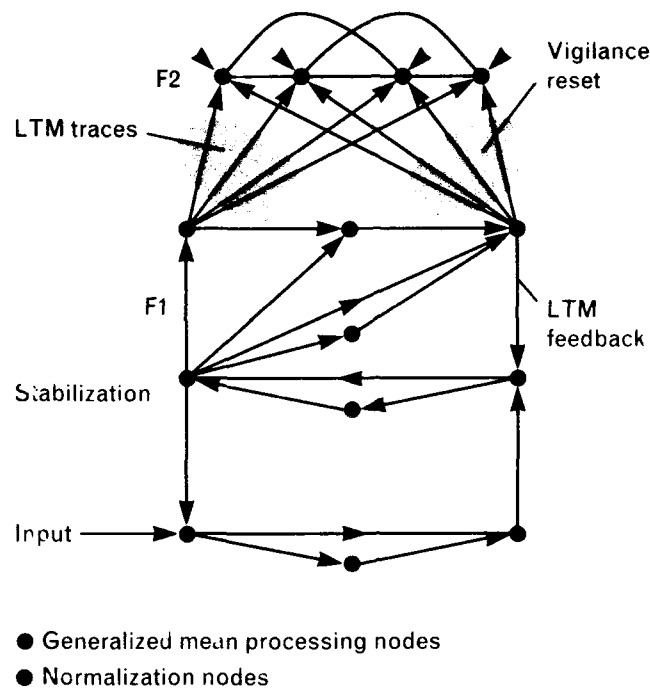
#### Classification

In the final stage of the ATR system, a neural network is used to classify the abstract invariant maps into potential target categories. The adaptive resonance theory (ART) developed by G. Carpenter and S.

Grossberg [14] defines a class of unsupervised neural network classifiers that cluster an  $N$ -dimensional input vector into a finite number of stable categories. This clustering is a necessity if large training sets are to be used.

Supervised networks and/or model-based systems require exact knowledge of the target, or *ground truth*, for each exemplar. For most large systems, thousands (or millions) of training frames would need to be ground truthed, which is a daunting if not impractical task. The ART-2 network, which is illustrated in Figure 14, is basically a two-level correlation classifier; this algorithm has been discussed by R. Lippmann [15] and Menon and Kolodzy [16] to be similar to the  $K$ -means clustering algorithm.

The ART-2 network is different from early ART structures because it is designed to classify analog, rather than binary, input patterns [17]. This analog capability requires a robust structure that pays strict attention to memory stability. The ART-2 network classifies and stores patterns in the following manner. The first layer (F1) normalizes the input with respect to the feedback signal from the second layer (F2), becoming a *short-term memory* (STM) trace. This trace activates nodes in the second layer proportional to the magnitude of its correlation with the corresponding



**FIGURE 14.** Processing diagram of the ART-2 neural network. The input image is presented and stabilized by the node-level processing in the F1 layer; the result is then correlated with stored long-term memory (LTM) patterns in the F2 layer. If the resultant correlation is not large enough with respect to the vigilance parameter, a reset signal is propagated as feedback down to the F1 layer. If none of the LTM patterns are sufficient, then a new LTM pattern is formed.

stored memory patterns, or *long-term memory* (LTM) traces. If the degree of match between the normalized input STM trace and the LTM trace associated with the most highly activated node in F2 passes a vigilance parameter, the STM trace in F1 is learned onto the LTM trace, thus storing the differences between these memory traces. Should a mismatch occur, a reset signal causes the input pattern to select another LTM category. If no existing LTM category can be found that matches the input pattern, a new category is created, which illustrates the ability of the ART-2 network to respond to a novel signal. Generally, similar patterns are categorized together because of high interpattern correlation, and these patterns continually activate the same category node in F2.

The final step of the classifier training is to associate target labels with LTM traces. Each LTM trace for

an individual target is provided a label unique to that target. Multiple LTM traces for an individual target are formed because of either different views or statistical variability of the target. For example, a tank at a head-on perspective looks different from a tank at a broadside perspective, and thus would form two different categories. Also, at a low SNR the signal could change significantly enough to cause the classifier to form a new category if it is presented with a new noise structure.

### Interpretation of Results

The performance of an ATR system can be indicated either by the score of each individual processing module or by the overall system score. This article uses the overall system score as a measure of results, with a higher concentration on the capabilities of the classi-

fier. For supervised classifiers, the performance is commonly measured by the number of correct responses of the system when it is given a test set of input images. The ATR system presented here incorporates an unsupervised classifier, which uses a larger variety of performance measures. This article uses a scoring method based on the number and population distribution of categories created by the classifier during training.

Unsupervised classifiers are generally clustering algorithms that group input feature vectors into a finite number of categories. A user-defined distance metric is used to determine whether an input vector is to be clustered, or *matched*, with an existing category. The number of categories produced (given a specific training set) indicates the ability of the classifier to *generalize*. A classifier responding with more than one category for an object is not unreasonable if the features the ATR is extracting change significantly. For the ATR system presented here, this change in features occurs for the log-polar map when the vehicles are rotated out of plane. A classifier that requires only a few categories to perform recognition is desirable.

Two measures of a classifier's ability to generalize are currently being used: (1) the number of categories required for a given training set, and (2) the number of populated categories formed for the same training set. The differences between these two measures are found in the interpretation of sparsely populated clusters or categories. For example, if 100 inputs produces five categories populated by 95, 2, 1, 1, and 1 examples, respectively, then either five separate categories or one single category with five incorrect responses are necessary.

The trade-offs of these measures are identical to those for fielded ATR systems: performance versus hardware requirements. If every vehicle needs to be recognized, then all possible variations, including those categories individually populated, or *outliers* (those variations whose characteristics are rarely viewed), are required to be modeled and retained in the ATR system. It is possible, as shown by the 100 input examples above, that a significant reduction in the hardware requirements (i.e., memory) of the system can be obtained by allowing a certain reduction in recognition capability.

### *Evaluations Using Laser Radar Imagery*

We have investigated the ability of this ATR system to classify laser radar range imagery of various military targets correctly. This system has been tested on a limited amount of imagery obtained with ground-based sensors built by the Opto-Radar Systems group at Lincoln Laboratory. The results of these tests are presented below.

The full capabilities (and deficiencies) of an ATR system, however, must also be determined, and this determination is possible only through exhaustive testing requiring large amounts of sensor data. Many conditions can be tested to determine the capabilities of the ATR system; we used three conditions: (1) CNR, (2) out-of-plane rotation, and (3) number of pixels on target. Unfortunately, the amount of sensor data required to test these three conditions thoroughly by using real sensor data is prohibitive in both time and cost. The use of synthetic imagery to place bounds on system capabilities is the logical alternative.

Synthetic laser radar range target imagery was generated by using Environmental Research Institute of Michigan (ERIM) wire-frame models of a variety of military and non-military vehicles. Background imagery was generated by using a flat ground plane projected with the attack angle of the sensor. Perfect object extraction from the ground plane was assumed for this study. Sensor statistics (e.g., noise) were added by using the laser radar range and intensity models developed by I. Shapiro et al. [18, 3, 4]. This procedure was used to generate targets for the three test conditions listed above.

### *Ground-Based Sensor Data and Results*

In 1981 at Camp Edwards, Massachusetts, the Opto-Radar Systems group recorded a large database of laser radar imagery of three vehicles—an M-48 tank, an M-113 armored personnel carrier (APC), and an M-110 howitzer. These vehicles were recorded at five orientations with five range backgrounds by using a transportable ground-based laser radar sensor that was the forerunner of the airborne IRAR system. This ground-based sensor allowed us to create a versatile database for testing ATR system performance. Each of the five background scenes consisted of sky, trees,

**Table 5. Classification Results for Three-Target Database of 18 Images**

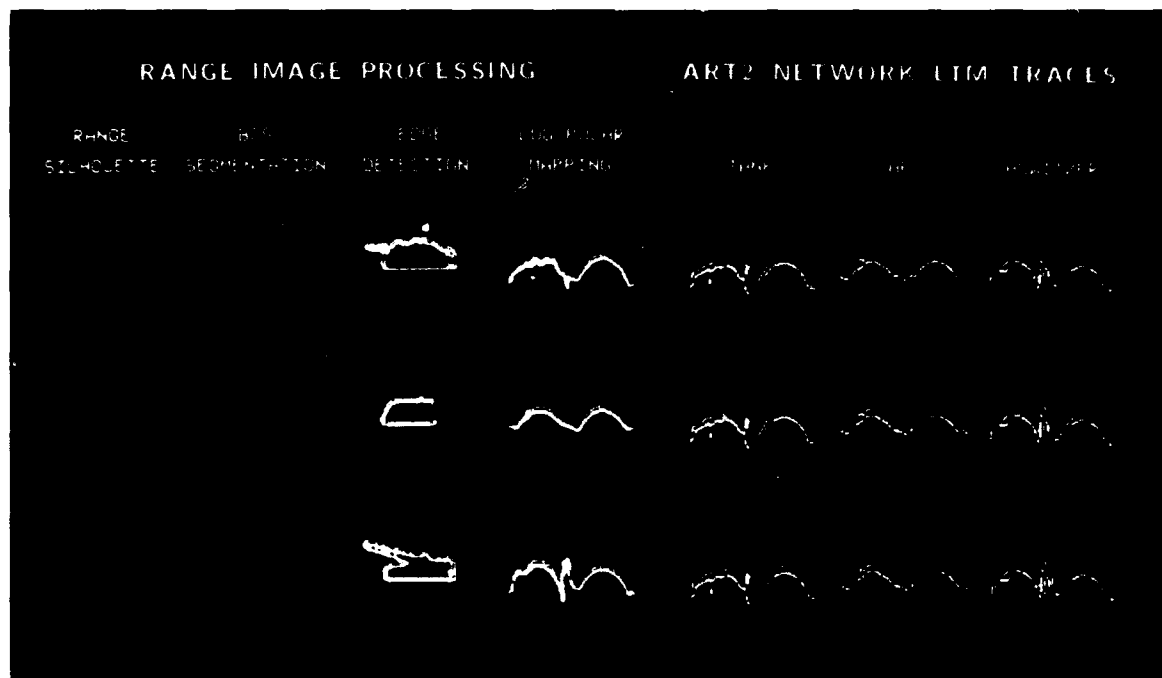
	<i>Number of Categories Formed</i>	<i>False Alarms</i>	<i>ART-2 Vigilance</i>
BCS filled silhouettes	5	1	0.785
BCS maximum edge strengths	5	0	0.694
BCS summed edge strengths	4	0	0.718
Raw extracted target silhouette	3	2	0.790

or hillside, which we created by changing the location of the ground-based sensor relative to the target.

We selected an 18-frame image subset of the Camp Edwards database and processed this image subset through the ATR system. This image subset consisted of three frames of three targets at 750 m and 1000 m in range. The 750-m imagery had a sky background

that provided infinite range contrast between the target and background. The 1000-m imagery had a hillside background that had almost no range contrast because of the high depression angle between the sensor and the target; many pixels in this imagery were only one range count different from the target.

The detection algorithm of the ATR system lo-



**FIGURE 15.** Classification of laser radar range imagery into stable recognition categories by using the ART-2 neural network. The range silhouette is shown for three input images—a tank, armored personnel carrier (APC), and howitzer—followed by the resultant image from the segmentation stage. The edge image is then computed, followed by the result of the log-polar mapping. The right side of the figure shows the three LTM patterns with a red box outlining the matched LTM category for the corresponding input. Note that the LTM patterns are not identical to the input log-polar patterns, because they are an aggregate of all the inputs classified with an individual LTM.

cated and extracted 100% of the targets in the test set. This result was not unexpected for the 750-m imagery. There was significant range contrast in the scene, so verticality measurements alone could be used for detection and the size filter was not required to extract the target. In the 1000-m imagery, however, the background was often only one range count different from the target, which required the size filter to extract accurately the region of interest defined as the target. A detection rate of 100% for the 1000-m targets demonstrated the robust behavior of the detection stage of the ATR system.

The ART-2 neural-network classification stage of the ATR system properly classified 95% of the targets into five stable recognition categories, as listed in Table 5. Sixteen targets formed four categories (specifically, six tanks, five APCs, three 750-m howitzers and two 1000-m howitzers), one 1000-m howitzer formed its own category, and one 1000-m APC was erroneously classified as a tank and counted as a false alarm. This performance is acceptable after careful examination of the imagery. The tanks and APCs formed relatively consistent invariant patterns for classification. The detected howitzers, however, were not

classified consistently because the detection stage either included part of the ground plane or it removed part of the target body.

Figure 15 shows a sample classification result. The images in the left column of the figure are the detected silhouettes determined by using the range imagery of a tank, APC, and howitzer from the detection stage of the ATR system. These silhouettes are processed by the segmentation stage, the edge strengths are computed, and then the edge-strength images are transformed into the log-polar domain, as shown in the next three columns of the figure. The right side of the figure indicates the three LTM traces created by the ATR system after processing the nine 750-m image frames. The red-box highlight indicates the LTM trace with which that particular input image on the left is matched.

Classification performance was investigated for a set of variations to the baseline ATR system. The baseline system uses the edge images computed from the filled silhouettes produced by the segmentation stage as the input to the log-polar map. The filled silhouettes are produced by using the summed-edge compression method. The variations investigated were

**Table 6. The Effect of CNR on the Number of Categories Formed for ATR System\***

<i>CNR (dB)</i>	<i>Percent Anomalies</i>	<i>With Image Cleanup</i>	<i>Without Image Cleanup</i>
100	0.0	8	8
35	0.2	8	8
30	0.6	8	17
25	1.9	8	26
19	7.3	8	
16	13.9	8	
13	25.2	8	
10	42.3	8	
7	63.1	26	

\* Tests included eight vehicles (jeeps, trucks, armored personnel carriers, and tanks) both with and without image cleanup.

the use of the maximum-edge image, the summed-edge image, and the edge image computed from the target silhouette produced by the detection stage. Each of the variations is related to a reduction of processing by either eliminating part of or the entire segmentation stage.

We describe the results of these variations to the baseline system in terms of the number of categories formed and the number of false alarms (false classifications) produced. The goal is to reduce both the number of categories (i.e., produce better generalization of the data) and the number of false alarms. The results given in Table 5 indicate that both the maximum-edge-strength image and the summed-edge-strength image eliminate the false alarms while the summed-edge-strength image also reduces the number of categories. The target-silhouette image further reduces the number of categories while sacrificing false-alarm performance.

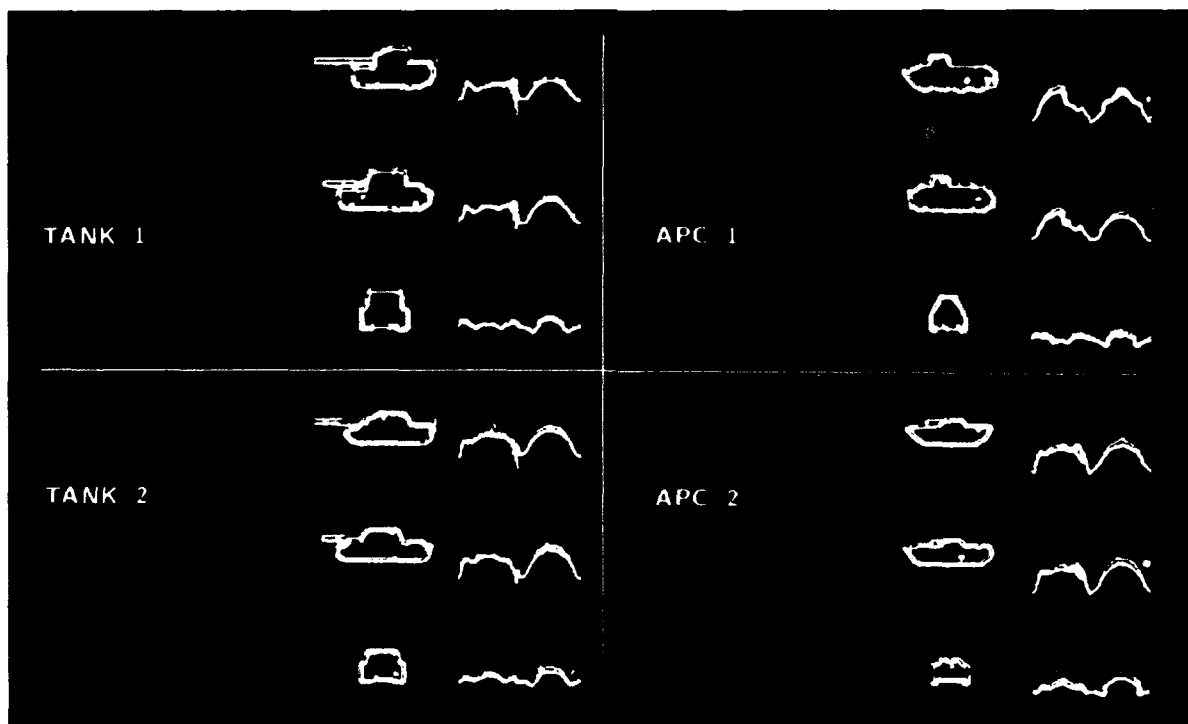
These preliminary results indicate that the classification results are sensitive to the algorithms used in the processing stages prior to the classification stage.

Additional results of tests using a larger database are required before we can conclude that summed edge strengths should be used exclusively as the input to the log-polar map.

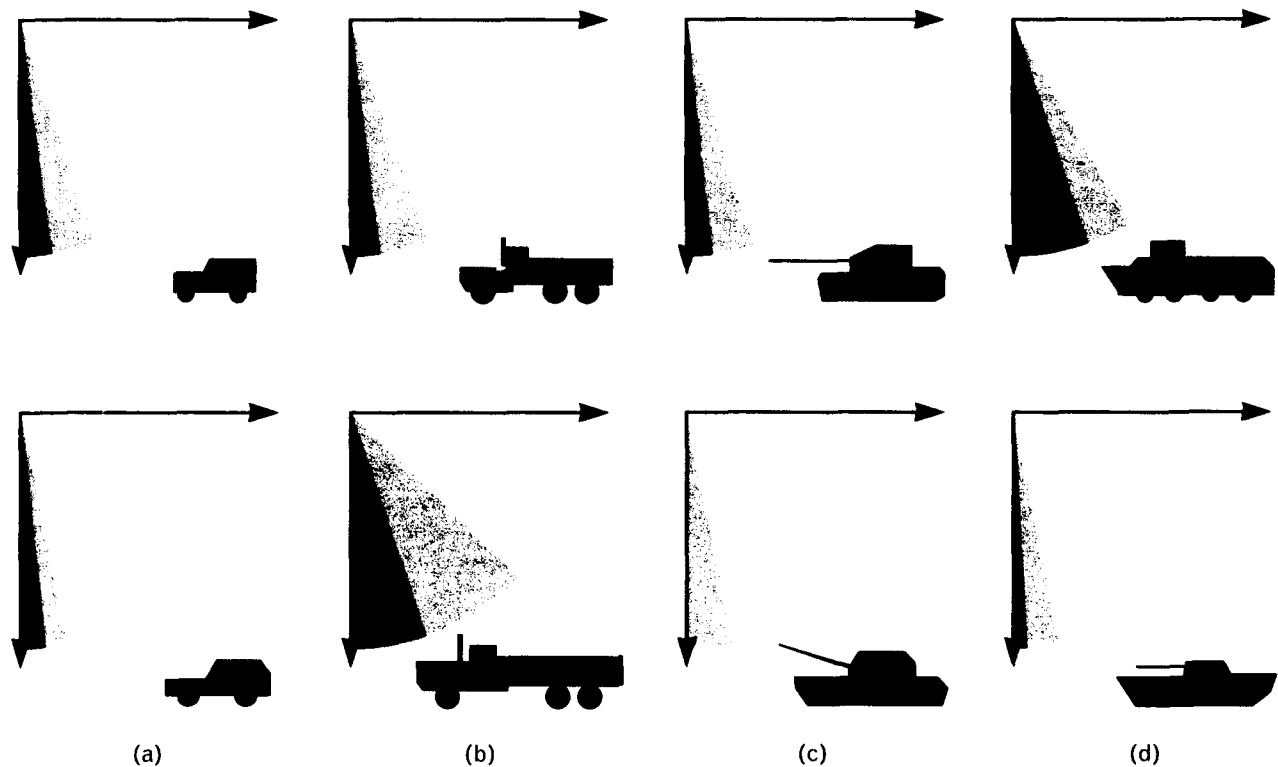
#### *Effect of CNR on Synthetic Broadside Target Recognition*

In the first test we evaluated the effect of CNR on the recognition of broadside targets. We performed two individual experiments to determine the number of categories formed without image cleanup and the number of categories formed with image cleanup. Table 6 shows the results of recognizing eight broadside vehicles (two jeeps, two trucks, two tanks, and two APCs) that are synthetically generated with a sensor of 100- $\mu$ rad angular resolution imaged at a distance of 750 m.

Without image enhancement, a high ART-2 classifier vigilance value was required to separate the eight vehicles. This high value forced the classifier to form multiple categories for each vehicle at a CNR value of 30 dB. The same result is obtained when image en-



**FIGURE 16.** Log-polar maps of tanks and APCs rotated out of plane. The log-polar map of the two tanks are similar for the broadside and near-broadside views but different for the head-on view. The same similarities and differences exist between the log-polar maps for the two APCs.



**FIGURE 17.** Approximate angular extent of each category for recognition of log-polar maps of eight vehicles with out-of-plane rotation; the vehicles are (a) two jeeps, (b) two trucks, (c) two tanks, and (d) two APCs. A total of 31 categories are formed. Each category and its angular extent is depicted by the shaded patterns in the figure. Each vehicle requires only a single category from the broadside view up to 45° of head on or greater. The majority of the categories are in the last 15° from near head on to head on because the log-polar maps change the greatest in that region.

hancement is included, in the form of the Bayesian preprocessor, at a CNR value of 7 dB. A typical operational sensor value of 19 dB at a distance of 1000 m indicates that image cleanup is a necessity. For further details of the experiment and results, see the report by S. Rak [19].

#### *Out-of-Plane Rotation Recognition*

A second test was performed to provide insight into the number of independent categories necessary to distinguish eight vehicles rotated out of plane from broadside to head on (a 90° rotation) [20]. When matched filters are used for recognition, we commonly create filters for every 5° of arc. This test was to provide experimental evidence for the number of categories necessary for recognition. Again, we used the same ATR system with the log-polar maps that we used with the ground-based sensor data.

A visual depiction of the information passed to the

classifier indicates that input to the log-polar maps from broadside to 50° of head on are similar, whereas the maps near head on change radically. Figure 16 shows the log-polar maps for two tanks and two APCs at broadside, 50°, and head-on orientations. Visually, Figure 16 indicates that more categories are necessary for the near head-on orientations while only a few categories are needed for the near-broadside orientations.

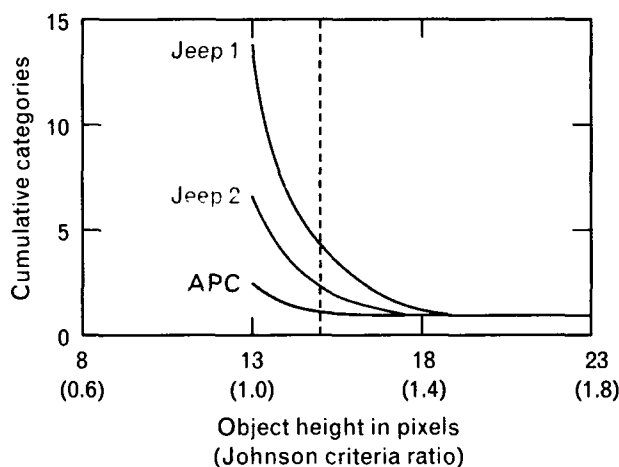
The same test was performed with the eight vehicles rotated from broadside to head on in 1° increments, which created 90 inputs per vehicle. The 720 aggregate inputs were then used to train the classifier, which determined that only 31 categories were necessary to distinguish the eight vehicles at any orientation from broadside to head on. Figure 17 indicates the approximate angular extent of each of the 31 categories. Some vehicles require more categories than others. A general trend seen in this figure is that only



one category is necessary for each vehicle to distinguish the vehicles from broadside to approximately 45° of head on. This result agrees with the intuitive understanding we have when viewing the log-polar maps.

#### Resolution Requirements and the Johnson Criterion

A final test of the ATR system is the comparison between the criteria indicated by J. Johnson [20] and the resolution requirements for recognition and identification. Johnson's work focused on determining the imaging requirements of a sensor to produce a level of discrimination and recognition for human observers. The work consisted of psychovisual experiments on U.S. Army personnel by using image intensifier imagery that is similar in quality to passive-IR imagery. The personnel were shown images of various vehicles at various resolutions and asked to identify the vehicles. The *Johnson criterion* is the number of pixels in a vehicle's minimum dimension (usually height) that is required for a 50% probability of correctly identifying the vehicle.



**FIGURE 18.** Johnson-criterion test to indicate the number of pixels necessary for identification of a target. In this case the targets are two jeeps and an APC. The number of cumulative categories formed for a set of training patterns at each object height in pixels is shown for object heights from 13 to 23 pixels. The increase in the number of categories from the baseline case height of 23 pixels indicates the inability of the classifier to generalize the patterns. The results shown in the figure indicate that the classifier performs well up to 20% above the Johnson criterion of 13 pixels, as indicated by the dashed line in the figure.

We performed an experiment with the spatial extent of each pixel as the variable; this experiment was identical to the one on the effect of CNR described above. Three broadside vehicles were used (two jeeps and an APC) for the training, and the number of pixels in the minimum dimension were varied from 13 to 23. Figure 18 indicates the number of categories formed as a function of the number of pixels in the minimum dimension. For complex vehicle outlines such as the jeeps, the classifier performs well up to 20% greater than the Johnson criterion for identification. For a much simpler vehicle such as the APC, the classifier is more robust and can still identify the vehicle at the Johnson criterion. For more details on the methodology and interpretation of results, see the report by Rak [19].

#### The ATR Evaluation Facility

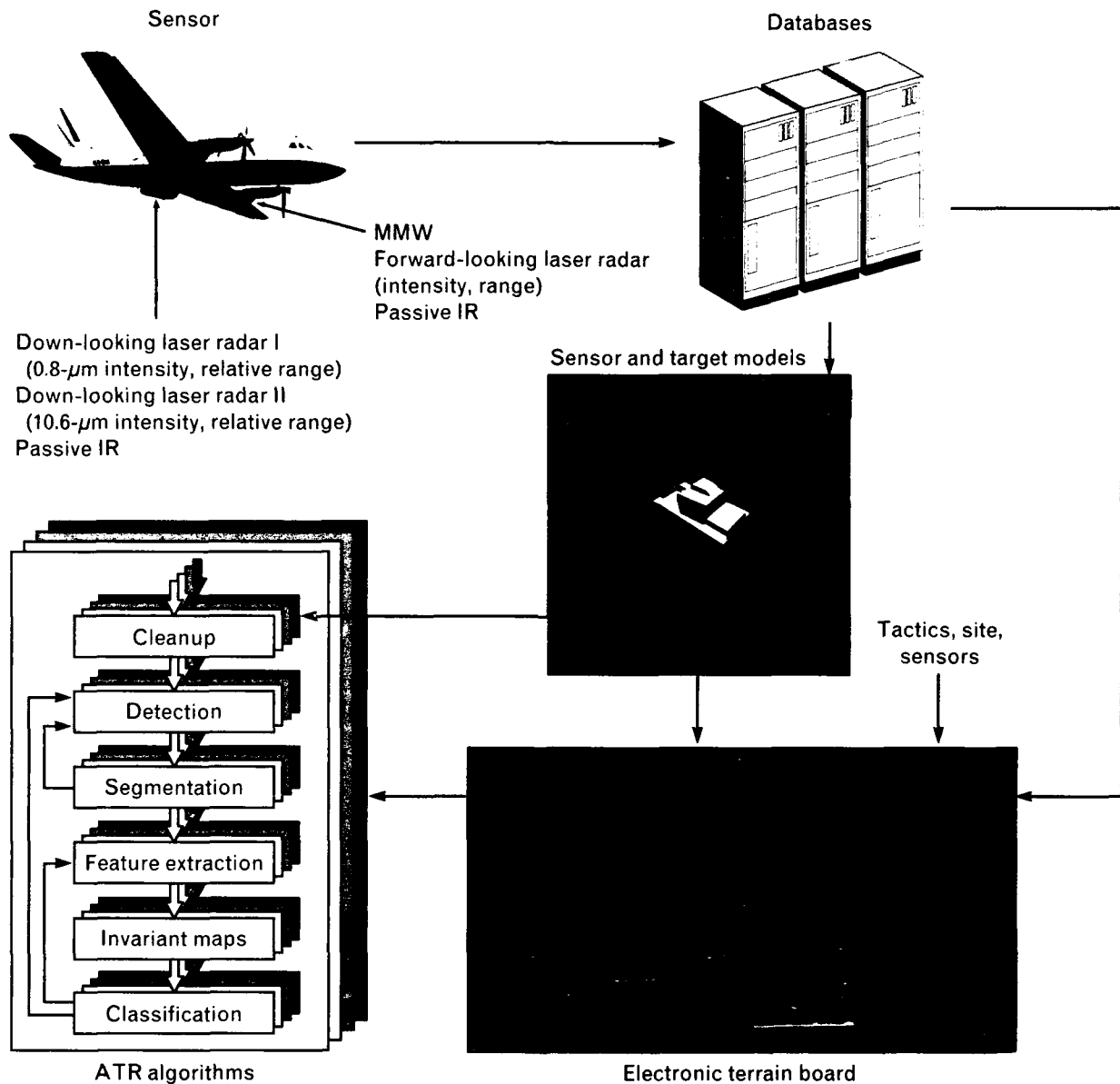
Military applications require the use of ATR systems in both semi-autonomous and autonomous modes (in a semi-autonomous mode we believe in the recognition capabilities of the ATR system enough for a user to apply the results, while in autonomous mode we let the system act on the results on its own). The testing and acceptance of ATR systems for these military applications has proven to be difficult. The resources necessary to provide useful test results are usually overburdening. Either we must use large amounts of real sensor imagery, sometimes in multiple sensor modalities, for each given mission scenario, or we must use synthetically generated data. The real sensor imagery requires expensive and time-consuming efforts to gather the data, while the synthetic imagery places an inherent trust in the validity of the sensor and target models used to generate the synthetic data. The recent development, however, of inexpensive computer graphics workstations and data-processing engines has begun to change the emphasis from measurement missions to computer-generated data.

We are currently developing an ATR evaluation facility (AEF) that exploits the recent developments in computer graphics and data processing to provide an effective test environment. This facility merges high-resolution data, an electronic terrain board (ETB) that combines sensor data with synthetic targets and

sensor models, and an ATR system that is under evaluation. The high-resolution data are taken in the modalities of interest (laser, passive IR, and visible) and stored in databases. The ETB uses the databases along with the sensor and target models to modify the measured imagery for ATR-system sensitivity analyses. This section describes the facility as well as current research on its development.

### Description of the ATR Evaluation Facility

The AEF merges existing sensor data in multiple modalities with synthetic data from sensor and target computer models. Figure 19 shows the conceptual flow of information in the AEF. The airborne IRAR sensor suite, which is described earlier in this article, collects high-resolution imagery in laser intensity,



**FIGURE 19.** The ATR evaluation facility incorporates the down-looking and forward-looking sensor imagery databases, sensor and target models, the electronic terrain board (ETB), and the ATR algorithm suite. Image databases and sensor and target models are fused within the ETB, which allows us to modify target models and vary the measured backgrounds.

range, passive IR, and MMW in a variety of wavebands and view aspects. These data are stored in large databases that are used to refine the synthetic data created from sensor and target models.

The modeling efforts and the databases are merged in the ETB. Most false alarms and missed detections as well as missed classifications of targets are due to the variability of background clutter signals. Model-based systems and trainable recognition systems are developed by using limited target signatures only; unfortunately, these systems do not develop internal models for backgrounds as well. Therefore, we must find a way to merge target signatures, which are predominantly models, with background clutter.

It is difficult, however, to model background signals because of their variable and unpredictable nature. The background models are therefore the weakest link of a completely synthetic sensor image. The combination of measured background imagery with the more well-defined synthetic target models circumvents this problem.

#### *Terrain Database*

The down-looking laser radar sensor described earlier provides high-resolution range imagery. This imagery is a 2½-D representation of the actual terrain and precludes the existence of speckle noise indicative of intensity images. The 2½-D imagery contains the range of the first object or part of object that is interrogated for each pixel. Therefore, any part of an object at a further range or area occluded is not represented in the data. The 2½-D notation indicates that a full 3-D image is produced, although the way we view the scene from above appears as if a blanket were covering the objects in the scene. Only the highest point of a pixel that is interrogated is recorded; any part of an object at a longer range or in an occluded area is not represented in the data. For example, a ball in midair viewed from above is represented as a hemisphere on top of a cylinder because no information is available on the space below the ball. Techniques for combining multiple views are being investigated to alleviate this current limitation.

The down-looking sensor simultaneously measures range as well as laser intensity and passive IR. The existence of 2½-D range imagery lends each of the

sensor domains to coordinate transformations. As described earlier, the ability to transform the range data and subsequent pixel-registered passive-IR data allows the sensor imagery to be used to train and test ATR systems with many viewing aspects. The specific method used for the coordinate transformation can have a dramatic effect both on the requirements for computation and, more importantly, on the quality of the resultant image.

Traditionally, Euler angles have been used to represent coordinate transformations, and these coordinate transformations can be expressed as 3-by-3 rotation matrices. Because the computer graphics community commonly uses rotation matrices, most of the specialized hardware developed to perform coordinate transformations employs this method. This choice has been motivated primarily by the fact that translation and scaling as well as rotation can be represented by one matrix. The same transformations, however, that can be performed by a matrix can be performed with fewer operations by using quaternions [21]

An important consideration in the choice between matrices and quaternions for coordinate transformations occurs when we interpolate between two orientations. Rotation matrices are not well defined for interpolations, because rotations are carried out by three successive rotations about three fixed axes. Because these successive rotations are not commutative, changing the order of the rotations produces different results, which introduces a significant problem known as *gimbal lock*. This problem occurs when the interaction of two rotations aligns two of the three rotation axes and causes a loss in one degree of rotational freedom. Quaternions are free of this problem because the cross-product interaction between successive rotations is preserved [21]. Because of this rotational stability, the aerospace industry for many years has preferred quaternions over matrices defined by Euler angles for spacecraft applications.

Because most computer graphics workstations have hardware that is specifically designed to implement matrix transformations, we must continue to maintain all viewing parameters in matrix form. The AEF system is designed to perform all interpolations by using quaternions, which are then converted to matrix form for rendering.

To illustrate the use of quaternions for interpolating rotations we must first define what a quaternion is and how it is used to perform a rotation. A quaternion consists of two components—a scalar part and a vector part. Consider a quaternion  $q = [s, \mathbf{v}]$ , where  $s$  is a scalar, and  $\mathbf{v}$  is a vector of three elements. In quaternion algebra, addition is defined as

$$q_1 + q_2 = [(s_1 + s_2), (\mathbf{v}_1 + \mathbf{v}_2)],$$

and multiplication is defined as

$$q_1 q_2 = [(s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2), (s_1 \mathbf{v}_2 + s_2 \mathbf{v}_1 + \mathbf{v}_1 \times \mathbf{v}_2)],$$

where  $\mathbf{v}_1 \cdot \mathbf{v}_2$  is the vector dot product and  $\mathbf{v}_1 \times \mathbf{v}_2$  is the vector cross product.

Before we can define rotations using quaternions we need to define the inverse operation

$$q^{-1} = \frac{1}{\|q\|^2} [s, -\mathbf{v}],$$

where

$$\|q\|^2 = s^2 + \mathbf{v} \cdot \mathbf{v}.$$

To rotate a point  $\mathbf{p}$  we embed it into a quaternion as  $[0, \mathbf{p}]$ . Rotation is then defined as

$$\mathbf{v}' = \text{Rot}(\mathbf{v}) = q \mathbf{v} q^{-1},$$

where  $q$  and  $q^{-1}$  are unit quaternions.

One consideration associated with the use of quaternions for coordinate transformations is that rotations are performed on the unit four-dimensional hypersphere, so that, as a result, simple linear interpolation between two orientations gives unequal rotations through the range of orientation values. The unequal rotations occur because the great arc of a unit hypersphere is the spherical equivalent of a line, and the linear interpolation steps fall on unequal portions of the line. These unequal rotations must be compensated for to give a smooth set of intermediate transformations. All of the interpolations between specified positions are performed by using unit quaternions and spherical linear interpolations, and then compen-

sating for unequal rotations [21].

Figure 20 shows an example of the transformation of down-looking range data into various viewing perspectives. Starting with the range data shown in Figure 4, the range data are transformed and displayed as sand-colored video data and synthetically generated laser radar range data in a viewing sequence typical of a target interrogation. In effect, this series of transformations is like an observing eye on a flying carpet; it begins at a long standoff distance at a high altitude, it detects a possible target, it dives to a lower altitude, and it flies along the road to the target. This series of transformations demonstrates how down-looking imagery can be used to train and test an ATR system with many viewing aspects.

### *Synthetic Laser Radar Imagery*

An important element in the ETB is the combination of synthetic imagery and modified sensor imagery. Synthetic imagery is derived from target and background models applied with the appropriate sensor statistics. In some cases, actual sensor imagery can be modified to degrade the quality of the imagery for test purposes. Both of these cases provide the additional flexibility necessary for ATR evaluation. This section describes the methodology used in creating or modifying laser radar imagery.

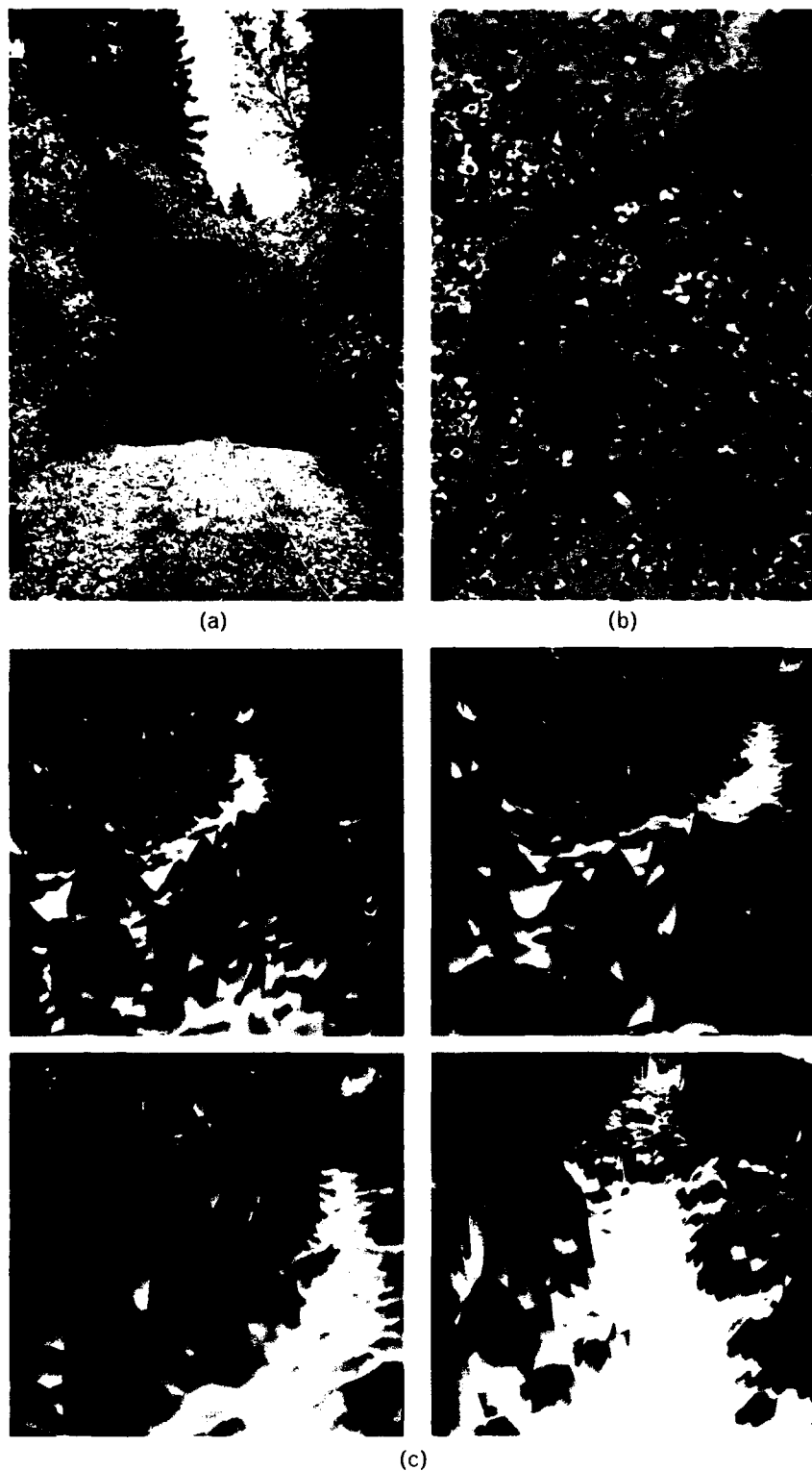
The statistics describing a monostatic pulsed ranging laser radar employing heterodyne detection are described by Hannon and Shapiro [3], and were used to develop a model for laser radar range data. This laser radar model requires that we select the range and CNR value for every pixel as well as the number of range bins  $Q$  available to the signal processor. The probabilities of an anomaly (equally distributed across  $Q-1$  range bins) and for the correct range value are given by

$$P_{I_c} = 1 - e^{-\frac{I^2}{\text{CNR}+1}}$$

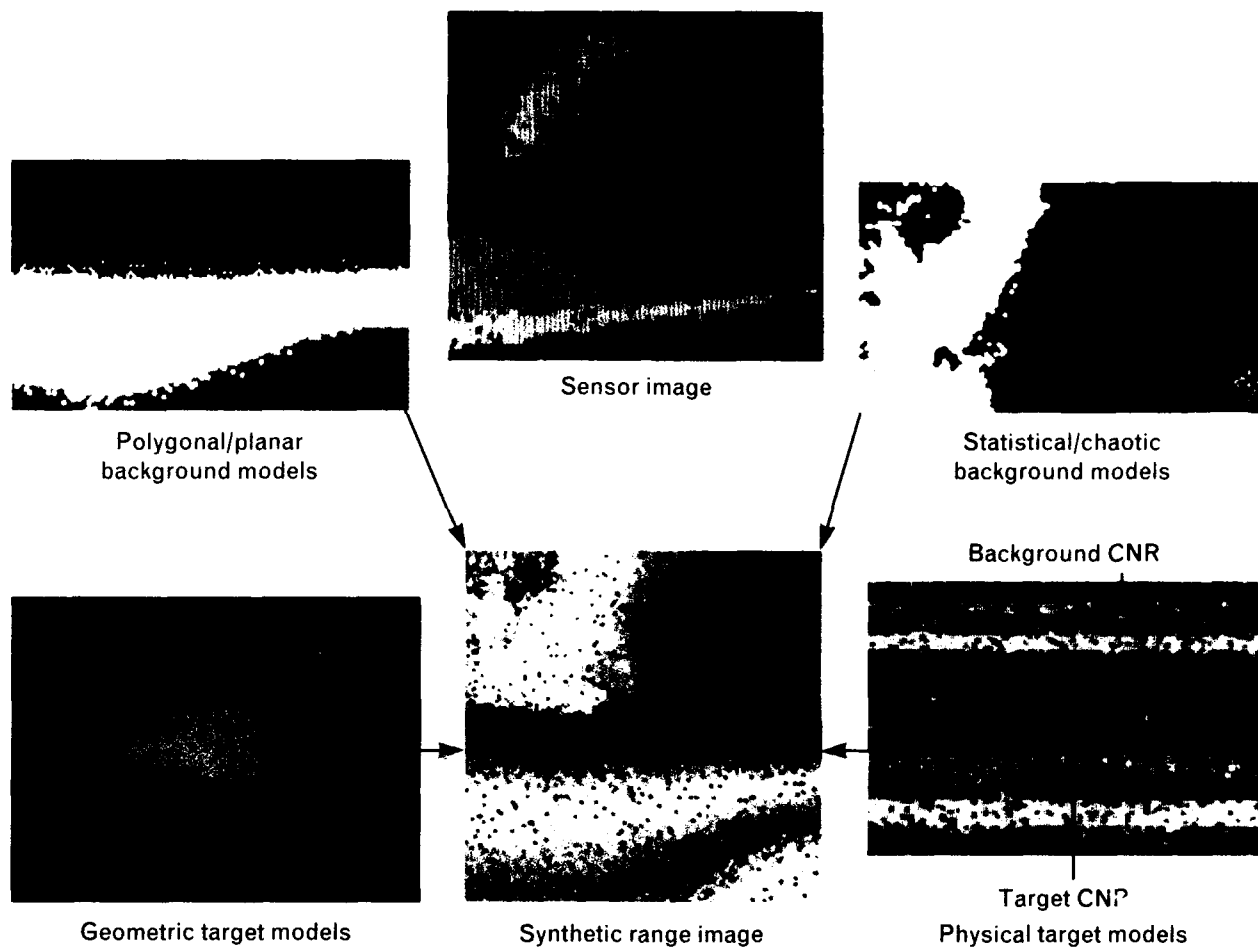
and

$$P_{I_w} = (1 - e^{-I^2})^{Q-1},$$

where  $P_{I_c}$  and  $P_{I_w}$  are the detected intensity probabilities from the correct range bins and the wrong



**FIGURE 20.** Down-looking laser radar data is transformed into a three-dimensional terrain-map view. (a) Photographic ground truth of a camouflaged truck, (b) down-looking laser radar image, and (c) a sequence of four views that were formed by using 3-D transformed laser radar imagery to "fly over" the target.



**FIGURE 21.** Scene decomposition and synthesis of a laser radar range image. The original sensor image (upper center) is decomposed into polygonal background (upper left); statistical, chaotic, or fractal background (upper right); facet target models (lower left); and physical target-background parameters (lower right). A new laser radar image is then synthesized (lower center).

range bins, respectively. These two probabilities are used in conjunction with a random-number generator to provide two random draws. The maximum value of the two random draws is selected as the intensity  $I$ .

The application of the described statistics can be demonstrated through an example. Figure 21 depicts the decomposition of a sensor laser radar range image into four primary parts. The background and target models constitute three parts: polygonal background models for relatively uniform terrain; statistical, chaotic, or fractal models for fragmented terrains such as foliage; and the geometric target models generated from wireframe or facet libraries (ERIM) or solid

geometry libraries (U.S. Army Ballistic Research Laboratory). The physical target models, which are described by the CNR for each pixel, are used to compute the intensity and range value by using the Hannon laser radar model [3]. In the example shown, a variety of fractal dimensions were attempted and a best visual fit was selected for the foliage. A uniform CNR value of 17 dB (which is a typical value for imaging generic terrain by the IRAR sensor at 700 m) was used to generate the synthetic range image. The generated image is visually similar to the original sensor image.

### Summary

A flyable, multisensor system has the ability to mea-

sure a combination of range, Doppler, laser intensity, and thermal signatures in both the forward-looking and down-looking aspects. Statistical advantages for incorporating multidimensional information exist for target-detection applications using theoretical analyses and heuristic algorithms. The use of multiple sensor modalities also provides some hope to address the vexing issues of ATR.

A modular, hybrid ATR system has been described that fuses statistical, model-based, and neural network processing structures. The system has been tested on laser radar range imagery as well as synthetic range imagery incorporating pulsed laser radar statistics. Results created by using the synthetic imagery indicate that target identification can occur in imagery with over 50% of the pixels corrupted by noise. Tests with out-of-plane rotated vehicles indicate that a finite number of nonuniform angularly spaced projections can be learned by the system to provide target identification. The current system can also provide identification with spatial resolution as low as 20% above the Johnson criteria.

To continue to test and evaluate complicated ATR systems, an ATR evaluation facility is being constructed to provide real, synthetic, and hybrid sensor image input to a selected ATR. This facility uses the available high-resolution down-looking laser radar range imagery and high-fidelity target models to generate the various operational scenarios.

### **Acknowledgments**

The work described in this article is the combined effort of a large group of scientists and technical support personnel. The author acknowledges the efforts of R. Hull, T. Quist, and S. Prutzer for the development of the multisensor measurement system, and the efforts of D. Biron, E. Van Allen, S. Rak, M. Menon, and J. Baum for the development of the ATR system and the subsequent algorithms. The author also thanks A. Gschwendtner, P. DiCaprio, and H. Thomas for their contributions to the ATR evaluation facility.

This work was sponsored by the Advanced Research Projects Agency, the Balanced Technology Initiative Program, and the U.S. Air Force under an Electronic Systems Division contract.

## REFERENCES

1. D. Biron, R. Hull, and T. Quist, private communication.
2. M.B. Mark, *Multipixel, Multidimensional Laser Radar System Performance*, Ph.D. thesis, MIT Dept. of Electrical Engineering and Computer Science (1986).
3. S.M. Hannon and J.H. Shapiro, "Laser Radar Target Detection with a Multipixel Joint Range-Intensity Processor," *SPIE* **999**, 162 (1989).
4. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Recognition Mach. Intell.* **PAMI-6**, 721 (1984).
5. S. Grossberg and E. Mingolla, "Neural Dynamics of Perception Grouping: Textures, Boundaries, and Emergent Segmentations," *Perception and Psychophysics* **38**, 141 (1985).
6. D. Hubel and T. Wiesel, "Functional Analysis of Macaque Monkey Visual Cortex," *Proc. Royal Soc. of London (B)* **198**, 1 (1977).
7. P.J. Kolodzy, M.M. Menon, and E.V. Allen, private communication.
8. E.J. Van Allen and P.J. Kolodzy, "Application of a Boundary Contour Neural Network to Illusions and Infrared Sensor Imagery," *IEEE First Intl. Conf. on Neural Networks* **4**, San Diego, 21-24 June 1987, p. 193.
9. E.J. Van Allen, private communication.
10. J.G. Verly, R.L. Delanoy, and D.E. Dudgeon, "Machine Intelligence Technology for Automatic Target Recognition," *Lin. Lab. J.* **2**, 277 (1989).
11. D.E. Dudgeon, P.J. Kolodzy, C. Mehanien, M.M. Menon, S.J. Rak, E.J. Van Allen, J.G. Verly, and R.L. Delanoy, private communication.
12. E. Schwartz, "Computational Anatomy and Functional Architecture of Striate Cortex: A Spatial Mapping Approach to Perceptual Coding," *Vision Research* **20**, 645 (1980).
13. S.J. Rak, D.G. Biron, P.J. Kolodzy, M.M. Menon, and E.J. Van Allen, private communication.
14. G. Carpenter and S. Grossberg, "Neural Dynamics of Category Learning and Recognition: Attention, Memory Consolidation, and Amnesia," in *Brain Structure, Learning, and Memory*, eds. J. Davis, R. Newburgh, and E. Wegman (AAA Symposium Series, 1985), pp. 1-49.
15. R. Lippmann, "An Introduction to Computing with Neural Networks," *IEEE ASSP Mag.* **3**, 4 (Apr. 1987).
16. M.M. Menon and P.J. Kolodzy, "A Comparative Study of Neural Network Classifiers," presented at the First Intl. Neural Networks Symp. (INNS), Sept. 1988.
17. G. Carpenter and S. Grossberg, "Self-Organization of Pattern Recognition Codes for Analog Input Patterns," *Appl. Opt.* **26**, 4919 (Dec. 1987).
18. J.H. Shapiro, R.W. Reinhold, and D. Park, "Performance Analyses for Peak-Detecting Laser Radars," *SPIE* **663**, 38 (1986).
19. S. Rak and P. Kolodzy, "Performance of a Neural Network Based 3-D ATR System," *Project Report NN-6*, Lincoln Laboratory (May 1991).
20. J. Johnson, "Analysis of Image Forming Systems," in *Proc. Image Intensifier Symp.* (U.S. Army ERDL, Fort Belvoir, VA, Oct. 1958).
21. K. Shoemake, "Quaternion Calculus for Animation," in "Math for SIGGRAPH," ACM SIGGRAPH '89 Course Notes, pp. 187-205 (Boston, MA), July 31-August 4, 1989.





**PAUL J. KOLODZY**

is assistant leader of the Opto-Radar Systems group. He received a B.S. degree from Purdue University, and M.S. and Ph.D. degrees from Case Western Reserve University, all in chemical engineering. His current areas of research include neuromorphic systems, automatic target recognition systems, and advanced distributed simulation technology. In 1983, he was a visiting scientist working in the area of laser remote measurement systems at Risø National Laboratory in Roskilde, Denmark. He was cochairman of the Simulation/Emulation Tools & Techniques Panel of the ARPA Neural Network National Study, and he is a member of the Sensor Fusion program committee. Paul has been at Lincoln Laboratory since 1986.

# An Efficient MRF Image-Restoration Technique Using Deterministic Scale-Based Optimization

Murali M. Menon

■ A method for performing piecewise smooth restorations on images corrupted with high levels of noise has been developed. Based on a Markov Random Field (MRF) model, the method uses a neural network sigmoid nonlinearity between pixels in the image to produce a restoration with sharp boundaries while providing noise reduction. The model equations are solved with the Gradient Descent Gain Annealing (GDGA) method—an efficient deterministic search algorithm that typically requires fewer than 200 iterations for image restoration when implemented as a digital computer simulation. A novel feature of the GDGA method is that it *automatically* develops an annealing schedule by adaptively selecting the scale step size during iteration. The algorithm is able to restore images that have up to 71% of their pixels corrupted with non-Gaussian sensor noise. Results from simulations indicate that the MRF-based restoration remains useful at signal-to-noise ratios 5 to 6 dB lower than with the more commonly used median-filtering technique. These results are among the first such quantitative results in the literature.

**A**N IMAGE-RESTORATION METHOD that reduces noise while preserving naturally occurring boundaries in a scene is presented. The method is useful as a preprocessor to enhance the performance of automatic target-recognition systems.

Target recognition is a process that can involve many stages, including measurement, preprocessing, detection, segmentation, feature extraction, and classification. For adequate recognition performance in a noisy environment, it is often important that the preprocessing stage be capable of restoring measured images. (We justify this statement in the section "Simulation Results.") The restoration should reduce the variability in the scene that results from measurement noise and clutter while preserving important features that make targets separable in the classification stage.

Both objectives can be accomplished by using prior statistical knowledge of the measurement process and the clutter in the scene, or by using an empirical formulation of the desired restoration. (Details of using either a statistical or empirical formulation are contained in the following section.)

Using the latter approach, the work described in this article is based on an empirical image-restoration model that requires nearest neighbor pixels to have similar values (smoothing), without losing fidelity to the original measurement. The pixel interaction of the model smooths small pixel differences, but allows large differences to remain as a discontinuity (edge). If detailed statistical information concerning the measurement and scene is available, the information can be quantitatively incorporated into the

image-restoration model.

This article describes an image-restoration model that is based on a neural network formulation using Markov Random Fields (MRF), as described in the box, "Markov Random Fields." In the model, a neural network sigmoid function provides pairwise pixel interaction potentials. The function behaves quadratically for small differences but saturates for large differences. The MRF property of the model allows an image to be, in effect, decoupled into a large number of connected local neighborhoods, each of which can be processed independently. The local-neighbor information is propagated during iteration such that a global image restoration is effected when the system reaches a steady state. The restored image can be found by solving an optimization problem that depends on the pixel interaction potentials. The MRF property that allows each pixel update to depend only on a local neighborhood of pixels eases the computational burden. For the case of a Gaussian pixel interaction, the potential function is quadratic, leading to a simple optimization problem that involves the solution of a large set of linear equations. For sigmoid interaction potentials (the present work), a difficult high-dimensional nonlinear optimization problem results. Stochastic methods are commonly used to solve such problems, but such methods are often very slow and sensitive to the choice of annealing schedule. We propose the novel deterministic Gradient Descent Gain Annealing (GDGA) method for solving high-dimensional nonlinear optimization problems. This method is fast and *automatically* chooses an annealing schedule. GDGA is used to solve optimization problems resulting from the neural-network-based MRF image-restoration model. Previous deterministic annealing work, such as mean field annealing [1, 2], does not incorporate an automatic annealing schedule.

The utility of the MRF model in restoring images corrupted with varying levels of non-Gaussian measurement noise has been investigated. Model performance has been evaluated quantitatively in terms of target detection and recognition, and the performance has been compared to that of the commonly used median-filtering technique. The quantitative results reported in this article are among the first such results

in the literature. Because of prohibitive computational requirements, few quantitative characterizations of image restoration algorithms have been performed. Most work in the literature has compared the restored imagery qualitatively, rather than determining the effect of the restoration stage on the *overall* system performance.

The same model can be applied to a large number of sensor measurements (Doppler, intensity, passive infrared, range, and video) by the adjustment of a single parameter. This feature is especially relevant for hardware implementation because it allows a single chip to be used for processing a wide variety of imagery. The model has a massively parallel architecture with local neighbor pixel interactions (four nearest neighbors) and can be implemented on a parallel-processing computer or a custom analog VLSI chip. Implementation of the model in analog VLSI would allow video-rate restoration of  $512 \times 512$  pixel images.

## Background

In this section the Bayesian formulation of image restoration is reviewed to show the formal connections to the restoration method that is the subject of this article. The Bayesian formulation relates the posterior probability that an estimate of the true image  $x^r$  is obtained given a measured image  $x^m$  and the prior probabilities:

$$P(x^r | x^m) = \frac{P(x^m | x^r)P(x^r)}{P(x^m)}. \quad (1)$$

The term  $P(x^m | x^r)$  incorporates prior knowledge of the measurement process, and  $P(x^r)$  incorporates prior knowledge of the scene. The present model finds an estimate  $\hat{x}^r$  that approximately maximizes  $P(x^r | x^m)$  given the measurement  $x^m$  and prior knowledge of the scene in the form of  $P(x^r)$ . In the present model each pixel depends only on its four surrounding neighbors and the measured pixel as shown in Figure 1.

The probabilistic (Bayesian) formulation is equivalent to a physical system description in terms of an energy [3]:

$$E \propto -\log P(x). \quad (2)$$

## MARKOV RANDOM FIELDS

A SERIES OF EVENTS in time form a *Markov Chain* if the probability of the outcome of an event at time  $t + 1$  depends only on the outcome of the event at time  $t$ . This concept can also be applied to processes on a lattice. A *Markov Random Field (MRF)* defined on a lattice implies that the update of a pixel at site  $ij$  depends only on the values of pixels in a local neighborhood of sites  $N_{ij}$  (Figure A). In terms of conditional probabilities,

$$\begin{aligned} P(X_{ij} = x_{ij} | X_{lk} = x_{lk}, \\ lk \in \text{lattice}, lk \neq ij) \\ = P(X_{ij} = x_{ij} | X_{lk} = x_{lk}, \\ lk \in N_{ij}), \end{aligned}$$

where  $X_{ij}$  is the real-value distri-

bution of a random variable associated with lattice site  $ij$  and  $x_{ij}$  is the specific value of the variable at that site. Thus the definition of an MRF on a lattice transforms a global problem into a more computationally tractable local problem.

It is also true that an MRF on a lattice has the following energy-based formulation:

$$P(x) = \frac{e^{-U(x)}}{Z},$$

where  $U$  is the global potential function for the entire lattice and  $Z$  is the partition function, which normalizes the probability  $P(x)$  to a range from 0 to 1.

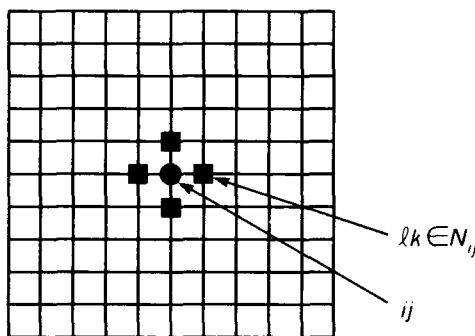
In the present work the energy  $E(x)$  is defined over all independent pairs of sites  $p$  on the lattice:

$$U(x) = \sum_p E_p(x).$$

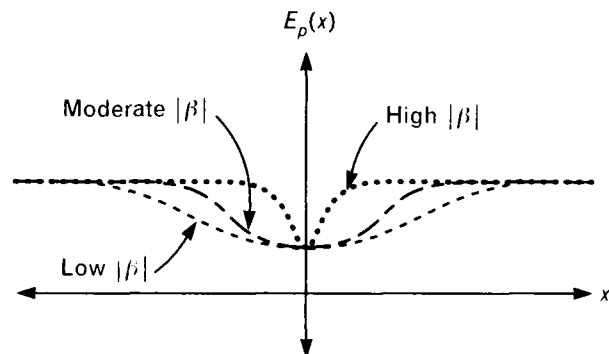
The specific energy of interaction for a pair of sites is given by a *sigmoid function*:

$$E_p(x) = \frac{1}{1 + e^{\beta x^2}},$$

where the gain ( $\beta < 0$ ) defines the scale of the sigmoid, as shown in Figure B. A small magnitude of the gain produces a large-scale (broad) sigmoid, while a large magnitude of the gain produces a small-scale (narrow) sigmoid. For either case, the sigmoid function has the property that the response saturates after the input exceeds a certain level. For a high magnitude of the gain, note that the sigmoid saturates very quickly, even for small inputs. The gain in the sigmoid function is inversely proportional to the temperature of an energy-based formulation.



**FIGURE A.** Markov Random Field (MRF) defined on a lattice. In the figure, the update of a pixel at site  $ij$  depends only on the values of pixels at sites  $lk$  in a local neighborhood of sites  $N_{ij}$ .



**FIGURE B.** Sigmoid function  $E_p(x)$  for different magnitudes of the gain  $\beta$ . A small magnitude of the gain produces a large-scale (broad) sigmoid, while a large magnitude of the gain produces a small-scale (narrow) sigmoid.

Thus an MRF image processor may be specified by defining the energy function rather than the probabilities. This empirical approach is used in the present work.

### Model Description

Equation 2 indicates that a minimization of the energy will result in a maximization of the probability  $P(x^r | x^m)$ . The total system energy can be expressed as the sum of a field term (which is due to the measured image) and a surround term (which is due to the neighbor interactions):

$$E = \lambda E^F + E^S. \quad (3)$$

The field coupling  $\lambda$  in Equation 3 is an adjustable parameter that determines the importance of the measurement term relative to the surround term: a small value of  $\lambda$  produces a highly smoothed image with little contribution from the measured image, whereas a large value essentially reproduces the measured image. The sigmoid function is used in both terms. The field term is given by

$$E^F = \sum_{ij} \frac{1}{1 + e^{\beta^F (\Delta_{ij}^m)^2}}, \quad (4)$$

where  $\Delta_{ij}^m$  is the difference between the restored and measured pixels (i.e.,  $\Delta_{ij}^m = x_{ij}^r - x_{ij}^m$ ), and  $\beta^F$  is the saturation gain term. The sigmoid function is also used for the surround term:

$$E^S = \sum_p \frac{1}{1 + e^{\beta^S (\Delta_p^s)^2}}, \quad (5)$$

where  $\Delta_p^s$  is the surround pair difference, i.e.,  $\Delta_p^s = x_{p1}^r - x_{p2}^r$ , where  $p$  refers to all independent nearest neighbor pixel pairs in the image, and  $p1$  and  $p2$  refer to the members of a pair). Note that for an  $M \times N$  lattice there are  $(N-1)M$  horizontal pairs and  $(M-1)N$  vertical pairs for a total of  $2MN - M - N$  independent pairs.

The estimate of the original image  $\hat{x}^r$  that minimizes the system energy is obtained with a deterministic search procedure. The present work uses the GDGA deterministic search (described in the subsec-

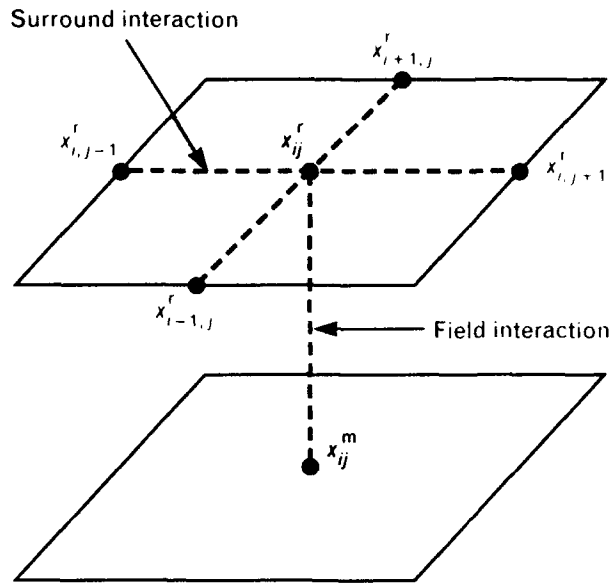


FIGURE 1. Nearest-neighbor architecture used in the Markov Random Field (MRF) image restoration.

tion "Deterministic Solution") to decrease the saturation gain term  $\beta$  in Equations 4 and 5 from  $-0.001$  to  $-10.0$ . (Note that the gain is negative in the present formulation.)

The saturating aspect of the sigmoid function (from the neural network literature [4]) in Equation 5 allows the formation of sharp boundaries between dissimilar regions. The main advantage of using a sigmoid surround term is that sharp segmentations can be obtained without a separate "line process" [3], which would require solving  $2MN - M - N$  extra equations. Hence the sigmoid term clearly reduces the computational load. For the same reason, a sigmoid function is used for the field, or measurement, term. The sigmoid function solves the problem of providing smoothing (noise reduction) while preserving naturally occurring boundary information in the scene.

### Optimization Methods

#### Stochastic Solution

To solve the nonlinear optimization problem suggested by Equation 3, researchers have often attempted stochastic methods, which do not require the derivative of the energy with respect to the restored

state and, as a consequence, can be used for a wide range of optimization problems. Stochastic methods are also well suited for high-dimensional problems that are characterized by many acceptable solutions (restored states) all having approximately the same energy. Image restoration requires a solution with low energy, but does not need *the* global minimum.

The present work derives a stochastic solution by relating the statistical description of the problem to an energy-based representation. By making a correspondence to a physical system at thermal equilibrium, we can express the formulation in Equation 1 in terms of minimizing the energy of a system. The probability that a physical system in equilibrium with a heat bath at temperature  $T$  is in state  $i$  with energy  $E_i$  is given by the Boltzmann distribution:

$$P_T(x = i) = \frac{e^{-E_i/k_B T}}{Z(T)},$$

where  $k_B$  is the Boltzmann constant and  $Z(T)$  is the partition function, which is simply the sum of the exponential term  $E_i/(k_B T)$  over all possible states  $i$ .

It is assumed that the solutions to the optimization problem are equivalent to the states of a physical system and the cost of a solution corresponds to the energy of a state. Asymptotic convergence to a set of globally optimal solutions can be obtained provided that the different states are generated properly and the appropriate conditions are used to decide whether a given state should be accepted [5]. Stochastic methods for solving the optimization problem involve starting at a high temperature and annealing (i.e., reducing) the temperature until the system "freezes" to the minimum energy state. Ideally, the procedure would be implemented reversibly such that the system is always at thermal equilibrium and a true global minimum is reached rather than a metastable state.

Stochastic methods for solving nonlinear optimization problems typically use a simulated annealing method [6] combined with a Monte Carlo technique such as the Metropolis algorithm [7] or the Gibbs

sampler [8]. For a comprehensive study that investigates the application of simulated annealing to image reconstruction, see Reference 3 by S. Geman and D. Geman. The problem with such stochastic solution techniques is that a good annealing schedule is difficult to determine, and the solution time can be prohibitive in terms of the number of iterations required because an equilibrium must be reached at each stage of annealing. At high temperatures a large temperature step is possible because the search covers a wide range of the state space. As the temperature is lowered, however, the system often reaches a critical point below which the state is "frozen," analogous to the phase diagram of real physical systems. If the critical point on the energy-versus-temperature curve were known, then large steps could be taken before the critical point were reached and small steps afterwards. Unfortunately, the "phase diagram" depends on the initial measurement, or field, term.

In practice a conservative annealing schedule is often used:

$$T \propto \frac{1}{\log k},$$

where  $k$  is the iteration number and  $T$  is the temperature. Such a schedule can require hundreds of thousands of iterations or more to produce an acceptable restoration. Automated Local Annealing (ALA) has been suggested to provide an automatic annealing schedule for neural networks [9], but the procedure is not directly applicable to an image-restoration formulation.

Another problem is that the computational expense of the stochastic method also depends on the number of allowable states per image pixel. An image with 8-bit pixels requires many more iterations for the full exploration of the state space as compared to, for example, a 4-bit image. Indeed, the solution of such nonlinear optimization problems remains a challenging research area.

#### *Deterministic Solution*

The large number of iterations that the stochastic approach requires in practice has motivated the use of a deterministic solution technique to solve the nonlinear image-restoration problem. The deterministic

approach attempts to minimize the system energy by iteratively updating pixel values across the lattice until a steady state is reached. In the approach, the use of high gain values for the iterative solution to Equation 3 produces a restored image with sharp boundaries. (Note: In the analogy of the physical system discussed earlier, a high gain value corresponds to a low temperature, or a small scale in that a small change in the input to the sigmoid function will produce a large change in the output.) The use of high gain values, however, will most likely lead to the procedure's being trapped in a local minimum. To remedy this problem we have developed the GDGA technique, which starts the solution procedure at a low gain (i.e., a high temperature, or large scale). The intermediate solution at low gain is then used as an initial condition to the problem at a higher gain, and the procedure is repeated until the final desired gain values are achieved. Solving a series of problems each at higher gain values is equivalent to temperature annealing in the stochastic approach. In addition, we have developed an *automatic annealing* (gain increase, or scale decrease) schedule that is described below.

An equation of motion based on the total energy from Equation 3 is defined by

$$\frac{\partial x_{ij}^r}{\partial t} = -\nabla_{x^r} E, \quad (6)$$

where  $t$  represents a pseudo-time quantity. If Equations 4 and 5 are substituted for the total energy term in Equation 6, then the equation of motion for a single pixel at a lattice site  $ij$  is

$$\begin{aligned} \frac{\partial x_{ij}^r}{\partial t} = & \frac{-\partial}{\partial x_{ij}^r} \sum_{lk} \left( \frac{\lambda}{1 + e^{\beta^l (\Delta_{lk}^m)^2}} \right) \\ & - \frac{\partial}{\partial x_{ij}^r} \sum_p \left( \frac{1}{1 + e^{\beta^s (\Delta_p^s)^2}} \right), \end{aligned}$$

where  $\sum_{lk}$  refers to all of the lattice sites in the image and  $\sum_p$  refers to all of the independent pixel pairs in the lattice. In the present work the MRF is given by the two horizontal and two vertical pairs associated with a given lattice site  $ij$ , resulting in a neighborhood

$N_{ij}$  given by

$$N_{ij} = \{x_{i+1,j}, x_{i-1,j}, x_{i,j+1}, x_{i,j-1}\}.$$

Hence, with this local neighborhood the update of a pixel at lattice site  $ij$  depends only on the pixel's four nearest neighbors.

The objective is to find the steady-state solution to Equation 6 that results in a state  $x^r$  that minimizes the system energy. The particular form of Equation 6, the equation of motion, guarantees that the steady-state solution minimizes the energy. This relationship can be shown by using the identity

$$\frac{\partial E}{\partial t} = \frac{\partial E}{\partial x^r} \frac{\partial x^r}{\partial t} \quad (7)$$

and substituting for  $\partial x^r / \partial t$  from Equation 6 into Equation 7, resulting in

$$\frac{\partial E}{\partial t} = - \left( \frac{\partial E}{\partial x^r} \right)^2.$$

In the present work the GDGA deterministic technique is used to minimize the energy. This formulation is similar to the Graduated Non Convexity (GNC) approach of A. Blake and A. Zisserman [10] and the technique used by Y.G. LeClerc [11], and has some similarity to mean field annealing [1]. We have found GDGA to be substantially faster than the stochastic techniques described in the literature. The GDGA technique iteratively solves Equation 6 by calculating the gradient of the energy and updating the state (similar to an Euler solution of a system of coupled differential equations). In this approach the magnitude of the gain terms  $\beta^l$  and  $\beta^s$  in Equations 4 and 5 are increased from a value starting at 0.001. At small gain magnitudes the restoration acts to smooth the image because the energy terms are approximately locally quadratic with the pixel difference. (An energy term that is quadratic generates a larger penalty for larger pixel differences. Hence a smooth image, i.e., an image with equal pixel values, minimizes this energy.) Also, at small gain magnitudes all edges in the image are smoothed, resulting in a blurred image. As the magnitude of the gain is increased the natural-

ly occurring boundaries in the measured image start to appear, and eventually a sharp segmentation results.

The steady-state solution of Equation 6 at a given gain value is found by setting the gradient of the energy to zero and iteratively solving for the new pixel value. The deterministic technique is implemented with a fixed-point iteration around each pixel, in which the pixels are updated with a Jacobi (fully parallel) scheme [12]. In the technique, the gradient of the energy is set to zero, and the term  $x_{ij}$  is updated based on the old values of its neighbors:

$$x_{ij}^{r(\text{new})} = \frac{\beta^S(A) + \lambda\beta^F x_{ij}^m g_{ij}^F}{\lambda\beta^F g_{ij}^F + \beta^S(B)},$$

where

$$\begin{aligned} A &= g_{i,j-1}^S x_{i,j-1}^{r(\text{old})} + g_{i,j+1}^S x_{i,j+1}^{r(\text{old})} + g_{i-1,j}^S x_{i-1,j}^{r(\text{old})} \\ &\quad + g_{i+1,j}^S x_{i+1,j}^{r(\text{old})}, \text{ and} \\ B &= g_{i,j-1}^S + g_{i,j+1}^S + g_{i-1,j}^S + g_{i+1,j}^S. \end{aligned} \quad (8)$$

In Equation 8 the nonlinear term  $g_{ij}$  is given by

$$g_{ij} = \frac{e^{\beta(\Delta_{ij})^2}}{\left(1 + e^{\beta(\Delta_{ij})^2}\right)^2}, \quad (9)$$

where  $\Delta_{ij}$  is calculated based on the old pixel values. At this point the lattice could be updated, but a gain annealing schedule has not yet been specified. The GDGA technique *automatically* selects an annealing schedule by using feedback from the total system energy to select the gain step size. The strategy involves varying the step in  $\beta$  in order to maintain constant steps in energy. The  $\beta$  step is given by

$$\Delta\beta = \frac{\Delta E}{\frac{\partial E}{\partial \beta}}. \quad (10)$$

In Equation 10 a constant energy step  $\Delta E$  is used,

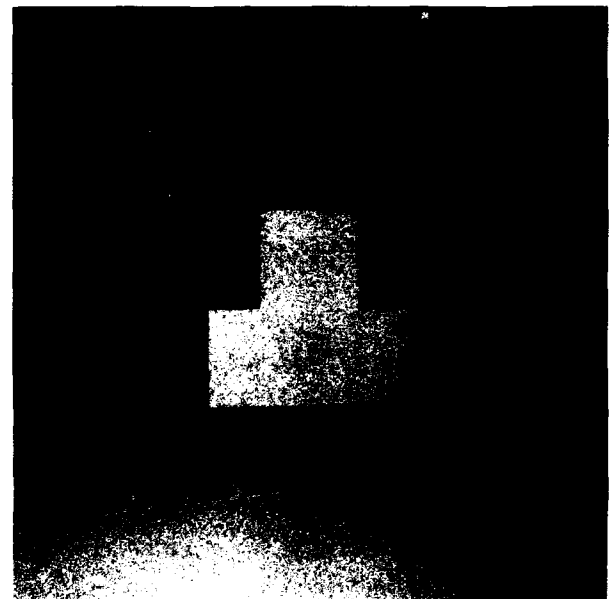
and the derivative is given by

$$\frac{\partial E}{\partial \beta} = \sum_{ij} (\Delta_{ij})^2 \frac{e^{\beta(\Delta_{ij})^2}}{\left(1 + e^{\beta(\Delta_{ij})^2}\right)}. \quad (11)$$

Note that in Equations 10 and 11 the gain term  $\beta$  refers to both the field and surround terms.

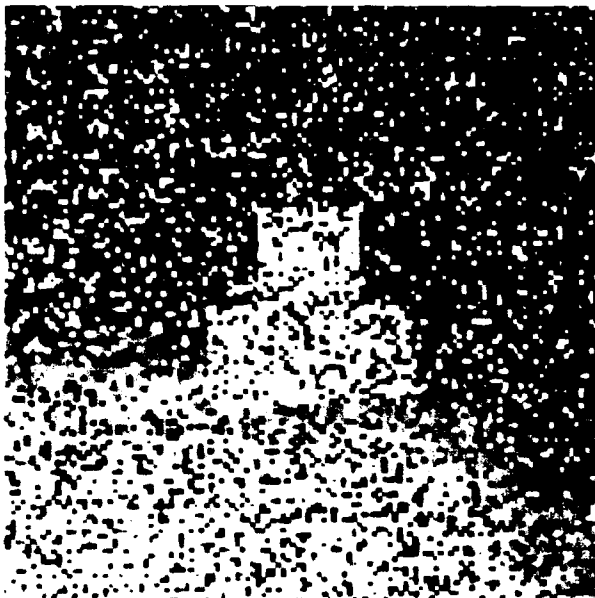
The GDGA technique starts at a small magnitude of gain and repeatedly applies Equation 8 until convergence, which typically requires fewer than 10 updates. Then Equation 10 is used to update the gain terms  $\beta^F$  and  $\beta^S$  and, with the new values, the pixels are again updated. The procedure is terminated when the magnitude of the gain becomes large—typically, a value of 10. For both simulated and real images, the GDGA algorithm is able to complete the restoration process (i.e., achieve sharp segmentation with noise removal) by using a total of 100 to 200 applications of the update equation (each application, or iteration, of Equation 8 updates all the pixels in the lattice). The restoration of a  $128 \times 128$  pixel, 8-bit image requires less than 5 min on a SUN-4 workstation.

Further experiments have revealed that the auto-



**FIGURE 2.** Original (noise free) image containing a linearly sloping background with a target at constant range.



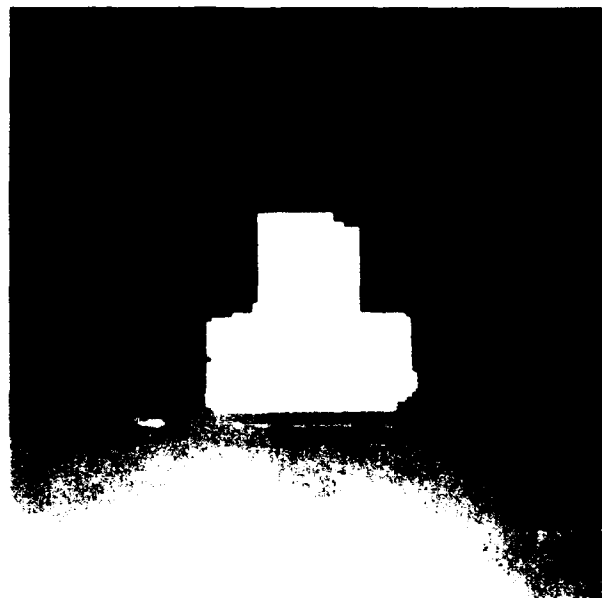


**FIGURE 3.** The image of Figure 2 has been synthesized for a carrier to noise ratio (CNR) of 10 dB, corresponding to an image in which 20% of the pixel values are anomalous. (An anomaly is defined as a pixel value in the corrupted image that differs from the value in the original image by more than two range counts.) Each pixel in the image has 8 bits (256 gray levels) of resolution.

matic gain annealing process takes small steps in  $\beta$  at small magnitudes of gain and toward the end of the annealing takes large steps as the magnitude of  $\beta$  becomes larger. Thus the GDGA method adaptively adjusts the step size to make efficient use of each iteration. The adaptive nature of the algorithm is especially evident when comparing restorations of low- and high noise images. Images with about 10% of the pixels corrupted with noise require fewer than 20 iterations for the entire restoration, while images with 70% noise require about 200 to 250 iterations.

### Simulation Results

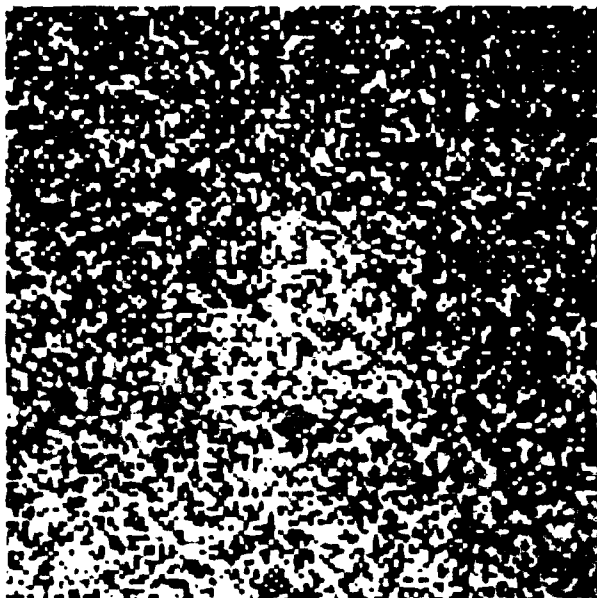
This section presents the qualitative and quantitative results obtained from applying the GDGA algorithm to the MRF image restoration model described earlier. The algorithm was tested on a synthetic range image that had been corrupted with noise by a range sensor measurement model described in the literature [3, 11]. The measurement model, which simulates



**FIGURE 4.** Results of MRF restoration on the synthetic range image of Figure 3. A four nearest neighbor MRF processor was used to restore the image. Note that the edges have been nearly perfectly preserved.

a peak detecting laser radar sensor that introduces anomalies into the measurement, was used to corrupt an image by relating the carrier to noise ratio (CNR) of the range sensor to the expected percent anomalies in the measurement. (An anomaly is defined as a pixel value in the corrupted image that differs from the value in the original image by more than two range counts.) In this work CNR values of 10 dB and 6 dB were used, corresponding to 20% and 71% anomalies, respectively, in the corrupted image. The measurement model does not assume a Gaussian distribution and is based on realistic sensor measurements.

The original (noise free) synthetic image shown in Figure 2 contains a simple shape at a constant pixel value against a background whose pixel values linearly increase from the top to the bottom of the image. In this work all of the input and restored images have 8 bits (256 gray levels) of resolution. Figure 3 shows the range image of Figure 2 after the image has been corrupted with 20% anomalies, and Figure 4 shows the result of the MRF restoration. Except for a few discrepancies at the boundary, the restoration is nearly perfect, especially in recovering the sloping background. Figure 5 shows the range image of Figure 2



**FIGURE 5.** The image of Figure 2 has been synthesized for a CNR of 6 dB, corresponding to an image that has 71% anomalies. Each pixel in the image has 8 bits (256 gray levels) of resolution.

with 71% anomalies, and Figure 6 shows the result of MRI restoration. Although the human visual system can barely recognize the original shape at this noise level, the MRI restoration is able to recover the underlying edges that define the target's shape, as shown in Figure 6. In both cases the MRF model produces a piecewise smooth restoration of the input image. The use of the sigmoid function facilitates noise reduction (smoothing) while preserving sharp discontinuities (edges).

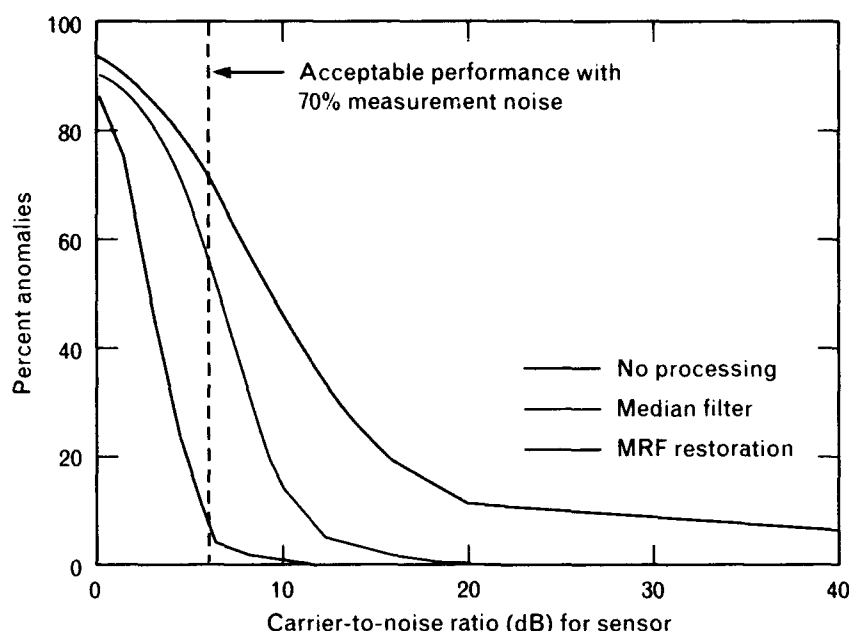
Next we describe a quantitative analysis of the model's performance based on the average percent of anomalies in the restored image. This measure is relevant to target detection in a noisy environment. Figure 7 shows a statistical comparison between corrupted images with and without restoration. In the figure, each point represents an average over 10 runs at a fixed noise level. The average percent anomalies in the image gives an indication of the difficulty of target detection; i.e., the probability of detection is lower at higher anomaly percentages. For effective detection, the percent of anomalies must be less than about 10%. Thus the results show that MRI restoration is effective up to a CNR of about 6 dB. At this CNR the average percent of anomalies



**FIGURE 6.** Results of MRF restoration on the synthetic range image of Figure 5. A four nearest neighbor MRF processor was used to restore the image. Note that the edges have been preserved.

is 71% for the input image, 55% for the median filtered image, and only 4.5% for the MRI restored image. Clearly the MRI restoration provides superior detection performance in a high noise environment.

We have also evaluated the utility of the MRI image restoration model as a preprocessor for target recognition in a noisy environment. Eight different binary silhouettes (broadside views of different vehicles (Figure 8)) were used in this experiment. Each of the silhouettes was centered in a  $128 \times 128$  pixel image to simulate range images, which were then corrupted with the range measurement model to produce images corresponding to realistic range measurements. Twelve such images were produced for each of the 8 silhouettes for a total of 96 images per each CNR value from 6 dB through 20 dB in 1 dB steps, and at 80 dB. Next, detection and segmentation were performed on these simulated range measurements to obtain noisy binary range slices in which only those pixels within a certain range are shown. The range slices were then used to train a Nearest Neighbor Classifier (NNC) [15] to separate the 8 different vehicles. We found that at very high CNR values (>10 dB) the classifier



**FIGURE 7.** Average percent anomalies for range imagery as a function of sensor CNR. The results from MRF restoration are compared with median filtering and the case in which no processing has been performed to restore the image. Each data point represents an average over 10 runs at a given noise level. For effective target detection, the percent of anomalies must be less than about 10%. Thus the results indicate that MRF restoration is effective up to a CNR of about 6 dB. At this CNR value the average percent of anomalies is 71% for the input image, 55% for the median-filtered image, and only 4.5% for the MRF-restored image. Clearly, MRF restoration provides superior detection performance in a high-noise environment.

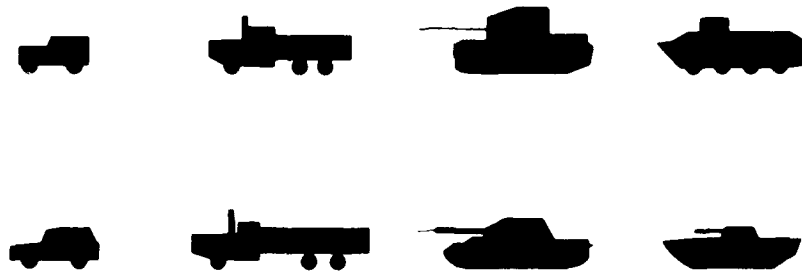
created a unique category for each silhouette. At lower CNR values (higher noise), however, the classifier formed extra categories because exemplars of the same target were sometimes classified into different categories. We repeated the above procedure twice: once using the median-filter technique on the corrupted images before the detection and segmentation steps, and once using MRF restoration. Figure 9 compares MRF restoration with iterated median filtering and with the case in which no processing had been performed to restore the image. The performance at each CNR value is defined as the fraction of the 96 examples that the NNC has classified correctly. Note that the MRF restoration is able to maintain an acceptable level of performance at a CNR that is 5 and 10 dB lower than with the median-filter and no-preprocessing case, respectively. Thus, with MRF restoration, a sensor can be operated at roughly 25% the power level required by

the use of a median filter.

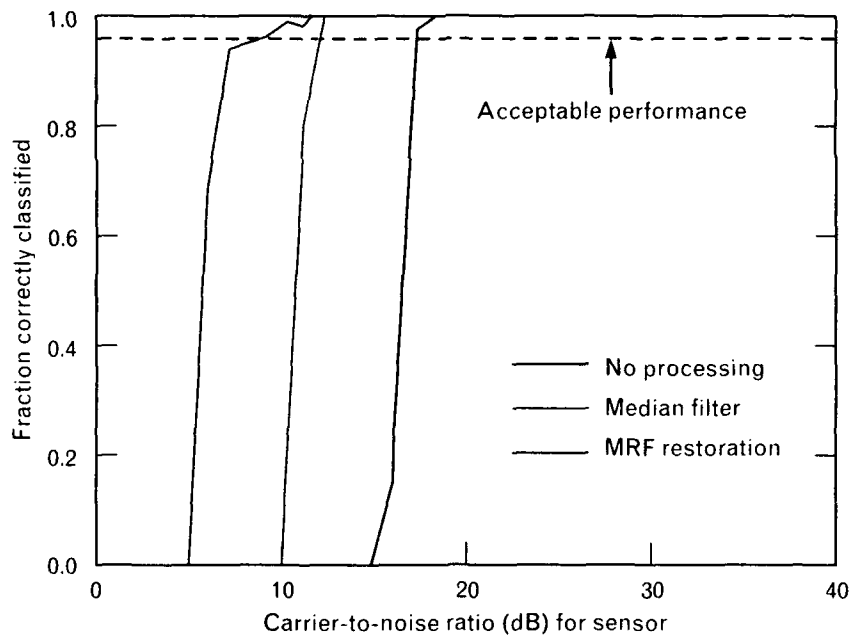
### Hardware Implementation

For real-time image restoration, the MRF model can be implemented either digitally—on custom digital signal processing (DSP) chips or on a single-instruction multiple-data (SIMD) computer such as the Connection machine manufactured by Thinking Machines Corp.—or in an analog manner on a custom VLSI chip.

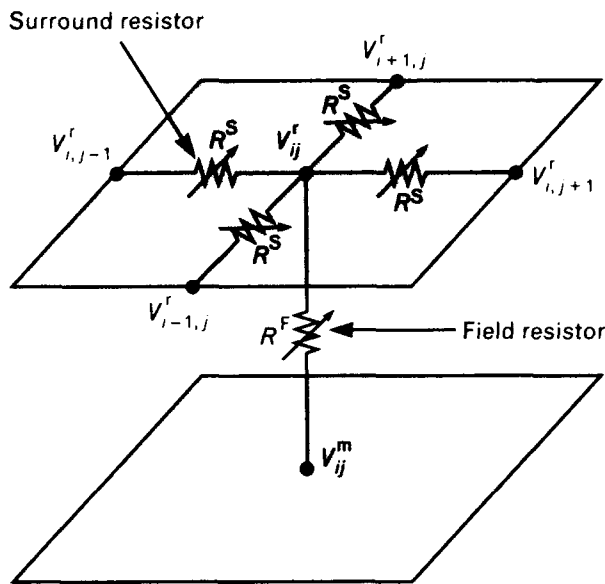
For digital implementation, 72 floating-point operations are required per pixel update. Typically, a pixel must be updated about 100 times over the course of the restoration. Thus a  $256 \times 256$  image restoration would require 472 million floating-point operations, and a frame-rate restoration of the same image would require digital hardware that delivers 14 GFLOPS of performance. This performance level is at the leading edge of current digital processing tech-



**FIGURE 8.** Binary silhouettes of 8 different vehicles that were used to evaluate the effect of MRF restoration on target recognition. (For the test results, see Figure 9.)



**FIGURE 9.** Fraction of correctly classified range slices as a function of sensor CNR. In the experiment, the binary silhouettes of 8 different vehicles (Figure 8) were used to create realistic noise-corrupted range measurements. Detection and segmentation were performed on these simulated range measurements to obtain noisy binary range slices in which only those pixels with a certain range are shown. A total of 12 such slices was produced for each of the 8 silhouettes at each CNR value. The 96 range slices at each CNR value were then used for training a Nearest Neighbor Classifier (NNC) [15] to separate the 8 different vehicles. Before being presented to the NNC, some of the range slices were restored by the MRF model and others by an iterated median filter. The results compare MRF restoration with the median-filter technique and with the case in which no processing has been performed.



**FIGURE 10.** System architecture implemented as a resistive grid (compare with Figure 1). The voltages  $V^m$  and  $V^r$  represent the measured and restored images, respectively. Note the presence of both field and surround resistors.

nology. The advantage of an all-digital implementation is that it does not require the hardwiring of any of the system parameters.

An all-analog implementation requires the implementation of the energy function as an analog circuit. In the current example the system architecture can be represented as a resistive grid (Figure 10). The resistors are nonlinear in that their resistances are voltage dependent:

$$R(\Delta_p) \propto \frac{(1 + e^{\beta(\Delta_p)^2})^2}{e^{\beta(\Delta_p)^2}},$$

where  $\Delta_p$  refers to the voltage difference between a pair of adjacent sites. The nonlinear resistor is essentially the inverse of Equation 9, and the circuit consists of separate field and surround resistors, as shown in Figure 10. At steady state there is a current balance at every site and the voltages  $V^r$  correspond to the intensities in the restored image. The advantage of this implementation is that there is essentially a "processor" at every site, and the processing speed is limited only by the settling time of the analog circuit.

The throughput is limited by the input/output onto and off the analog chip, and not the circuit. With current technology, images can be restored at a rate in the thousands of frames per second.

### Summary

An efficient Markov Random Field (MRF) based method for performing piecewise smooth image restorations has been demonstrated. The underlying model uses a neural network sigmoid potential between pixel pairs to allow the formation of sharp boundaries between dissimilar regions in the presence of noise. A novel deterministic method—called Gradient Descent Gain Annealing (GDGA)—for solving the nonlinear coupled set of differential equations that the MRF model introduces was presented. The GDGA algorithm typically requires fewer than 200 iterations to restore an image, where the number of iterations is roughly proportional to the level of noise in the image. Computer simulations on noisy images have shown that restorations can be performed for very high noise levels (i.e., images that have up to 71% of their pixels corrupted with non-Gaussian sensor noise). Simulation results indicate that MRF restoration provides a 5-dB advantage in the carrier-to-noise ratio (CNR) over conventional iterated median filtering. Although the same model is currently used to restore images from different sensors, arbitrary potentials can be incorporated for the pixel interactions so that the system can be tailored to specific natural scenes and sensors. The system uses a massively parallel set of local neighborhoods (four nearest neighboring pixels) for efficient implementation on a parallel-processing computer or a custom analog VLSI chip.

### Acknowledgments

The author wishes to thank members of the Opto-Radar Systems Group at Lincoln Laboratory for very helpful technical discussions and computer support throughout the course of this research. In particular, the author is indebted to William M. Wells III for many helpful discussions and suggestions, and for help in reviewing the existing image-restoration literature.

This work was sponsored by the Defense Advanced

Research Projects Agency.

## REFERENCES

1. G.L. Bilbro and W.E. Snyder, "Range Image Restoration Using Mean Field Annealing," in *Advances in Neural Information Processing Systems I*, D.S. Touretzky, ed. (Morgan Kaufmann, San Mateo, CA, 1989), pp. 594-601.
2. D. Geiger and F. Girosi, "Parallel and Deterministic Algorithms for MRF's: Surface Reconstruction and Integration," *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 401 (1991).
3. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 721 (1984).
4. D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 2, D.E. Rumelhart and J.L. McClelland, eds. (MIT Press, Cambridge, MA, 1986), p. 422.
5. E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines* (John Wiley, NY, 1989), p. 130.
6. S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi, "Optimization by Simulated Annealing," *Science* 220, 671 (1983).
7. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.* 21, 1087 (1953).
8. J.L. Marroquin, "Probabilistic Solution of Inverse Problems," Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1985.
9. J. Leinbach, "Automatic Local Annealing," in *Advances in Neural Information Processing Systems I*, D.S. Touretzky, ed. (Morgan Kaufmann, San Mateo, CA, 1989), pp. 602-609.
10. A. Blake and A. Zisserman, *Visual Reconstruction* (MIT Press, Cambridge, MA, 1987), p. 131.
11. Y.G. LeClerc, "Constructing Simple Stable Descriptions for Image Partitioning," *Intl. J. Comput. Vision* 3, 73 (May 1990).
12. S.D. Conte and C. de Boor, *Elementary Numerical Analysis: An Algorithmic Approach*, 3rd ed. (McGraw-Hill, NY, 1980), p. 226.
13. M.B. Mark, "Multipixel, Multidimensional Laser Radar System Performance," Ph.D. Thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1986.
14. A.B. Gschwendtner, R.C. Harvey, and R.J. Hull, "Coherent IR Laser Technology," *Optical and Laser Remote Sensing*, D.K. Killinger and A. Mooradian, eds. (Springer-Verlag, NY, 1983), p. 327.
15. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis* (John Wiley, NY, 1973).



MURALI M. MENON is currently a research staff member in the Opto-Radar Systems Group. He received a B.S., an M.S., and a Ph.D. degree in chemical engineering from Case Western Reserve University in Cleveland. His research interests include applied pattern recognition, signal processing, and image processing, with special interests in wavelets and artificial neural networks. He has spent the past six years at Lincoln Laboratory working on applications of neural networks for processing sensor data, including the design of automatic target recognition (ATR) systems.

# Machine Intelligent Automatic Recognition of Critical Mobile Targets in Laser Radar Imagery

Richard L. Delanoy, Jacques G. Verly, and Dan E. Dudgeon

■ A variety of machine intelligence (MI) techniques have been developed at Lincoln Laboratory to increase the performance reliability of automatic target recognition (ATR) systems. Useful for recognizing targets that are only marginally visible (due to sensor limitations or to the intentional concealment of the targets), these MI techniques have become integral parts of the Experimental Target Recognition System (XTRS)—a general-purpose system for model-based ATR. Using laser radar images collected by an airborne sensor, the prototype system recognized a variety of semi-trailer trucks with high reliability, even though the trucks were deployed in high-clutter environments.

THE CONSTRUCTION of an automatic target recognition (ATR) system is a demanding task. ATR systems must be able to locate and identify specific targets that can be concealed intentionally through obscuration or camouflage, that are often designed to be nearly invisible in radar imagery, and that can be deployed in the midst of distracting signals. To gain tactical advantage, it is generally important that an ATR system be able to find a target from as far away as possible. Under such conditions, the selectively indicative signal features (signatures) associated with a target are often barely discernible from the background. Thus, inevitably, practical ATR systems must be able to discriminate targets from background in spite of weak, ambiguous, uncertain, variable, or even contradictory evidence.

ATR system development can be particularly difficult under certain mission constraints and when the costs of system error are high. One such ATR application is the use of airborne sensors to recognize strategic relocatable targets (SRT) such as the SS-25 ICBM

of the former Soviet Union (Figure 1 [left]). An ATR system for recognizing SRTs must search through images generated by one or more sensors (laser radars, real- or synthetic-aperture radars, passive infrared imagers, and video cameras), requiring techniques of data fusion. The search is for a very small number of targets in a continent-sized area. The targets might be caught in the open, but more likely will be found along tree lines, perhaps partially occluded by foliage. Because of the nature of the targets, a high probability of detection is crucial. And yet the ATR system must generate few false alarms (FA) due to mission limits on the number of weapons that an aircraft can carry to destroy the SRTs, the flying time of the aircraft over the target area, and the processing capabilities of human operators who must decide which detections to pursue. A closely related mission is the detection of Scud launchers, such as those used by Iraq in the Persian Gulf War. A general term, critical mobile target (CMT), refers to all mobile missile launchers, including those used with SS-25 and Scud missiles.





**FIGURE 1.** Photographs of (left) mobile missile launcher carrying a strategic SS-25 ICBM of the former Soviet Union, as shown in a Soviet newspaper, and (right) tank truck used by Lincoln Laboratory as a substitute vehicle to develop and test an automatic target recognition (ATR) system for detecting critical mobile targets (CMT) such as the SS-25 launcher.

To achieve reliable detection and recognition performance in such demanding applications, we have developed several new machine intelligence (MI) techniques, including new approaches for model-based classification [1–3], automatic learning of models [4], knowledge-based signal processing [5, 6], selective attention [7], and pixel-level data fusion [7]. The Experimental Target Recognition System (XTRS) developed at Lincoln Laboratory [8–10] provides a framework for the application of these techniques and for the rapid prototyping of ATR systems. Though XTRS and these new MI techniques were intended specifically for ATR, they constitute a general-purpose approach to object recognition, with many potential applications. For example, XTRS has been applied successfully to the detection and tracking of hazardous weather phenomena, as described in the article "Machine Intelligent Gust Front Detection" by Richard L. Delaney and Seth W. Troxel in this issue [11]. In the current article, we apply XTRS to the detection and recognition of CMTs, specifically, tank trucks (Figure 1 [right]) and logging trucks used as substitutes for missile launchers.

### Low-Level Machine Intelligence

Computer vision systems have traditionally been designed in terms of a hierarchy of levels. Low-level

vision works on a domain of pixel-level data. Because the associated image processing operations are highly repetitive and therefore relatively slow, low-level operations tend to be kept simple. And, because a scene can contain many objects of potential interest, low-level operations tend to be generic and relatively devoid of object-dependent knowledge. A typical low-level operation is edge detection. In high-level vision, the pixel-level data (for the edge-detection example, a pair of images showing the strength and orientation of edges) are transformed into symbolically described features. Object identification is then performed by matching these features against prior knowledge of object characteristics.

This basic organization of computer vision has often been used in the design of ATR systems. In the detection process, which is analogous to low-level vision, a threshold is applied to a set of signals. The signals can come directly from a sensor or they can be the result of a signal processing operation. For the threshold to be effective, the signals associated with targets must form a distribution that is distinguishable from the distribution of signals associated with other objects in the background (i.e., clutter). Figure 2(a) illustrates this point. In any realistic detection problem, there will exist some targets that have signals below the detection threshold; those targets will

not be detected. There will also exist instances of clutter associated with signals that are above the detection threshold; such instances will result in FAs.

Developers of ATR systems have generally followed the strategy of keeping low-level vision devoid of object-dependent knowledge, and the processing done in preparation for the application of thresholds is usually kept simple. As a result, the thresholds must be set fairly low to maximize the likelihood of detecting targets. The high-level recognition process is then responsible for suppressing as many FAs as possible. Usually, FA suppression is accomplished through the use of classifiers based on statistical techniques or MI techniques, including those involving expert systems, model-based matching, and neural networks. However, when a given ATR application involves only one or at most a few intended targets or classes of targets, the use of object-dependent knowledge for the detection process is feasible. And, in fact, because targets are often hidden, camouflaged, or otherwise only marginally visible, object-dependent knowledge can play an important role in enhancing detectability. In particular, detection performance can be improved by separating the distributions of targets and clutter, as shown in Figure 2(b). Although various techniques have been developed to increase this separation (see, for example, Reference 12), the techniques do not typically involve any detailed object-dependent knowledge.

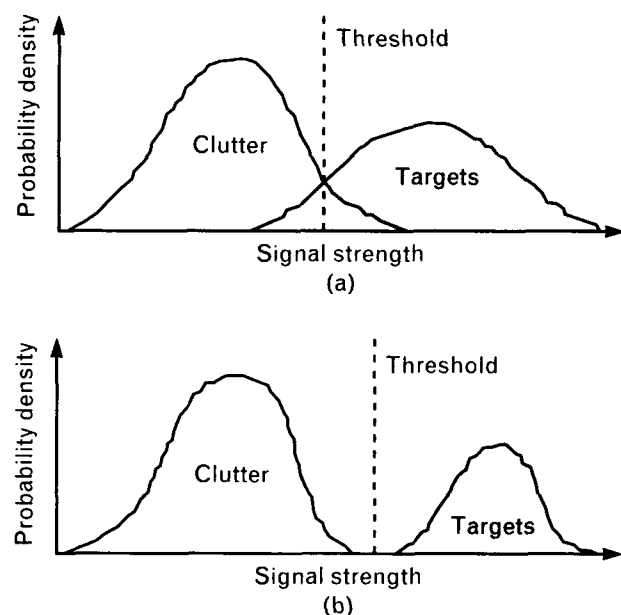
Thus, in addition to using MI techniques in the conventional role of high-level classification and FA suppression, we have developed a set of new techniques for low-level MI. Thresholds are still an unavoidable part of detection but, when low-level MI is applied directly to the pixel-level data early in the detection process, the use of thresholds is in a relative sense postponed and, as a result, made more effective.

### *Interest Images*

A key to implementing low-level MI in XTRS is the concept of *interest* and *interest image* [7]. By our definition, *interest* is a dimensionless quantity indicating the likelihood that a specific feature, indicative of a target or class of targets, is present at a given image pixel. A spatial map of such interest values, each constrained to the range [0,1], constitutes an *interest*

*image*. Clusters of high interest values are used as a guide to focus computational resources on likely targets. In Figure 2(a), a threshold was applied to the quantity "signal strength." For simple ATR systems, the signal is typically the intensity of returned electromagnetic energy or a simple function thereof. Interest provides an alternative flexible metric to which thresholds can also be applied. The power of this approach is that the output of any sensor modality or feature detector can arguably be expressed as an interest image. Furthermore, the use of interest as a common denominator greatly simplifies the fusion of pixel-level data. Specifically, interest enables the use of simple arithmetic or fuzzy logic to fuse spatial evidence from a variety of sources.

There are three steps in low-level MI as used by XTRS in the detection process. First, relevant feature detectors are selected, given knowledge of the situational context. The context may include the intended set of targets, various sensor-related parameters, and identifiable environmental conditions affecting sensor performance and target appearance. Often the use



**FIGURE 2.** Discrimination of target and clutter signals through the application of a threshold: (a) typical overlap of targets and clutter distributions, and (b) illustration of how signal processing, either conventional or machine intelligence (MI) based, can improve detection performance by increasing the separation of the distributions.

of just one feature detector can accomplish adequate target-detection performance.

In the second step, the selected feature detectors are applied to the appropriately prepared input imagery. Each detector generates as its output an interest image that provides spatial evidence for the presence of particular target features. A targeted object may be represented by more than one detector; each detector looks, for example, for a distinct set of features or for an alternative target configuration.

The final step calls for the fusion of evidence, which is accomplished with a rule of combination prescribing how interest values from multiple interest images are to be combined. The rule of combination depends on the set of feature detectors selected. In the case of multiple feature detectors looking for alternative target configurations, the rule of combination could be the maximum (*fuzzy-or* in fuzzy set theory); i.e., at a specific pixel location, the maximum of the interest values across all interest images at that location could be used. In the case of several feature detectors looking for different vehicle features that are likely to be present all the time, the fusion of interest values might be done by an averaging process. Although not fully exercised in the CMT version of XTRS, the rule of combination could be arbitrarily complex to reflect knowledge of the variable reliability of different feature detectors under different viewing conditions.

For situations in which only targets return strong intensity signals, the intensity signal returns might provide a ready-made interest image. In practice, however, laser intensity can be an unreliable discriminant because the energy returned from a target surface depends on the specularity of the surface and its orientation relative to the incident laser beam. Also, high (or low) range values from a laser radar are usually unreliable predictors of target locations because targets are not customarily parked at the highest (or lowest) points in a locality. Thus the effective use of laser radar imagery requires that objects in the imagery be identified also on the basis of shape.

#### *Functional Template Correlation*

In studying the principal techniques for shape analysis, we found that the basic equations of cross-correla-

tion and mathematical morphology (MM) [13] can be generalized into a single class of operations, which we have called *functional shape matching*. Briefly described, these shape-analysis tools all use kernels (structuring elements in MM), which are basically subimages that are looked for within the image to be probed. For probing a pixel in an image, the origin of a particular kernel is positioned over the pixel's location. A two-argument function is then applied to each kernel value and the corresponding value in the image. (For cross-correlation, the two-argument function is multiplication. For MM dilation and erosion, the functions are addition and subtraction, respectively.) Next, an arbitrary operator is applied to the function values obtained for the set of pixel locations on the kernel. (For cross-correlation, the operator is summation. For MM dilation and erosion, the operators are maximum and minimum, respectively.)

Eventually, we realized that functional shape matching not only includes the classic shape-analysis tools, but it also encompasses a variety of signal processing techniques that have never been tried before. From functional shape matching, we implemented a tool for generalized matched filtering called *functional template correlation* (FTC) [5]. Whereas the kernel of the classic techniques is a subimage indicating specific expectations of image values for a successful match, the kernel used in FTC is a set of indexes, each corresponding to a unique scoring function. Each of these scoring functions can define arbitrary expectations for image values at each pixel location on the kernel. The outputs of these scoring functions are scalar values, which are averaged and then "clipped" to the range [0,1]. (In the clipping process, those averaged scores which are less than zero are assigned a value of zero, while those averaged scores which are greater than one are assigned a value of one.) Comparable in spirit to the membership functions of fuzzy set theory, scoring functions provide a means of encoding uncertainties. But, in addition, scoring functions can be used to encode a surprising amount of knowledge of the physics of a matching problem. Using FTC, we can construct customized matching techniques that are more powerful than the classic shape-analysis operations. (Note: For a brief introduction to FTC, see the box, "Functional Template

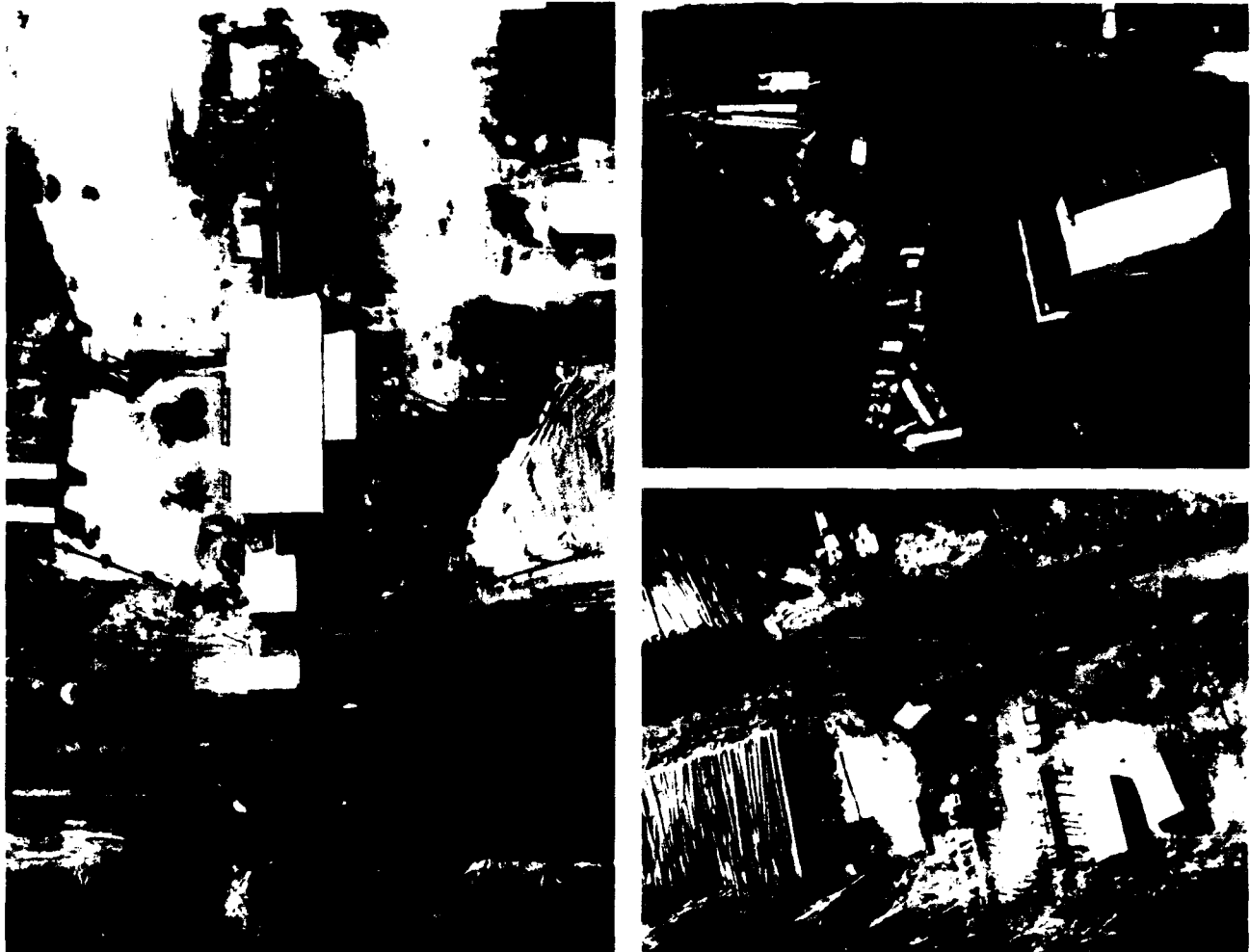
Correlation," in the article "Machine Intelligent Gust Front Detection," by Delanoy and Troxel in this issue [11].)

### Data Used in System Development and Testing

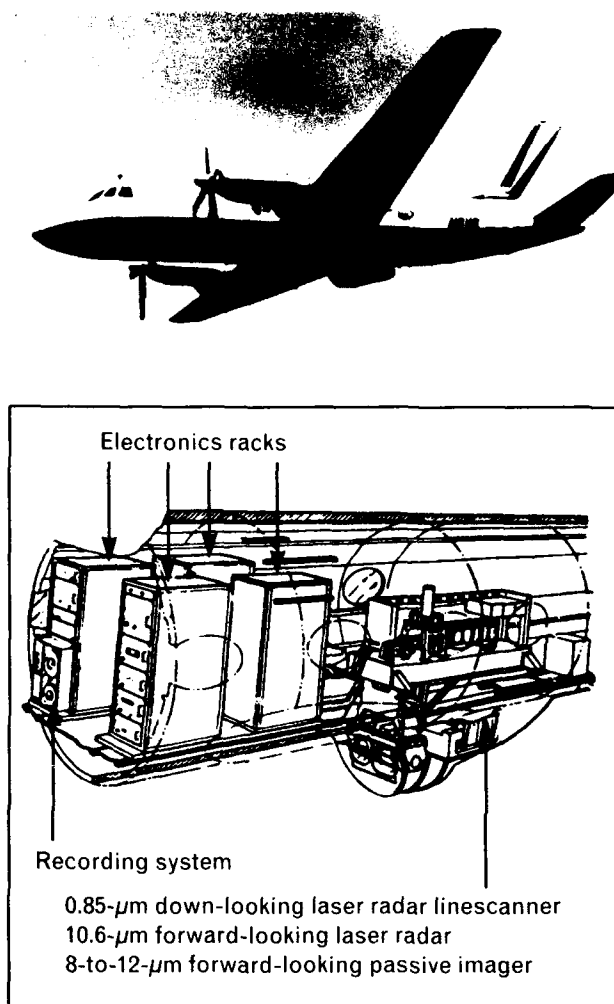
For the development and testing of a CMT version of XTRS, a large dataset of images in Maine was collected to simulate the detection of CMTs in a high-clutter, mainly forested environment. Semi-trailer trucks, which approximate the appearance of missile launchers, were positioned amidst both natural and man-made clutter. The simulated targets included the tank truck shown in Figure 1 (right), the same tank truck but under camouflage netting, a loaded logging

truck, and an empty logging truck. This variety of vehicles was used to test XTRS's ability to discriminate between targets of similar shapes and sizes. The vehicles were positioned on or near roads, both in the open and along tree lines. The man-made, or cultural, clutter included residential neighborhoods and a logging camp (Figure 3) that contained heavy logging equipment such as other semi-trailer trucks.

Pixel-registered range and intensity images of the various vehicles were generated with the Hughes-Danbury GaAs Laser Linescanner carried aboard a Gulfstream G-1 aircraft (Figure 4). Characterized by a  $0.85\text{-}\mu\text{m}$  wavelength, the linescanner has a range ambiguity of 10 m, a precision of ambiguous-range



**FIGURE 3.** Aerial photographs of man-made, or cultural, clutter represented in the laser-radar-image dataset. Contained in the photographs are railroad cars, fuel tanks, stacks of logs, empty logging trucks, other road vehicles, and heavy logging machinery. All of these objects, easily confused with the set of targets being sought, are potential sources of false alarms (FA).



**FIGURE 4.** Lincoln Laboratory airborne sensor platform. The Gulfstream G-1 carries a 0.85- $\mu\text{m}$  down-looking laser radar linescanner, a 10.6- $\mu\text{m}$  forward-looking laser radar, and an 8-to-12- $\mu\text{m}$  forward-looking passive imager. The images used in the experiments described in this article were collected with the 0.85- $\mu\text{m}$  linescanner.

values of 15 cm, and an angular resolution of 1.0 mrad. Images were collected during the winter and summer of 1989 with a down-looking sensor platform that was operated at altitudes between 200 and 300 m by the Opto-Radar Systems Group of Lincoln Laboratory. The example range and intensity images shown in Figure 5 reveal the high resolution achieved with the 0.85- $\mu\text{m}$  linescanner. Image widths were between 64 and 150 m; image lengths could be arbitrarily long because of the linescanner used. Long scans were subdivided into overlapping images ranging in length from 100 to 400 m. In total, the dataset

collected contained 2303 image pairs (range and intensity) covering 17.13  $\text{km}^2$  of ground area under both winter (snow) and summer (dense foliage) conditions.

### System Description

The architecture of the CMT version of XTRS consists of five modules (Figure 6): preprocessing, detection, extraction, decomposition, and matching. Each module has a standard structure (Figure 7) that consists of four main elements: (1) a parameter library—a collection of algorithms, numbers, and/or data structures that encode knowledge relevant to the current stage of processing; (2) a parameter selector—a rule-based expert, i.e., a collection of rules, that uses contextual information and previous results to choose parameters from the parameter library; (3) a generic processing engine; and (4) a rule-based feedback expert that evaluates the output of the processing engine and decides where control should be directed. In the complete system, the feedback expert of one module and the parameter selector of the subsequent module conceptually form a local-control node.

### Preprocessing

The Hughes-Danbury GaAs Laser Linescanner produces pixel-registered range and intensity images that, in preparation for the detection process, require a number of data transformations.

First, because an aircraft's speed with respect to the ground varies depending on the wind velocity vector, the aspect ratios of targets that have been imaged by the linescanner can often become distorted. In an operational system, these distortions can be avoided by using an inertial navigation system either to regulate the linescan rate or to provide data that would allow the images to be corrected by interpolation. For the data used in this article, the interpolation was performed interactively to obtain the correct target aspect ratios.

Second, the ambiguous-range values are converted to absolute altitudes above some arbitrary reference altitude. Once the absolute altitudes have been determined, a map of altitudes for the local ground level can be computed with a technique based on morphological operations [14–16]. In the technique, only

those surface shapes which are wider than the intended targets are retained as part of the local ground level. (Note: There exist other techniques for estimating the local ground level; see, for example, Reference 17.) When the altitudes of the local ground level are subtracted from the absolute altitudes, the resulting image will contain values that are the heights of small objects (including targets) above the local ground level. In this article, subsequent uses of the term *range image* will refer to this image of heights above the local ground level.

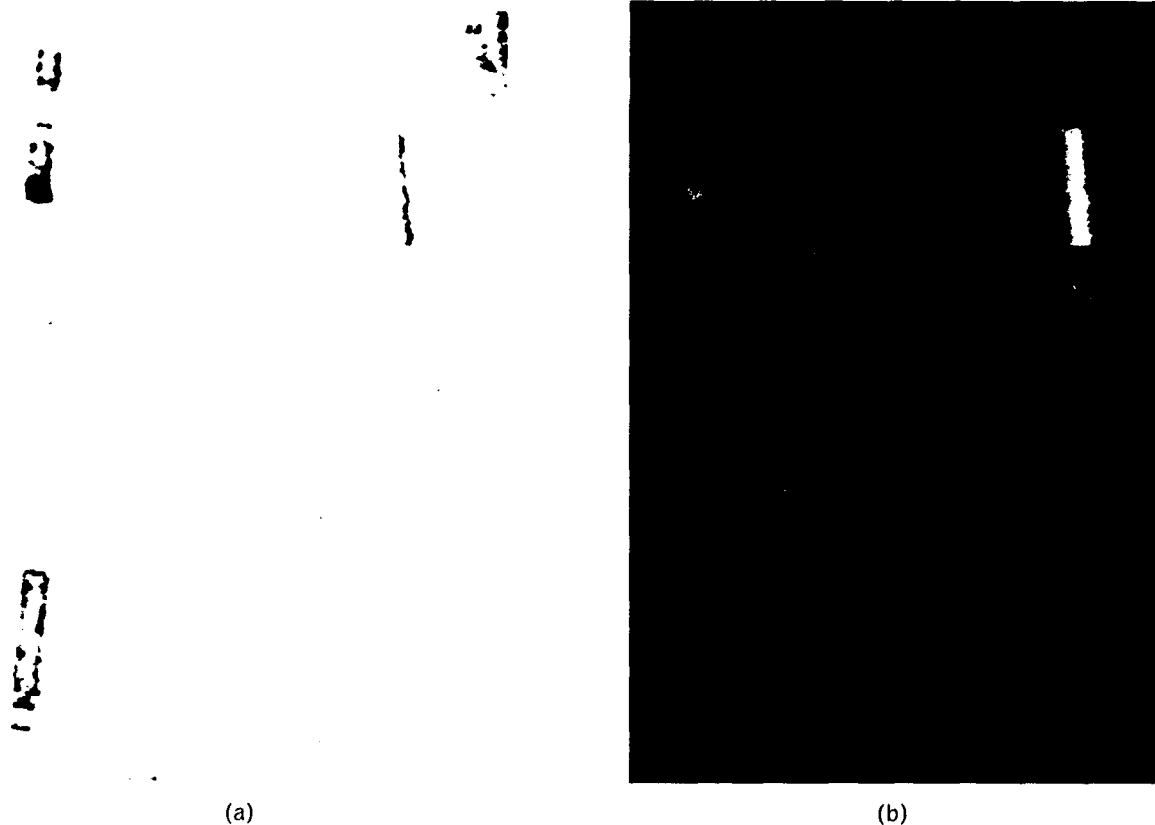
For the last step of preparation, the range and intensity images are scaled by linear interpolation to a resolution of 0.25 m per pixel side. Images of lower resolution (0.5 and 1.0 m per pixel side) are then generated by a subsampling of the data. The lower-resolution images are used for detection, while the high-resolution images are used for extraction and

high-level matching.

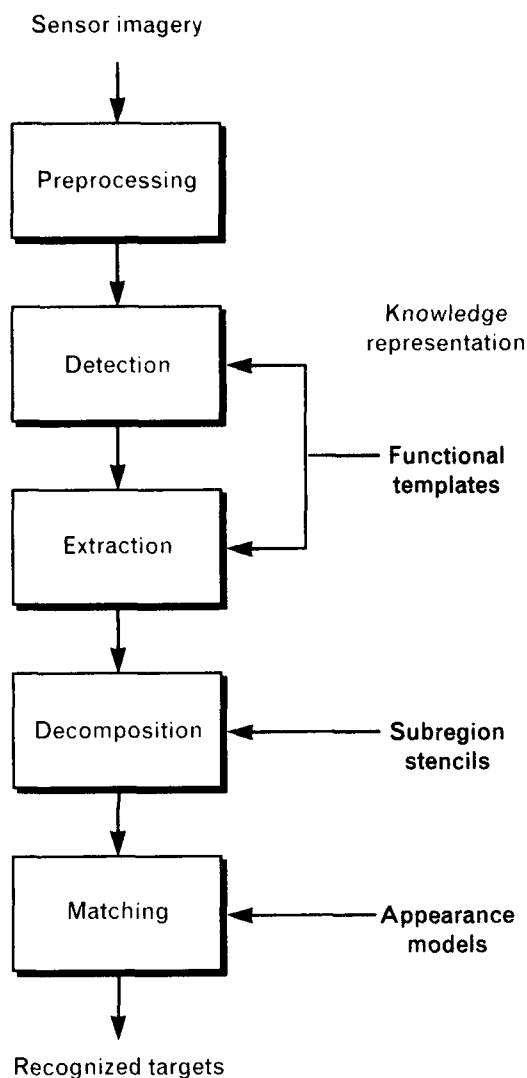
(Note: During preprocessing, no attempt was made to clean the linescanner imagery of noise. Such a procedure was unnecessary due, in part, to both the high quality of the imagery and the noise-resistant properties of FTC.)

### Detection

In the CMT version of XTRS, three-dimensional detection is essentially performed by four target detectors (i.e., feature detectors representing whole targets instead of individual features). The tank truck is represented by two alternative target detectors, one in which the truck is exposed, the other in which the truck is covered with camouflage netting. The logging truck is similarly represented by two target detectors, one in which the truck is empty, the other in which the truck is loaded with logs.



**FIGURE 5.** Example (a) intensity and (b) range images taken during winter with the 0.85- $\mu$ m down-looking laser linescanner shown in Figure 4. Note the tank truck (lower left), empty logging truck (upper left), and house trailer (upper right) in both images. The range image has been transformed such that each pixel value represents a height above an arbitrary reference altitude with lighter pixels indicating a greater height.



**FIGURE 6.** Architecture of Experimental Target Recognition System (XTRS), showing the five processing modules as well as the way in which knowledge is represented at each level.

The target detector for the exposed tank truck consists of the two functional templates shown in Figure 8. The first functional template encodes the expected appearance of the truck in range imagery (i.e., images of heights above the local ground level). In scoring function 1, which corresponds to the top surfaces of the cab and trailer, a maximal score of 1.0 is returned for heights from 2.5 to 3.5 m. The uncertainty comes from signal noise and inaccuracies in estimating the local ground level. The negative scores reflect the fact that tank trucks are opaque to laser illumination; i.e., the presence of ground-level heights

where the cab or trailer is expected constitutes strong evidence that a target is not present at that location. On the other hand, heights greater than the expected interval of 2.5 to 3.5 m result in scores no less than 0.5, the level of ambiguity, because such heights could potentially indicate the presence of an occluding surface. In other words, the cab of a tank truck might or might not be present under an occluding surface that is at least 4.0 m high.

The other scoring functions work in the same manner, except that the expected interval of heights for the background in scoring function 0 is from 0.0 to 0.5 m, and the expected interval for the hitch area in scoring function 2 is from around 1.0 to 2.0 m. These scoring functions are tuned such that, when the template is applied to a patch of bare ground (zero height), the negative scores from scoring function 1 balance the positive scores generated by scoring functions 0 and 2, resulting in an overall score near 0.0. And, of course, an unobscured target should generate a score near 1.0.

The above functional-template design provides a simple means of minimizing the effects of occlusion. This capability, not easily obtained with cross-correlation or MM operations, is necessary for finding targets partially covered by foliage. In the extreme, a target that is completely occluded should generate a score of 0.5 (Figure 9).

In Figure 8, the second functional template for the exposed tank truck is designed for intensity imagery. Because the surface of the truck is smooth (specular) with regard to the laser wavelength, the reflected laser beam will tend either to miss the sensor (resulting in a low intensity value) or hit the sensor directly (resulting in a high value). Scoring function 4, corresponding to the trailer and cab, encodes these expectations by returning high scores for very low and very high intensity values, and low scores for intermediate intensity values. The hitch area returns the laser energy more diffusely; thus, scoring function 5, which corresponds to the hitch area, returns high scores for intermediate intensity values. The intensity values associated with surrounding ground areas are highly variable and unpredictable, except that they are seldom very low. Scoring function 3 encodes this expectation with highly negative scores for low intensity values, but *nil*

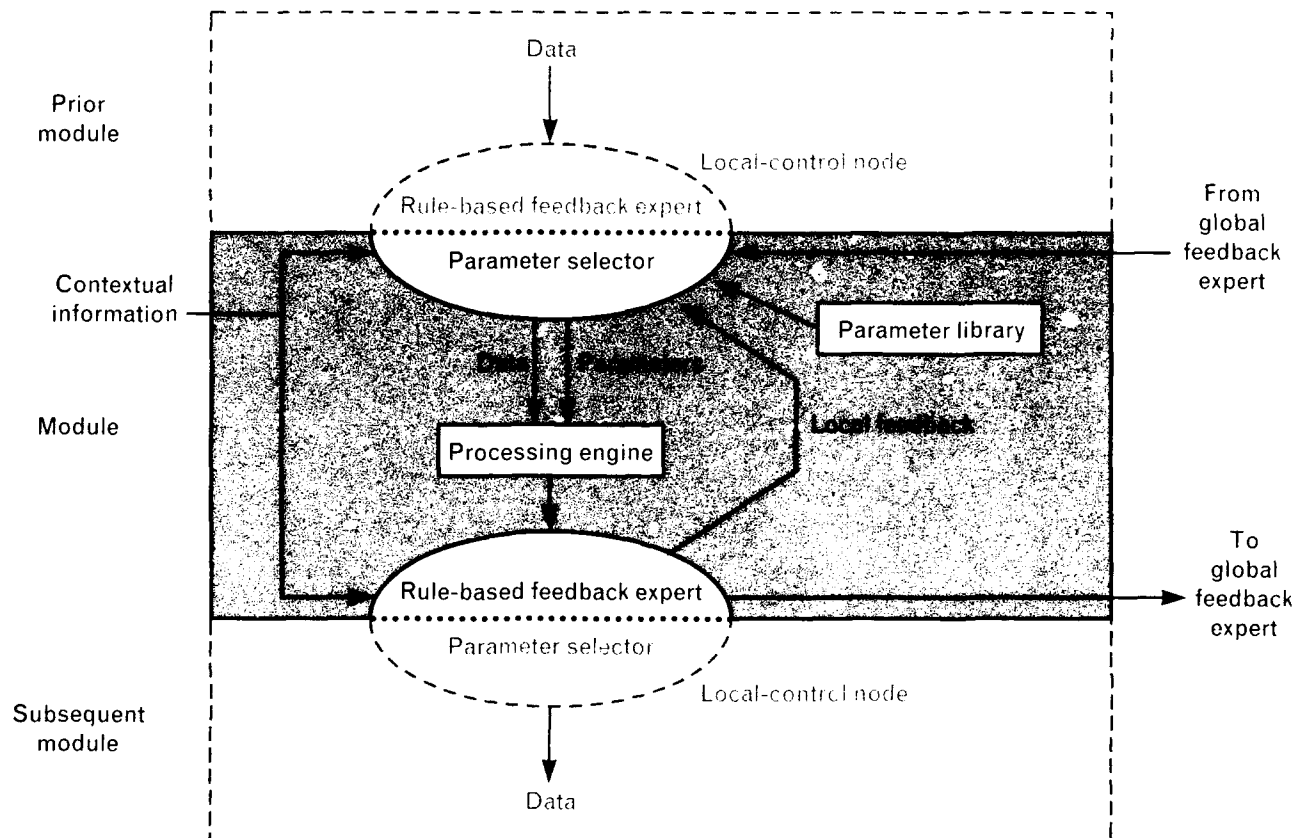
(i.e., no opinion) for intermediate and high intensity values.

The two functional templates shown in Figure 8 were applied simultaneously to the input range and intensity images, and overall scores were computed as the average of the scores returned from the six scoring functions. Because the orientations of targets are typically arbitrary and unknown *a priori*, an FTC score was computed for each of 36 uniformly spaced template rotations ( $10^\circ$  increments) at each pixel location of the input imagery. For a particular pixel location in an input image, the score associated with the maximally scoring orientation was assigned to the corresponding pixel location in the output image. As previously indicated, each such output image is treated as an interest image, indicative of the likelihood of finding a target at any particular pixel.

In a similar way, output interest images were also generated for each of the three other target detectors, and the four interest images were combined by taking

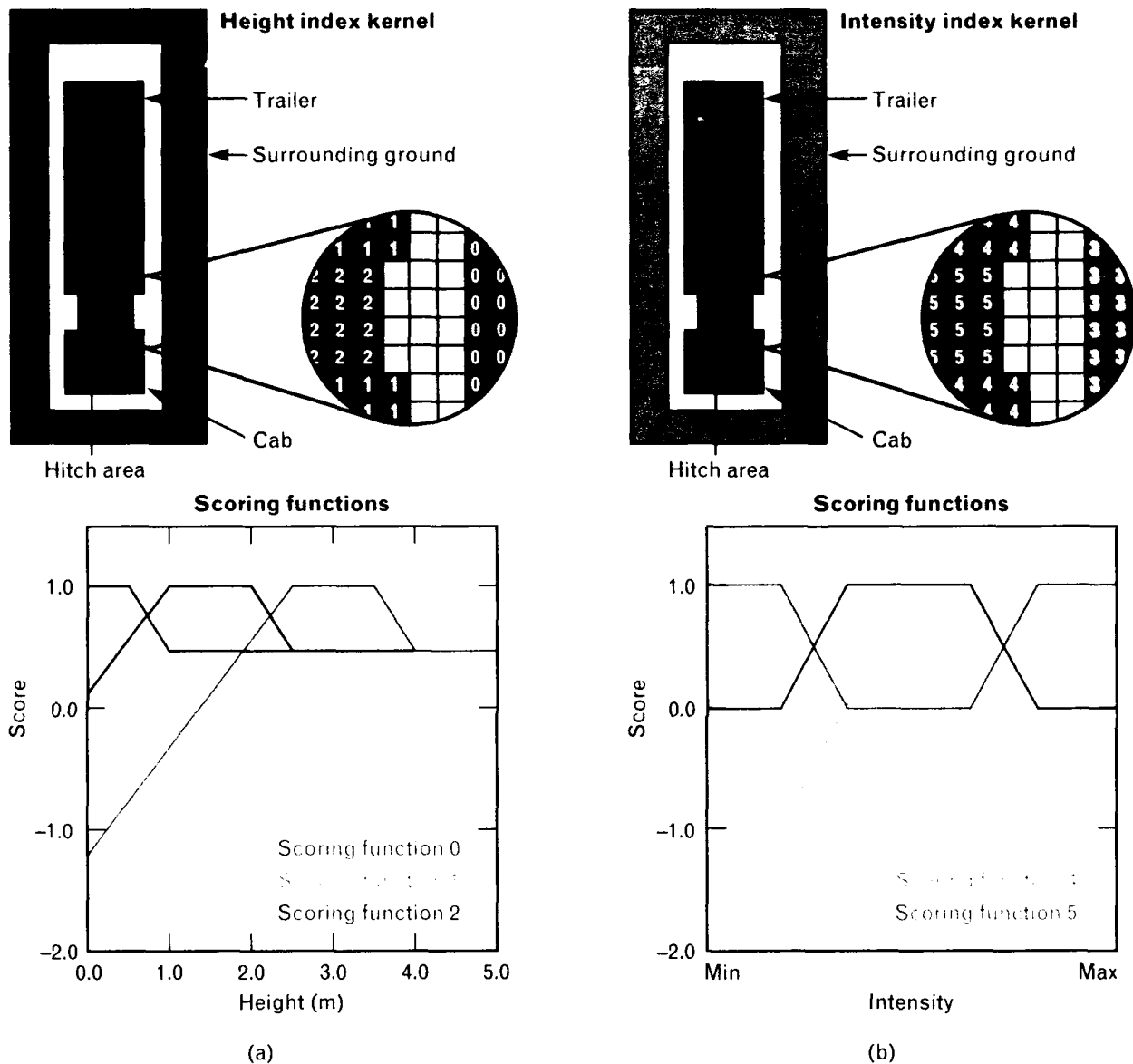
the maximal scores at each pixel location. The resulting combined interest image was then scanned for pixels having interest scores above a certain threshold (typically 0.75), and the above-threshold pixels were grouped into clusters.

Next, a box was placed around each cluster. The boxes were used to extract range and intensity subimages containing the interest cluster and thus the candidate target. The boxes were square, with sides 50% longer than the longest dimension of the targets being sought. Up to four boxes with above-threshold interest scores and a minimum of overlap with each other were constructed for each image. The cluster of above-threshold interest scores that led to the creation of a particular box was used to create a list of target hypotheses. At each pixel in a cluster, the interest score is always associated with the highest-scoring target detector at the highest-scoring orientation. Each pixel's hypothesis consisted of the highest-scoring target detector's name, pixel coordinates, orientation,



**FIGURE 7.** Standard structure of the XTRS processing modules shown in Figure 6. Note that the feedback expert of one module and the parameter selector of the subsequent module conceptually form a local-control node.





**FIGURE 8.** Functional template for the top view of the tank truck. Shown are the index kernels and indexed scoring functions for (a) range and (b) intensity images. Note that scoring function 3, i.e., the scoring function for the surrounding ground in intensity imagery, has no opinion above a certain intensity value.

and interest score, which was used to rank the hypotheses. For each box generated, information regarding the size and location of the box as well as hypotheses about what might be in the box was placed in a data structure referred to as a "window." Because the CMT version of XTRS uses the function maximum for the rule of combination of interest scores, the score achieved at any pixel location by the highest-scoring target detector is also the value stored at that same location in the window.

#### Extraction and Decomposition

Windows generated by the detection process are used as input to extraction. The position and size of each window are used to extract full resolution (0.25 m per pixel side) subimages of range, intensity, and interest. In the extraction module (Figures 6 and 7), the parameter selector chooses from the library an extractor corresponding to the highest-ranking hypothesis in a window. In the current implementation of our sys-

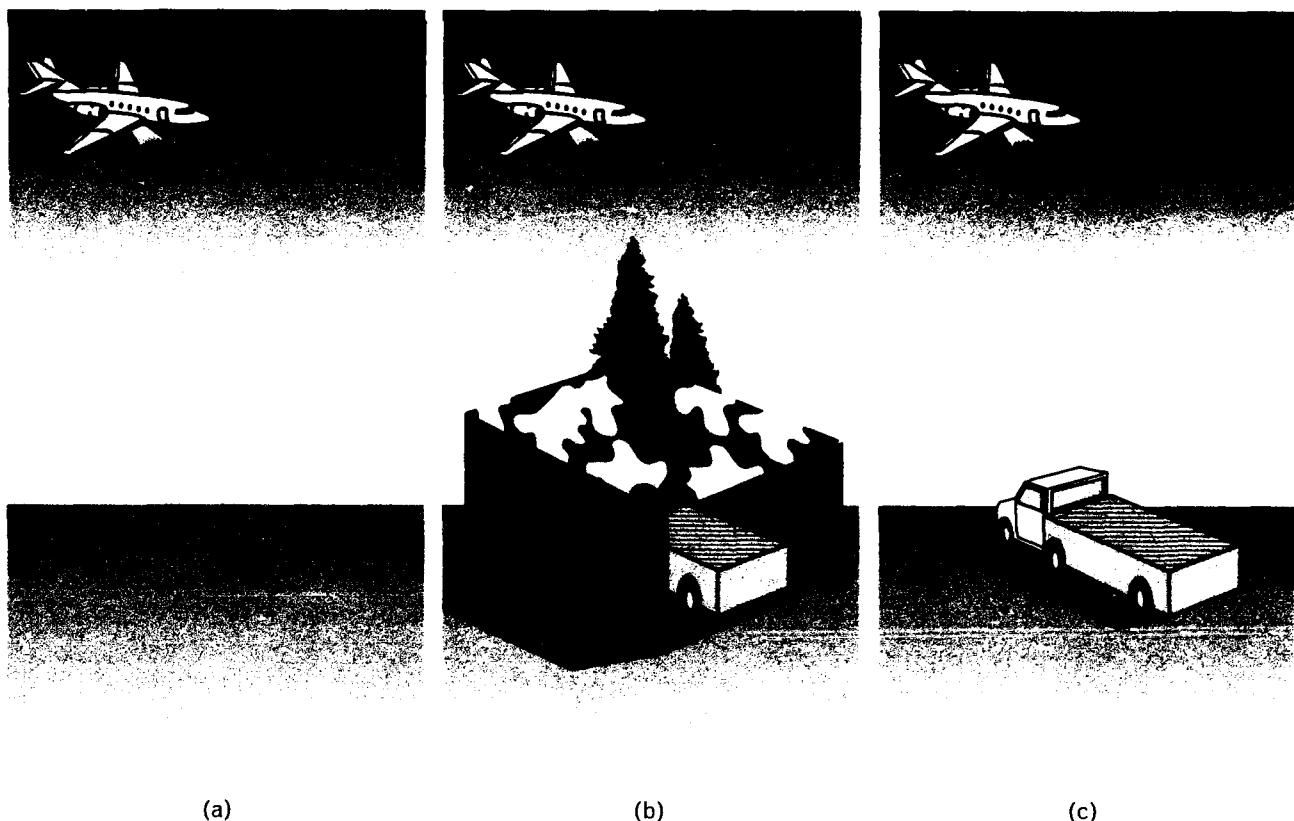
tem, a full-resolution functional template is created for the extractor by a zooming process that is applied to the corresponding detection template. The preliminary location and orientation information recorded in the hypothesis is then used to probe the window with the full-resolution template. The window is probed only at the pixels immediately surrounding the hypothesis location and only for orientations within  $10^\circ$  of the hypothesis orientation. Although the angular increment for FTC is  $10^\circ$  for detection, an increment of  $1^\circ$  is used for extraction.

At the location and orientation of the best full-resolution FTC match, a rectangular mask with outside dimensions that approximate the dimensions of the candidate target is positioned to isolate an image region. The isolated region then undergoes the application of height thresholds followed by a cleaning with MM, and the resulting region is subdivided with a stencil consisting of an array of rectangles (six or eight in our application), each marking the area limits

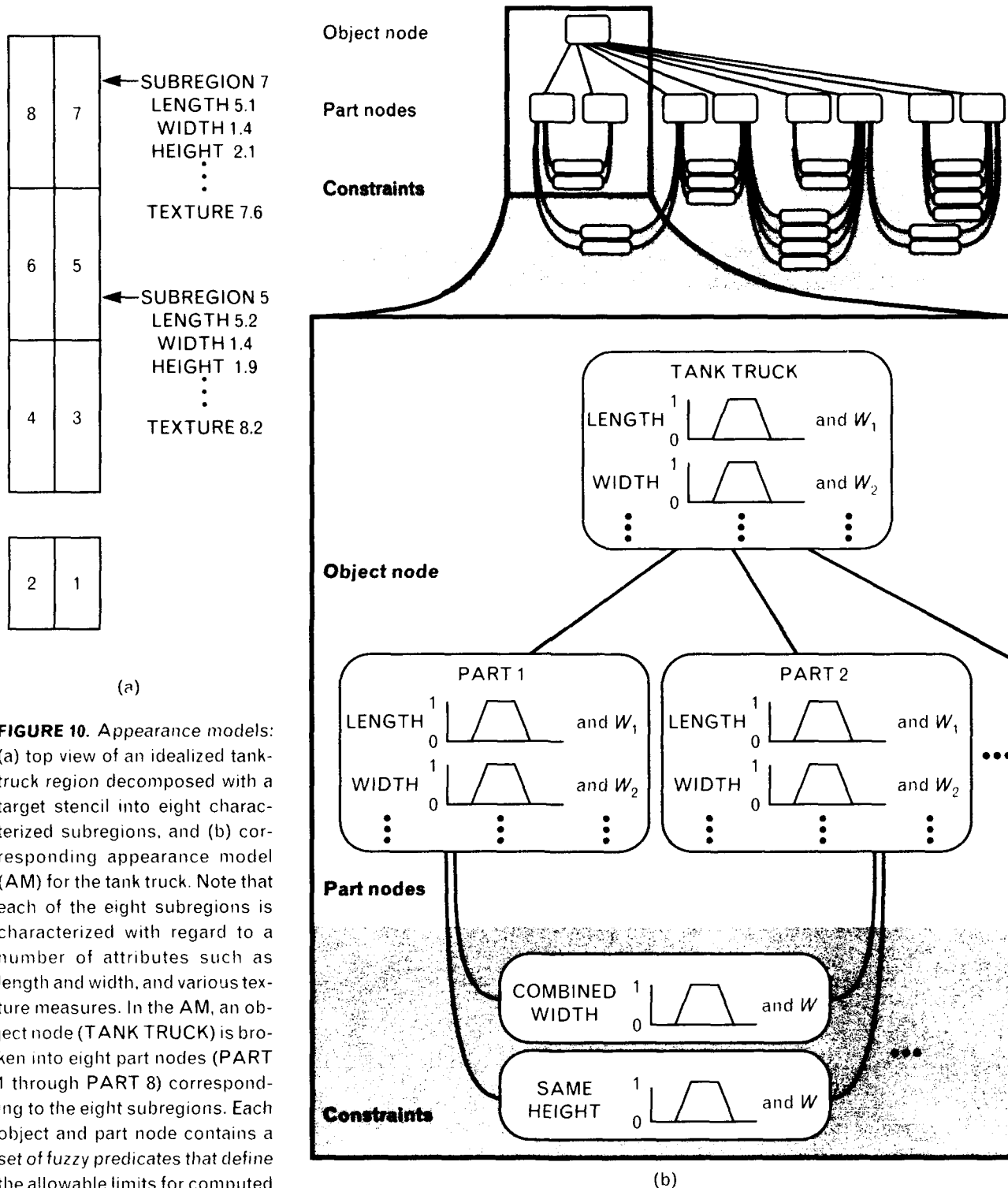
of one subregion. For the tank truck, Figure 10(a) shows the eight idealized subregions, each of which is characterized with regard to a number of attributes such as length and width, and various texture measures such as a measure of the local variance in the subregion. The characterized object region and part subregions together with a list of candidate target identities extracted from the window hypotheses serve as input to the matching process. (Note: If the matching module fails to make an identification, control is directed back to the beginning of the extraction module. In such a case, the extraction process is repeated for the hypothesis that has the next highest interest score. The processing stops either when the target has been identified or when all hypotheses have been examined.)

#### *Matching*

Candidate targets are identified by matching the object region and part subregions against appearance



**FIGURE 9.** The functional template for a tank truck is applied to (a) bare ground, (b) a 100% occluded target, and (c) a fully exposed target. The corresponding expected scores for the three cases are 0.0, 0.5, and 1.0, respectively.



**FIGURE 10.** Appearance models: (a) top view of an idealized tank-truck region decomposed with a target stencil into eight characterized subregions, and (b) corresponding appearance model (AM) for the tank truck. Note that each of the eight subregions is characterized with regard to a number of attributes such as length and width, and various texture measures. In the AM, an object node (TANK TRUCK) is broken into eight part nodes (PART 1 through PART 8) corresponding to the eight subregions. Each object and part node contains a set of fuzzy predicates that define the allowable limits for computed values of the different attributes such as length and width. Each predicate has an associated weight  $W_i$  that is used to bias computed match scores. In a similar way, constraints (e.g., COMBINED WIDTH and SAME HEIGHT) specify the limits of the relationships between the different parts. The AM shown here has been simplified. In practice, the AMs of the modeled trucks have as many as 80 constraints between the different parts. Note, also, that for the sake of simplicity, the existence of constraints between certain parts (e.g., between PART 2 and PART 4) has not been shown.

models (AM) [1-3]. Figure 10(b) illustrates the general construction of an AM for the tank truck. Note that the AM consists of an object node (TANK TRUCK) and a series of part nodes (e.g., PART 1) that specify the limits of properties of the different parts. Attributes of the object region and part subregions can include length, width, aspect ratio, circumference, average image value, and texture measurements, among other quantities. Each object and part node contains a set of fuzzy predicates and associated weights that define the allowable limits for the computed values of the different attributes. There is typically one fuzzy predicate for each attribute. In a similar way, constraints (e.g., COMBINED WIDTH and SAME HEIGHT) specify the limits of the relations between parts.

By treating a computed attribute  $\theta$  as the argument of the corresponding fuzzy predicate  $f(x)$ , we can easily obtain a score  $f(\theta)$  for the computed value  $\theta$ . The scores obtained from a set of fuzzy predicates together with the weights associated with those predicates can then be used to calculate a weighted average that provides an overall match score for each part. Similarly, a match score can be computed for each constraint. For example, the sum of the widths of PART 1 and PART 2 would be the input to the constraint COMBINED WIDTH shown in Figure 10(b). Match scores for each part and each constraint become pieces of evidence that can then be combined with the Dempster-Shafer theory of evidence. The output is a target identity, which may be *none* to indicate an unknown target type. (Note: References 1 through 3 provide a detailed description of matching based on AMs, including a description of the Dempster-Shafer theory of evidence.)

Using the above approach, we constructed five AMs, one each for the exposed tank truck, the camouflaged tank truck, the loaded logging truck, the empty logging truck, and the truck cabs. The cabs were modeled through a separate AM because, in several cases, the frame boundary of the images had occluded the trailers.

Through experience, we learned that the AMs that were more successful were generally more complicated. As the size and complexity of the AMs grew, however, it became apparent that we could not con-

tinue to construct AMs manually. Thus automatic construction techniques were needed.

Automatic model building requires example sets of the decomposed targets. For each attribute of each part of each target, fuzzy predicates can be constructed from the population of values found in the example set. Figure 11 shows a fuzzy predicate that has been constructed for the attribute LENGTH of the part node PART 1. The red dots at the top of the figure represent the population of length values from all PART 1s in the example set. During the construction of a fuzzy predicate, outlier (i.e., statistically inconsistent) values are discarded, and a cluster analysis is performed to determine the number of clusters that might best explain variances in the remaining values. For each cluster, the mean and standard deviation are computed, and an interval of maximum returned score (1.0) is established between the minimum and maximum lengths of each cluster. Outside this interval of maximum returned score, the fuzzy-predicate curve ramps down from 1.0 to 0.0 with a slope that is proportional to the standard deviation  $\sigma$  of the cluster population. The value of  $\sigma$  is multiplied by the coefficient  $\beta$  called the recognition tolerance, to determine the width of the ramping interval. For small values of  $\beta$ , the fuzzy predicate is relatively intolerant of lengths that are outside the already observed range of values, while high values of  $\beta$  result in a greater tolerance of such variations. The final fuzzy predicate is the maximum of the individual functions generated for each cluster. The weight associated with each fuzzy predicate is initialized to 0.1, a value chosen to allow an increase (and decrease) by at least an order of magnitude.

Fuzzy predicates are constructed for each attribute of each part. Not all attributes, however, are equally effective in discriminating targets from clutter. To determine which attributes are effective discriminants, we use a second phase of model building called *supervised discrimination learning*. In the process, weights associated with attributes that are weakly discriminating are decreased, while weights for attributes that are strongly discriminating are increased. Whether an attribute is discriminating or not is determined by individually reevaluating each attribute within the AMs of targets after an incorrect identification has been

made. If a fuzzy predicate returned a high score that contributed to the error, then the associated attribute is *nondiscriminating* and the corresponding weight is decreased. For example, consider the response to an FA in which some piece of clutter has been incorrectly identified as a target. In the identification process, fuzzy predicates were evaluated for the different attributes. If the score from a particular evaluation was greater than 0.5 (ambiguity), then that attribute contributed to the mistaken identity and is thus not discriminatory; consequently, the corresponding weight is decreased. On the other hand, if the score was less than 0.5, indicating that the attribute had correctly denied the mistaken identity but was outvoted by the other fuzzy predicates, then the associated weight is increased. (Note: Reference 4 contains specific equations and schedules for the weight adjustments, along with a more detailed description of supervised discrimination learning.)

## Results

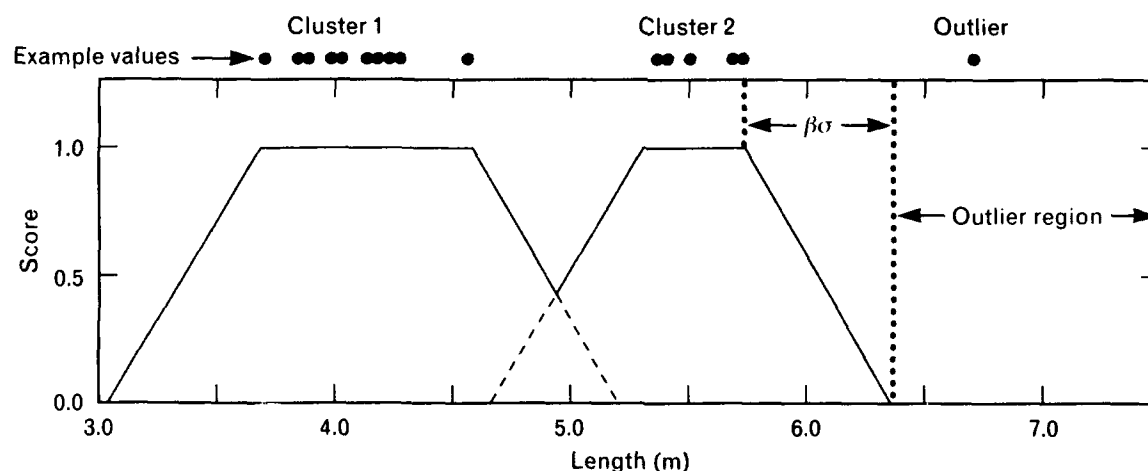
Much of the innovation of the CMT version of XTRS is in the development of techniques for low-level MI. To evaluate the effectiveness of these tech-

niques, this section will present the detection results first, separate from the results of the overall system recognition performance.

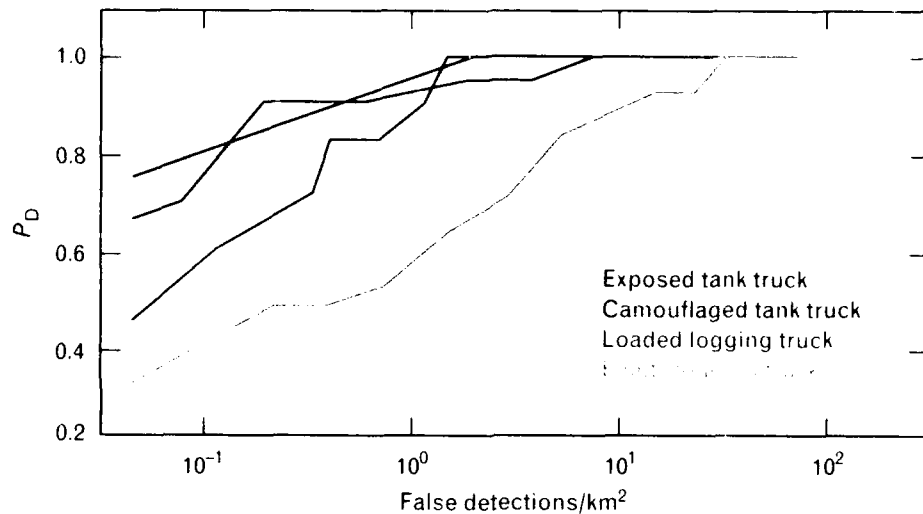
### Detection Performance

Each of the four FTC-based target detectors was tested individually for a range of interest thresholds. Figure 12 shows the probability of detection  $P_D$  plotted as a function of the false detection (FD) rate for the four implemented detectors. (Note: FD is distinct from FA, which is the false-alarm level for the *overall* system.)

For the tank-truck detectors, both exposed and camouflaged, the detection performance was quite good. In both cases,  $P_D$  was around 0.7 at the threshold level where the first FD occurred. Given the 17.13 km<sup>2</sup> of ground area covered by the dataset, the one FD resulted in a rate of 0.058 FD/km<sup>2</sup>. For a  $P_D$  of 1.0, the associated minimum FD rate was approximately 2.0 FD/km<sup>2</sup>. The target detector for the loaded logging truck performed slightly less well. Because the shape of the vehicle changed with each load of logs, the detector's functional template had to be constructed with more fuzziness; i.e., the template had to



**FIGURE 11.** Automatic construction of a fuzzy predicate for the attribute LENGTH of the part node PART 1 of Figure 10(b). The red dots at the top of the figure represent the population of length values from all PART 1s in the example set. During the construction of a fuzzy predicate, outlier (i.e., statistically inconsistent) values are discarded, and a cluster analysis is performed to determine the number of clusters that might best explain variances in the remaining values. For each cluster, an interval of maximum returned score (1.0) is then established between the minimum and maximum lengths of that cluster. Outside this interval, the fuzzy-predicate curve for the cluster ramps down to 0.0 with a slope that is proportional to the standard deviation  $\sigma$  of the cluster population multiplied by the recognition tolerance  $\beta$ . The final fuzzy predicate is the maximum of the individual functions generated for each cluster.



**FIGURE 12.** Probability of detection,  $P_D$ , plotted as a function of the false-detection rate for the four target detectors. Each point along any of the curves shown is associated with a particular value of the interest (or detection) threshold.

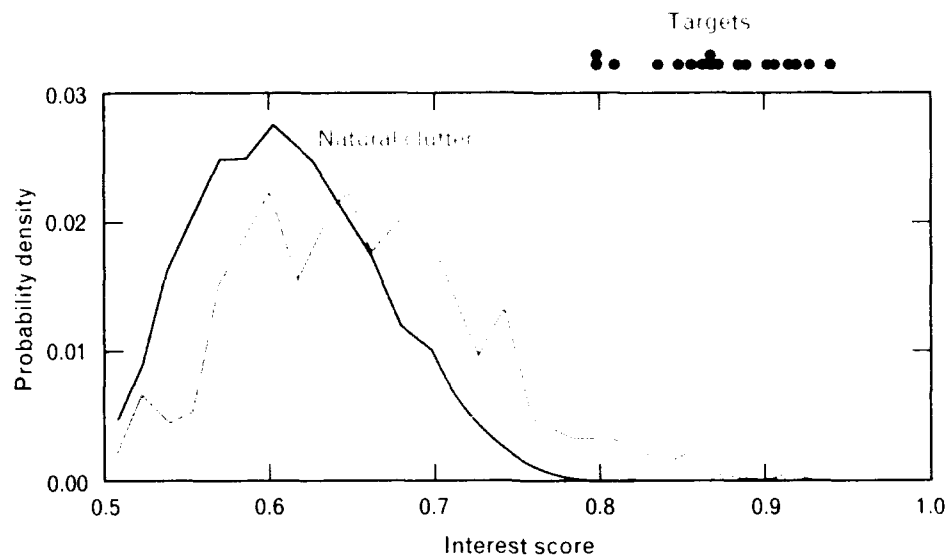
provide a greater tolerance for variations in shape. But the most difficult to represent as a functional template was the empty logging truck, because of the small size of the vehicle's trailer. With an elongate shape such as that of the tank truck, small uncorrected distortions in the length of the target did not have a serious effect. With the empty logging truck, however, the rear axles of the vehicle are the only reliably visible part of the trailer and, because this portion of the truck is short relative to the overall truck length, even a small distortion in the truck length can move the axles ahead or behind the patch of the functional template representing the axles. Consequently, the shape and appearance of the empty logging truck could not be defined as precisely as for the other targets. Fortunately, scoring functions can be modified easily to adjust the degree of tolerance to variations in shape and appearance. Although the detection rates for the two logging-truck configurations were lower for a given FD rate than for the tank truck, the performance was still respectable.

For a better understanding of the sources of FDs, the clutter data were divided into natural and man-made (cultural) clutter. Any image containing a large, man-made object (e.g., a building, non-target vehicles, or stacks of logs) was placed in the cultural-clutter group. Each of the target detectors was then applied to both divisions of the data. Figure 13 shows

example results for the loaded logging truck. The probability density was computed as the percent of all detections found within each successive short interval of interest scores (0.016 in the range from 0.0 to 1.0).

Most instances of natural clutter (mainly trees and shrubs) tended to have interest scores around 0.65, with no interest scores above 0.8. The population of target interest scores (shown as red dots at the top of Figure 13) had scores ranging from 0.79 to 0.94. Thus the detector for the loaded logging truck achieved a perfect partition between targets and natural clutter (i.e., a threshold of 0.78 resulted in 100% detection with no FDs). In contrast, man-made objects were a more troublesome source of FDs because such objects generated a few interest scores that were as high as 0.9. Included in the high-scoring cultural objects were other semi-trailer trucks, such as the deployed tank truck, and stacks of logs similar in shape to the loads carried by the logging trucks. The results for the other target detectors were roughly the same as that for the loaded logging truck with the detectors for the tank truck providing a slightly better partition between targets and clutter, and the detector for the empty logging truck a slightly worse partition.

It can be argued that logging trucks provide more stressful testing than would arise from an actual CMT application. Because missile launchers are large and nonarticulated, their detection is less vulnerable to



**FIGURE 13.** Distributions of interest (or detection) scores for natural and man-made (cultural) clutter for the loaded-logging-truck target detector. The red dots at the top of the figure indicate the interest scores for instances of the deployed loaded logging trucks. At an interest (or detection) threshold of about 0.78, the loaded-logging-truck target detector achieves perfect discrimination between deployed loaded logging trucks and natural clutter. There is no threshold, however, that would yield a perfect discrimination between these targets and cultural clutter.

the effects of distortion. Also, CMTs have only three basic shape variations: missile down for transport, missile erected for launch, and without a missile following launch. These three variations have precise known shapes, in contrast to the amorphous nature of log loads.

#### *Occlusion Experiments*

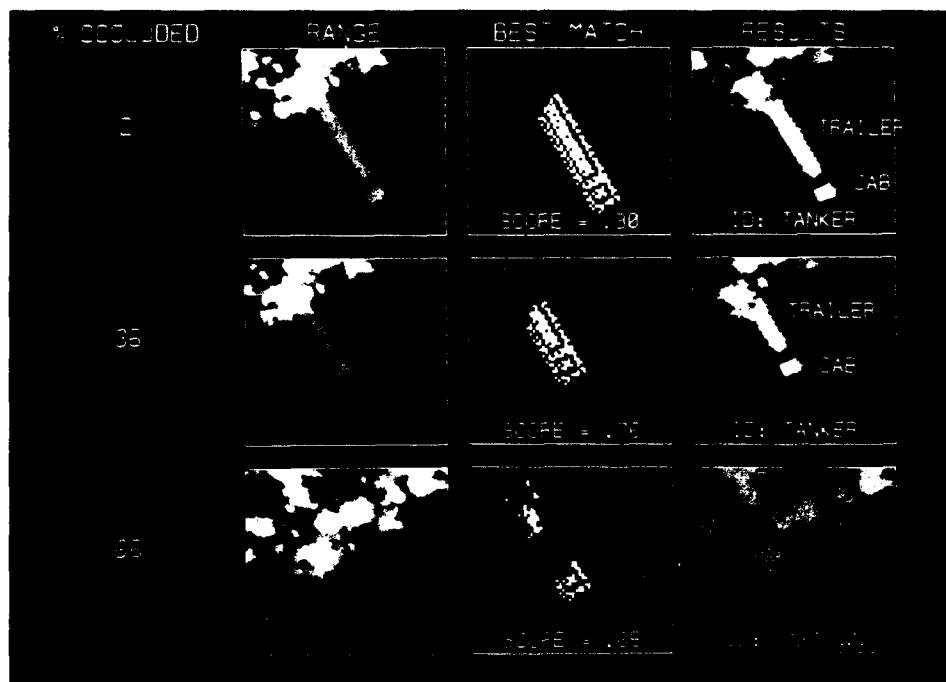
One of the primary motivations for the development of FTC was to overcome the way in which occlusion disrupted the more traditional shape-matching techniques. Various attempts to design an MM approach for detection and extraction failed with targets that had as little as 5% of their surface areas occluded by foliage. With functional templates, however, all targets could be detected readily without much challenge.

To explore the limitations of FTC in detecting targets occluded by foliage, we designed experiments in which targets that were cut out from one image were positioned along a tree line within another image. Beginning at locations where the target was completely unobscured, the target was incrementally moved under the foliage, with the vehicle's major (i.e., longi-

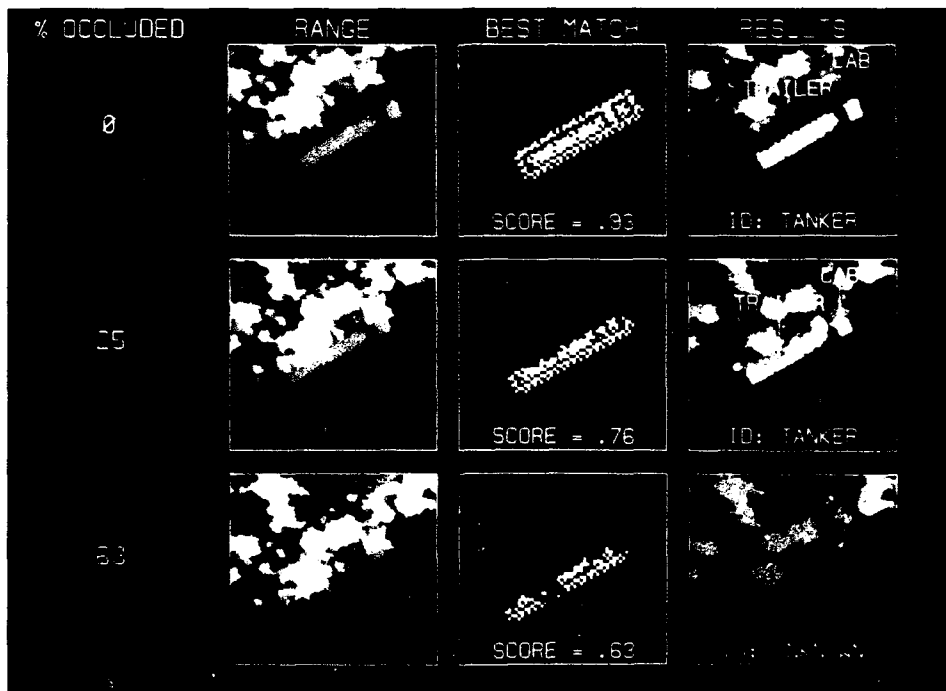
tudinal) axis either perpendicular or parallel to the tree line. Figure 14 summarizes the results of the experiment with the target perpendicular to the tree line for target occlusions of 2%, 36%, and 66%.

The left frames in each row are range images of a tank truck that has been synthetically placed perpendicular to the tree line. The center frames show the location and orientation of the best match for the tank-truck functional template in the images, along with the corresponding interest scores. The pixels themselves indicate the scores returned from individual scoring functions for each location on the template: black, white, and gray pixels represent 0.0, 1.0, and intermediate values, respectively. The right frames show the results of recognition based on the matching of the decomposed target with AMs of the targets. It should be noted that occluded targets were included in the example set used to build the AMs.

The first row of Figure 14 shows that a target that is almost completely exposed (only 2% occlusion) results in a strong interest score and correct recognition. For a target occlusion of 36%, the interest score is barely above the interest threshold of 0.75, but the target is recognized correctly nonetheless. For an oc-



**FIGURE 14.** Summary of foliage occlusion experiment for a tank truck that has been synthetically positioned perpendicular to a tree line. The top, middle, and bottom rows are for the target with 2%, 36%, and 66%, respectively, of its surface area occluded by foliage. The left frames are range (height above ground) images, the center frames indicate the locations and orientations corresponding to the highest interest scores (indicated in the frames), and the right frames show the final AM-based recognition results.

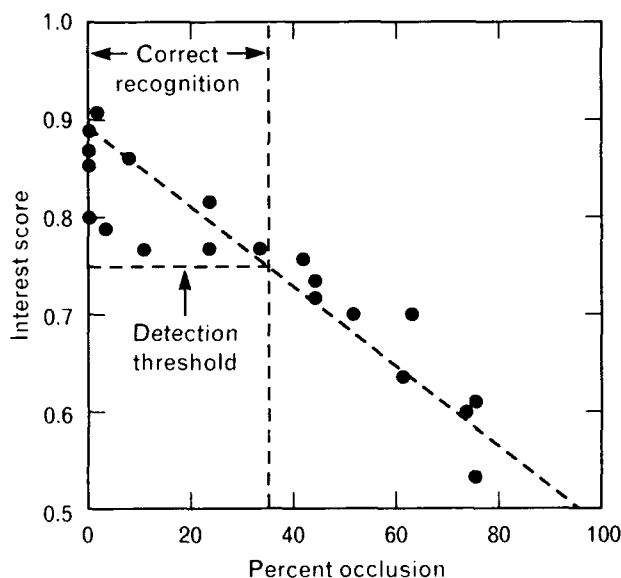


**FIGURE 15.** Summary of foliage occlusion experiment similar to that of Figure 14, except the tank truck has been positioned parallel to the tree line.



clusion of 66%, the interest score is only 0.65, which is below the interest threshold of 0.75. Consequently, this target is not detected and therefore not processed by the matching module. (The UNKNOWN identification in the figure is the result of our dropping the interest threshold to below 0.63.) And yet, as shown in the center frame, the best match produced by FTC correctly determines the location and orientation of the target, despite the target's being more than half occluded. We have obtained similar results for the case in which the major axis of the tank truck is parallel to the tree line, as shown in Figure 15.

For the combined data of perpendicular and parallel target placements, Figure 16 contains a plot of interest score as a function of percent occlusion. The figure shows that the decrease in interest score as a function of percent occlusion conforms to an expected linear relationship: performance degrades gradually as occlusion increases, without any intervals of rapid degradation. At an interest threshold of 0.75, targets occluded up to around 36% are detected and recognized. Lowering the threshold would permit the detection and recognition of targets with an even higher percent of occlusion, but would also increase the FD rate.



**FIGURE 16.** Summary of results for the experiments described in Figures 14 and 15. Note that at the detection interest threshold of 0.75, the system is able to detect and recognize targets with up to 36% of their surface areas occluded by foliage.

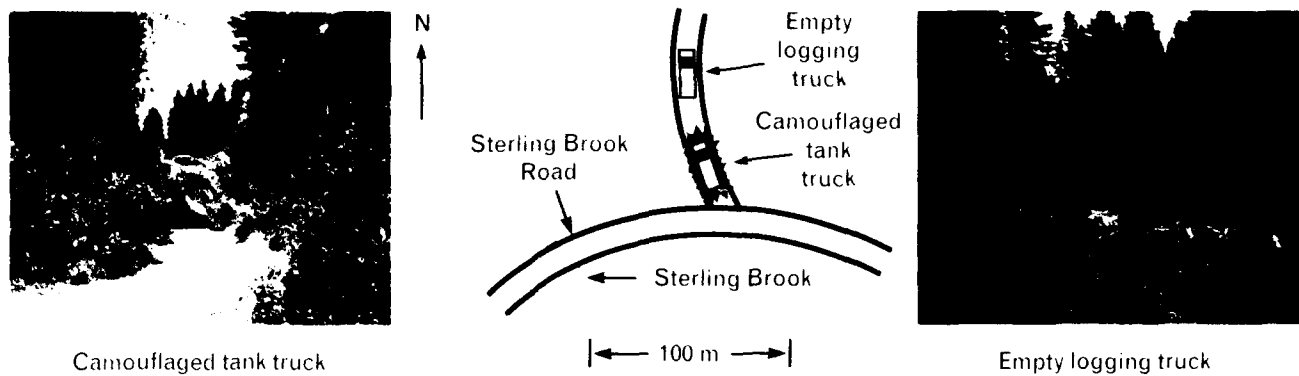
### Overall System Performance

Although FTC can provide high detection rates with few FDs, the fewer FDs the better. Because FTC bases its interest scores solely on how well image values compare to expectations at different locations on the kernel, it does not exploit the known relationships of target parts. We have relied on the technique of AMs as a means of modeling such additional information and, by so doing, have provided the means for rejecting FDs and discriminating between multiple, similarly shaped target classes.

For initial testing, we used an interest threshold of 0.75 to detect clusters of high interest values in the Maine dataset. A total of 492 detections resulted, including all 63 deployed targets. These detected targets were then extracted, characterized, and matched against AMs, as described earlier.

Initially, when we built the AMs we used 50% of the targets as examples and a recognition tolerance  $\beta$  of 0.3. The remaining targets that were classified as UNKNOWN (i.e., insufficiently like any modeled target) were subsequently added to the example set as we refined the models. Eventually, 80% of the targets were included in the example set to reach a recognition performance of 100%. No supervised learning of weights was done for this test. Under these conditions, there were no FAs in all 17.13 km<sup>2</sup> (2303 image pairs) of data.

The above results include some targets deployed in the open, but they also include a number of very difficult cases. Figure 17 shows photographs and a map of a deployment in which the empty logging truck and the camouflaged tank truck were placed on a narrow dirt road with tall trees on either side. For the empty logging truck, Figure 18 contains the range and intensity images, an interest image highlighting pixels having above-threshold interest values and showing the selected windows, and an image showing the final recognition results. The truck, visible in the lower left corner, has been correctly recognized. An FD, triggered by a collection of shrubs having roughly the size and spacing of the parts of the empty logging truck, was correctly rejected during AM-based matching. Figure 19 shows the results for the camouflaged tank truck of Figure 17. Note that in this case the



**FIGURE 17.** Photographs and deployment map of "hidden targets" used in the Portage, Maine, experiments.

target is represented in the range image mostly as a broad tent with only a few pixels having height values corresponding to the ground. Because the shape of the camouflage netting can vary from site to site, the scoring functions for the range functional template were constructed to incorporate considerable uncertainty and were thus weakly discriminating. In intensity images, however, the camouflage netting actually helps make the target stand out against the background, probably because of interference effects caused by some of the laser energy being reflected by the netting material and some being reflected by the ground [17]. The scoring function for the intensity functional template for the camouflaged tank truck exploits this phenomenon.

#### *Generalization of AMs*

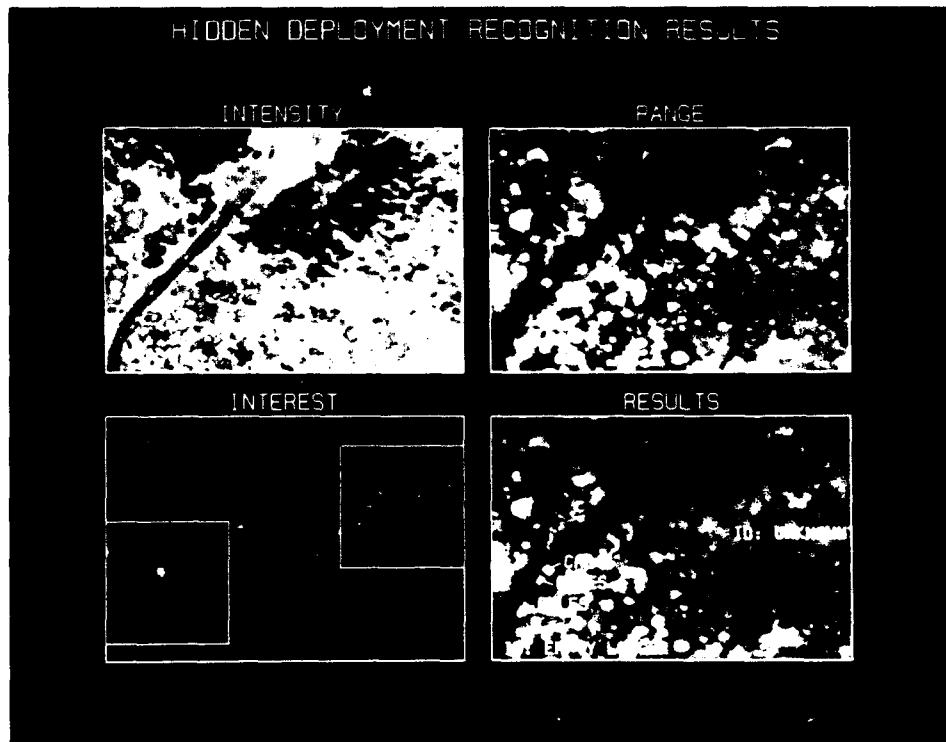
In the course of evaluating the dataset, we discovered that, in addition to the one logging truck that Lincoln Laboratory personnel had deployed, there were six other empty logging trucks in the vicinity. Only two of these six trucks had interest scores greater than 0.75, and neither of the two was recognized as an empty logging truck. Figure 20 shows intensity and range images containing three of the six non-Lincoln Laboratory trucks, along with a road-mobile crane (top center). The truck in the image that had an interest score above threshold was classified as UNKNOWN. Initially, we were disappointed with this result until we realized that the discrimination made by XTRS was in fact reasonable and useful. Figure 21 (left) is a range image of the logging truck deployed by Lincoln Laboratory, and Figure 21 (right) is a

range image of one of the six other trucks. Note that although both vehicles serve the same function and are called logging trucks, their appearances are in fact distinct. The truck deployed by Lincoln Laboratory has a tractor with the cab directly over the engine, while the other vehicle has a hooded tractor with the cab behind the engine. Also, although the overall lengths of the trailers are the same, the trailer of the Lincoln Laboratory truck is narrower and lighter in appearance. If the two vehicles are considered as two distinct objects, then XTRS did successfully discriminate between the two variants with 100% accuracy.

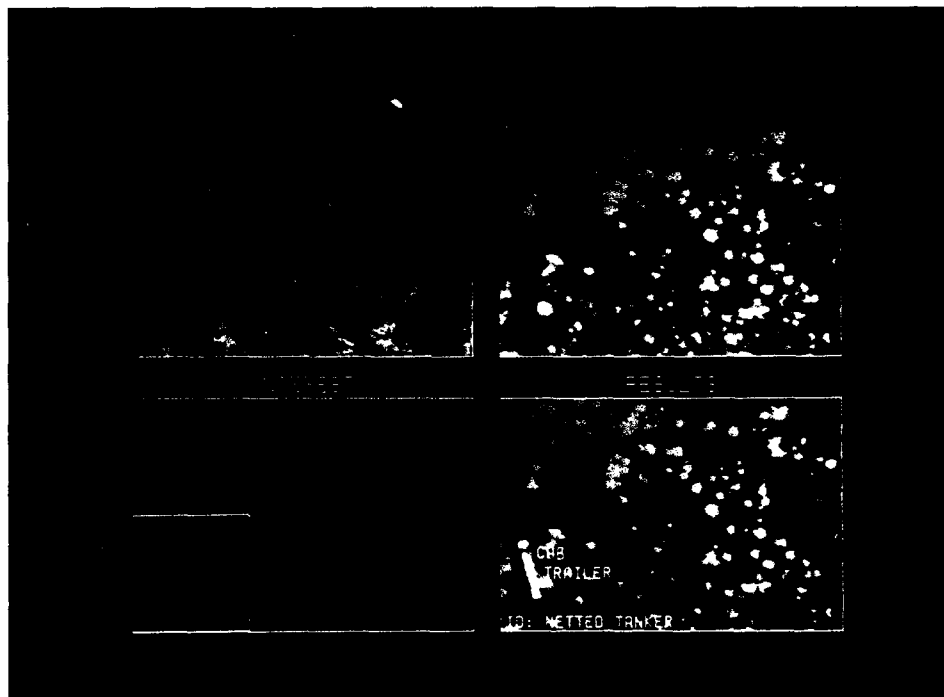
Suppose, however, that AMs we built using examples of one model within a target class were to be used to recognize a more general class of targets, including other models not represented in the example set. All six of the trucks not deployed by Lincoln Laboratory were detected by decreasing the interest threshold from 0.75 to 0.72. And all six vehicles were recognized as empty logging trucks by increasing the recognition tolerance  $\beta$  from 0.3 to 1.7. Thus, by using only two tunable parameters, we could adjust the generality of the recognition for the entire system. But the relaxation needed to generalize the recognition had an associated cost: the FA rate for the system as a whole increased from 0.0 to 1.7 FA/km<sup>2</sup>. Of course, instead of generalizing the AMs to include similar related targets, we could have constructed additional functional templates and AMs.

#### *Supervised Discrimination Learning of Model Weights*

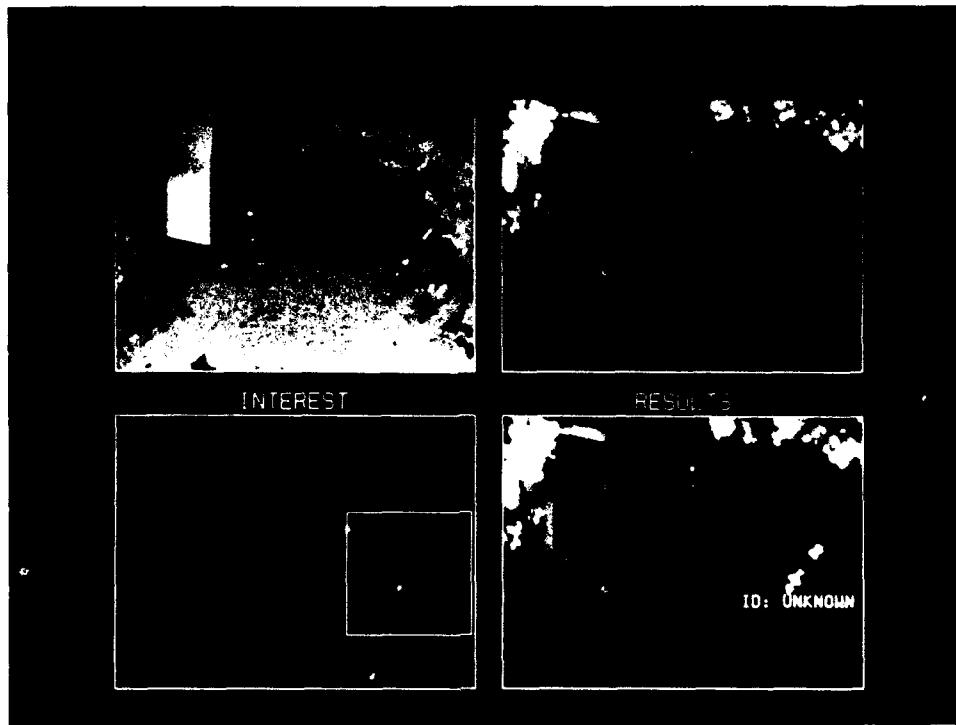
A common criticism of many research ATR systems is that, because of the limited availability of data, the



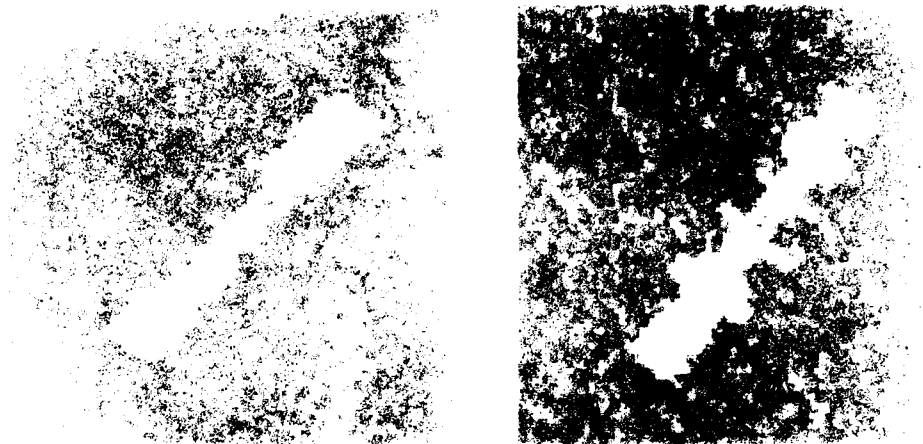
**FIGURE 18.** Detection and recognition results for the hidden empty logging truck of Figure 17. The truck is visible in the lower left corner of the images. A false detection, triggered by a collection of shrubs, has been correctly rejected as UNKNOWN.



**FIGURE 19.** Detection and recognition results for the hidden camouflaged tank truck of Figure 17.



**FIGURE 20.** Detection and recognition results for three empty logging trucks that were not part of the test deployment. (Note: The images also contain other vehicles, including pickup trucks and a road-mobile crane, at the top center. Also note that in the range image the building in the upper left corner has incorrect height values due to an artifact in estimating the height of local ground for large objects.) Two of the three trucks received below-threshold interest scores and were thus not detected. The remaining truck had an interest score above threshold but was classified as **UNKNOWN**. The system failed to detect and recognize the three trucks because of differences between them and the logging truck deployed by Lincoln Laboratory (see Figure 21).

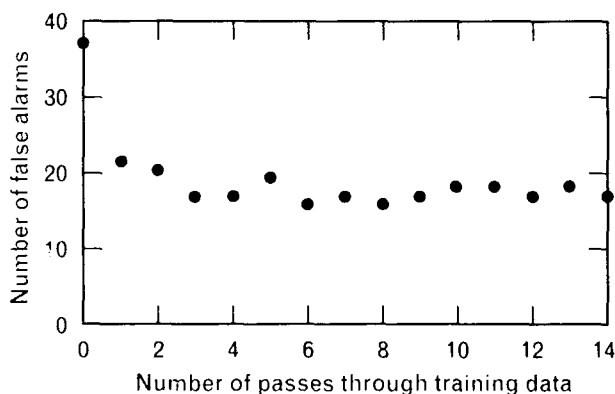


**FIGURE 21.** Enlarged range images for (left) empty logging truck deployed by Lincoln Laboratory, and (right) empty logging truck discovered in the image dataset. Although both vehicles are called logging trucks, their appearances are in fact distinct. The cab for the truck on the left is directly over the engine, while the cab for the other vehicle is behind the engine. Also, although the overall lengths of the trailers for the two trucks are the same, the trailer for the truck on the left is narrower and lighter in appearance.

dataset used to train a system is the same set used for testing. To a certain extent, we addressed this criticism by dividing our available data into two sets, one for training, the other for testing. Because at most only 50% of the targets were to be used in the example training set, we assumed that a large recognition tolerance  $\beta$  would be required to achieve a  $P_D$  of 1.0. With a large recognition tolerance, we also expected that the FA rate might be high. Consequently, the supervised learning of weights was used to suppress the FAs.

For training, we built the AMs with a high  $\beta$  value of 5.0 and an example set consisting of 1165 image pairs (range and intensity) containing 28 targets. AM weights were all initialized to 0.1. To establish a baseline FA rate, we did not use supervised discrimination learning to process the training data. The high  $\beta$  of 5.0 and a low interest threshold of 0.72 were selected so that enough FAs would be generated to promote opportunities for learning. The number of FAs under these conditions was 37 (4.3 FA/km<sup>2</sup>). Next, supervised discrimination learning was initiated and, with each complete pass through the training data, the number of FAs generated during that pass was recorded. Figure 22 shows the results of 14 passes through the training data. Note that the number of FAs dropped from 37 to 21 during the first pass and stabilized to an average of 19 FAs (2.2 FA/km<sup>2</sup>) by the fourth pass.

After the completion of training, testing was done on 27 targets in 1134 image pairs covering 8.4 km<sup>2</sup> of ground area. Using the weights learned from training,



**FIGURE 22.** Learning curve showing the decrease in false alarms with training.

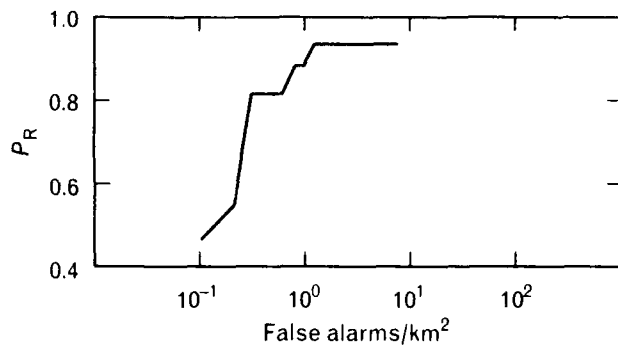
we built the AMs at progressively larger values of  $\beta$  ranging from 0.1 to 7.0. For each  $\beta$  value, the probability of correct recognition  $P_R$  was plotted as a function of the FA rate. Figure 23 shows the results. At the value of  $\beta$  in which the first FA occurs, the  $P_R$  was 0.44. For a  $P_R$  of 0.93, the FA rate was 1.42 FA/km<sup>2</sup>. Two of the targets were not recognized (i.e., classification UNKNOWN) even with  $\beta$  set to 7.0. At some higher value of  $\beta$ , we do expect to achieve a  $P_R$  of 1.0, but we did not attempt to find that particular  $\beta$  value. In this test, as well as in the previously described tests, no targets were mislabeled as another target identity.

Out of a total of 63 targets, only 55 were used: 28 for training and 27 for testing. The reason for this intentional omission of eight targets was that there were only four images of the tank truck in the open and four images of an empty logging truck with the trailer completely occluded by the frame boundary of the image. With only two examples of a target for training, the resulting AMs were too restrictive to recognize any targets other than the two training examples. This finding highlights how the building of robust AMs depends on the proper selection of a training set. As with any learning system, a realistic and representative sampling of variations of object appearance is necessary to achieve robust performance.

## Conclusions

The Experimental Target Recognition System (XTRS) provides a framework for applying machine intelligence (MI) techniques to the task of automatic target recognition (ATR). Based largely on aspects of fuzzy set theory, these MI techniques enable the representation of uncertainties and known variabilities in target appearance.

With rule-based experts and libraries of functions and data structures, XTRS can be organized to adapt automatically to environmental context and to reconfigure the search for alternative targets. Using multiple target detectors, XTRS can look simultaneously for different variations in target shape. The outputs of all target detectors are expressed as interest images, permitting the fusion of all sources of evidence into a single spatial map. Despite the apparent complexity of XTRS, system performance can be con-



**FIGURE 23.** Probability of correct recognition  $P_R$  as a function of the false-alarm rate. Each point along the curve corresponds to a particular value of the recognition tolerance parameter  $\beta$ .

trolled effectively with only two tunable parameters: the interest threshold for controlling the output of low-level detection and the recognition tolerance for controlling the output of high-level matching based on appearance models (AM).

XTRS uses AMs to model how targets and their constituent parts appear in sensor imagery, thus providing an alternative to other classifiers, including those based on neural network, statistical, and other model-based approaches. Unlike other model-based approaches that encode the three-dimensional structure of an object, AMs define the observable appearance of targets in specific sensor data within the constraints of the likely target orientations. AMs provide a more controllable representation than neural networks. Because knowledge is represented in neural networks as a diffuse population of weights, it is difficult to identify which image features are being used. Not only are the attributes and weights of AMs easy to interpret, they can be modified by a user with predictable effects on recognition performance. Neural networks have gained in popularity as classifiers principally because of their ability to learn and encode discriminants automatically. As we have shown in this article, the automatic learning of class discriminants is also possible with AMs, but in a representation that is more amenable to understanding and selective editing.

Other techniques developed for XTRS embody what we call low-level MI. Most existing MI techniques used in computer vision rely on a preliminary abstraction of raw data into a symbolic form. But the

process of abstraction necessarily reduces the amount of information available for decision making, thus handicapping an observer, no matter how intelligent. Tools for knowledge-based signal processing and pixel-level accumulation of evidence provide the intelligent means of using object- and context-dependent knowledge to guide the extraction of information directly from raw image data without the need for abstraction. In particular, functional template correlation (FTC) allows the construction of generalized matched filters that encode knowledge of the physics of a detection problem. Customized operations constructed with FTC are generally more powerful (i.e., more discriminating) than comparable traditional signal processing operations. In ATR versions of XTRS, we have used FTC as a one-step three-dimensional target recognizer. For other applications, we have developed knowledge-based fuzzy variations of standard image processing operations, including thin-line detection, smoothing operations, basic mathematical morphology (MM) operations, and pattern matching.

The need for FTC arose from a perceived inadequacy of the standard techniques of shape analysis. Although MM worked very well for unobscured targets, we could not devise a sequence of MM operations that would reliably detect and extract targets in high-clutter environments, especially when the target was partially occluded. We believe that our failure was due in part to the all-or-nothing nature of MM operations [18]. We have also investigated the use of normalized cross-correlation, the other commonly used tool for shape analysis. In its favor, cross-correlation does provide a variable degree of match that can be translated easily to interest values. But the matches generated by cross-correlation are too literal in that the interest scores are based on very specific, inflexible patterns of image values.

The repetitive evaluation of all scoring functions in a functional template—for all orientations for each pixel location—sounds computationally prohibitive. But the process becomes feasible when the input image values are scaled to some integer range (e.g., 0 to 255) and the scoring functions are implemented as a precomputed two-dimensional lookup table that is indexed by the scoring-function numbers and the integer image values. The use of such a lookup table is

**Table 1. Performance of Prototype System**

	<i>Experiment</i>			
	1	2	3	
			Training	Testing
Number of targets	63	69*	28	27
Ground area (km <sup>2</sup> )	17.13	17.13	8.6	8.4
Detection threshold	0.75	0.72	0.72	0.75
Number of detections	492	1173	656	342
Number of targets in AM example set	50	50	28	—
Recognition tolerance	0.3	1.7	5.0	4.5
% correct recognition	100%	100%	100%	93%
FAs/km <sup>2</sup>	0.0	1.7	2.2**	1.4

\* Includes six empty logging trucks not intentionally deployed

\*\* After discrimination learning

generally faster than multiplication, making FTC evaluation quicker than cross-correlation.

Low-level MI also allows XTRS to delay the application of thresholds. Instead of applying thresholds either to a single image consisting of raw data or to the output of some simple transformation of the raw data, we can apply the thresholds to maps of interest containing evidence that has been extracted from a variety of sources.

Unlike the AMs, the FTC-based target detectors were constructed and tuned manually. The development of a useful, operational ATR system that is able to adapt swiftly to different targets and mission scenarios requires a mechanism for constructing functional templates automatically. We have developed methods for building functional templates from statistics accumulated from example targets, but these methods have not yet been implemented. Functional templates might also be constructed by using the emerging techniques of genetic programming.

The success of our approach to ATR is indicated by the overall system performance of the prototype system, as summarized in Table 1. In experiment 1, in which we used strict tolerances for the automatic construction of the target AMs, we were able to achieve

100% correct target recognition in the available data with no mislabelings and no false alarms. It is important to note that the AMs are flexible and can be generalized to broader classes of vehicles by the manipulation of a single recognition tolerance. Experiment 2 demonstrates this flexibility and, in particular, the capability for generalization by increasing the recognition tolerances. Six empty logging trucks were found in the dataset that were somewhat different from the one logging truck that was intentionally deployed. These six were appropriately rejected as clutter in experiment 1. Suppose, however, that the additional six trucks were to be included in a broader class of empty logging trucks. By changing just the recognition tolerance from 0.3 to 1.7 in experiment 2, the system was able to recognize the six trucks as empty logging trucks. Of course, the cost of generalizing all models in this manner was that the FA rate increased from 0.0 to 1.7 FA/km<sup>2</sup>. Experiment 3 shows that AMs constructed from more limited training sets can be used to recognize targets with reasonable reliability in a separate test set. The training sets were limited in size and did not provide a good representative sampling of vehicle appearances. Consequently, AMs were constructed with large recognition

tolerances in order to achieve high detection rates. The resulting elevated false-alarm rate was suppressed by roughly 50% through the use of supervised discrimination learning. Despite these limitations, reasonably good performance was evident in the separate test dataset.

In contrast, without the techniques of low-level MI and the automatic construction of complex AMs, we were unable to construct an ATR system for this application anywhere near as accurate, flexible, or robust as the one described in this article [18]. Of course, some credit for the performance of the system must go to the quality of the sensor images used. But images of good quality do not necessarily guarantee reliable detection performance. Even with an image of excellent quality, concealment and clutter can make target detection a challenging problem.

So far, XTRS has been applied to two other ATR problems: the recognition of armored vehicles both in forward-looking laser radar images [8] and in fully polarimetric synthetic-aperture radar images [9]. But XTRS provides the means of solving a more general class of object-detection problems. In addition to its use in recognizing military targets, XTRS has been applied successfully to the task of detecting and tracking hazardous weather phenomena in Doppler weather radars [11].

### Acknowledgments

The authors thank their colleagues in the Opto-Radar System Group at Lincoln Laboratory for providing the extensive laser radar dataset used in this research. This work was sponsored by the Defense Advanced Research Projects Agency (DARPA).

## REFERENCES

1. J.G. Verly and R.L. Delanoy, "Appearance-Model-Based Representation and Matching of 3-D Objects," *Proc. 3rd Intl. Conf. on Computer Vision* (Osaka, Japan, 4-7 Dec. 1990), p. 248.
2. J.G. Verly, B. Williams, and R.L. Delanoy, "Model-Based Pattern Recognition," U.S. Patent No. 5,123,057 (June 1992).
3. J.G. Verly, B. Williams, and R.L. Delanoy, private communication.
4. R.L. Delanoy, J.G. Verly, and D.E. Dudgeon, "Automatic Building and Supervised Discrimination Learning of Appearance Models of 3-D Objects," *SPIE* 1708, 549 (1992).
5. R.L. Delanoy, J.G. Verly, and D.E. Dudgeon, "Functional Templates and Their Application to 3-D Object Recognition," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, 23-26 Mar. 1992, p. III-141.
6. R.L. Delanoy and J.G. Verly, "Computer Apparatus and Method for Fuzzy Template Shape Matching Using a Scoring Function," U.S. Patent No. 5,222,155 (June 1993).
7. R.L. Delanoy, J.G. Verly, and D.E. Dudgeon, "Pixel-Level Fusion Using Interest Images," *Technical Report 979*, MIT Lincoln Laboratory (26 Apr. 1993).
8. J.G. Verly, R.L. Delanoy, and D.E. Dudgeon, "Machine Intelligence Technology for Automatic Target Recognition," *Linc. Lab. J.* 2, 277 (1989).
9. J.G. Verly, R.L. Delanoy, and C. Lazott, "Principles and Evaluation of an Automatic Target Recognition System for Synthetic Aperture Imagery Based on the Use of Functional Templates," *SPIE* 1960 (1993), to be published.
10. J.G. Verly, R.L. Delanoy, and D.E. Dudgeon, "Model-Based System for Automatic Target Recognition from Forward-Looking Laser-Radar Imagery," *Opt. Eng.* 31, 2540 (1992).
11. R.L. Delanoy and S.W. Troxel, "Machine Intelligent Gust Front Detection," in this issue.
12. L.M. Novak, M.C. Burl, R.D. Chaney, and G.J. Owirka, "Optimal Processing of Polarimetric Synthetic-Aperture Radar Imagery," *Linc. Lab. J.* 3, 273 (1990).
13. J. Serra, *Image Analysis and Mathematical Morphology* (Academic Press, New York, 1982).
14. T.R. Esselman and J.G. Verly, "Applications of Mathematical Morphology to Range Imagery," *Technical Report TR-797*, MIT Lincoln Laboratory (Dec. 1987), DTIC #AD-189316.
15. T.R. Esselman and J.G. Verly, "Some Applications of Mathematical Morphology to Range Imagery," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)* 1, Dallas, Apr. 6-9 1987, p. 245.
16. T.R. Esselman and J.G. Verly, "Feature Extraction from Range Imagery Using Mathematical Morphology," *SPIE* 845, 233 (1987).
17. Private communication.
18. Private communication.





**RICHARD L. DELANOY** is a staff member of the Machine Intelligence Technology Group. His work spans the fields of computer vision, machine learning, and construction of object recognition systems. From 1980 to 1983, he was a research scientist at the University of Virginia Department of Psychology, where he investigated the biochemical correlates of learning and the effects of stress related hormones on electrophysiological models of memory. Before joining Lincoln Laboratory in 1987, he worked for GE Fanuc Automation N.A., Inc., as a software engineer developing numerical and programmable controllers for manufacturing automation. Dick received a B.A. degree in biology from Wake Forest University in 1973, a Ph.D. degree in neuroscience from the University of Florida College of Medicine in 1979, and an M.S. degree in computer science from the University of Virginia in 1987. He was a National Science Foundation Predoctoral Fellow and a National Institute of Mental Health Postdoctoral Fellow.



**JACQUES G. VERLY** received the Ingénieur Electronicien degree from the University of Liege, Belgium, in 1975. Through a sponsorship of the Belgian American Educational Foundation, he came to the United States that year and received an M.S. degree and a Ph.D. degree in electrical engineering from Stanford University in 1976 and 1980, respectively. At Stanford he performed doctoral research in the fields of image reconstruction and restoration. Upon graduation, he joined Lincoln Laboratory, where he has worked on (among other things) computer vision problems associated with laser radar data and, more recently, with fully polarimetric synthetic-aperture radar (SAR) imagery. He has over 40 publications in the areas of image reconstruction from projections (tomography), image processing, optical signal processing, distributed signal processing, mathematical morphology, computer vision, and automatic target recognition. Jacques is the coholder of two U.S. patents in the area of model-based object recognition, and is a fellow of the Belgian American Educational Foundation.



**DAN E. DUDGEON** is a senior staff member in the Machine Intelligence Technology Group, where his focus of research has been in automatic target recognition. Before joining Lincoln Laboratory in 1979, he worked for Bolt, Beranek, and Newman, Inc., in Cambridge, Massachusetts. Dan received the following degrees from MIT: an S.B. in electrical science and engineering, and an S.M., an E.E., and a Sc.D. in signal processing. He was the corecipient of the 1976 IEEE Browder L. Thompson Prize, and is the coauthor of two books: *Multidimensional Digital Signal Processing* (Prentice Hall, Englewood Cliffs, New Jersey, 1984) and *Array Signal Processing* (Prentice Hall, 1993). Because of his contribution in the field of multidimensional signal processing, Dan was elected an IEEE Fellow in 1987. He is also a fellow of the National Science Foundation.

---

# Machine Intelligent Gust Front Detection

Richard L. Delanoy and Seth W. Troxel

■ Techniques of low-level machine intelligence, originally developed at Lincoln Laboratory to recognize military ground vehicles obscured by camouflage and foliage, are being used to detect gust fronts in Doppler weather radar imagery. This Machine Intelligent Gust Front Algorithm (MIGFA) is part of a suite of hazardous-weather-detection functions being developed under contract with the Federal Aviation Administration. Initially developed for use with the latest-generation Airport Surveillance Radar equipped with a wind shear processor (ASR-9 WSP), MIGFA was deployed for operational testing in Orlando, Florida, during the summer of 1992. MIGFA has demonstrated levels of detection performance that have not only markedly exceeded the capabilities of existing gust front algorithms, but are competitive with human interpreters.

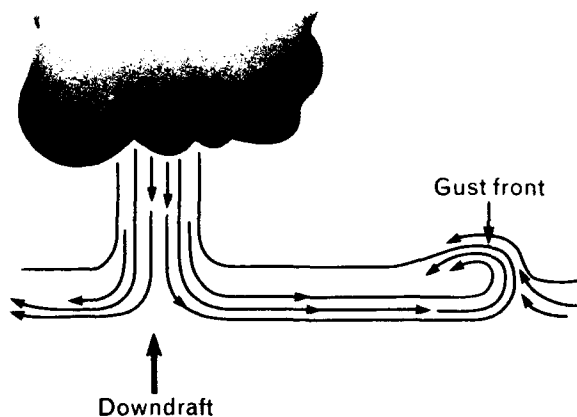
Gust fronts generated by thunderstorms can seriously affect the safety and efficiency of airport operations. Lincoln Laboratory, under contract with the Federal Aviation Administration (FAA), has had a significant role in the development of two Doppler radar systems that are capable of detecting low-altitude wind shears, including gust fronts, in the airport terminal control area. These systems are the Terminal Doppler Weather Radar (TDWR) and the latest-generation Airport Surveillance Radar enhanced with a wind shear processor (ASR-9 WSP).

By examining images generated by these radars, experienced human observers can reliably detect and track gust fronts. But the development of automated gust front detection algorithms having sufficiently high detection rates with few false alarms has been elusive. The gap between human and computer performance is due to several limitations of the detection algorithms. These limitations include the lack of means for handling and maintaining weak, ambiguous, and contradictory evidence, the use of multiple sequentially applied thresholds for object discrimination (such thresholds can inadvertently result in the discarding

of important data), a failure to use all of the relevant information available in the input data, and the ineffective use of knowledge regarding the behavior or appearance of gust fronts under different circumstances.

Given clear, unambiguous radar gust front signatures, existing detection algorithms perform reasonably well. The challenge is in constructing algorithms that can handle the marginally detectable ambiguous cases. In such cases, various factors must be considered. For example, gust fronts can be obscured by large areas of precipitation, or gust front signatures can disappear in Doppler velocity images whenever the Doppler viewing angle is perpendicular to the direction of motion. Furthermore, gust fronts can be mimicked by other natural phenomena, such as flocks of birds, clouds of dust stirred up at construction sites, low-intensity rain, and ground clutter. And gust fronts can have very low radar cross-section densities, sometimes below the sensitivity of the radar system.

The preceding paragraph should sound familiar to those involved in the development of automatic target recognition (ATR) systems, for the issues are basically the same. In addition to the continual trade-off



**FIGURE 1.** Thunderstorm downdraft and resulting gust front. The cool outflow beneath a thunderstorm spreads out in all directions. The leading edge, where the cool outflow and the warmer ambient air converge, is called the gust front.

between detection rates and numbers of false alarms, the issues for gust front detection are

1. obscuration and camouflage,
2. sensor limitations,
3. clutter and decoys, and
4. stealth.

Not surprisingly, the overall design of existing gust front detection algorithms is similar to that of most ATR systems. This traditional design is characterized by a hierarchy of modules, typically called detection, extraction (or discrimination), and classification. The detection process is essentially the application of some threshold that has been chosen to maximize the probability of detection at some acceptable level of false detections. Where signals are found that are above threshold, features are extracted, producing an abstraction, or symbolic representation, of the raw data. Given the set of extracted features, a signal is then classified as either one of the object types being sought or as clutter. In both the existing gust front detection algorithms and the traditional ATR systems, detection is generally unsophisticated: the threshold is applied either to raw radar data or to a simple transformation (such as a matched filtering) of the raw data. Sophisticated machine intelligence techniques are generally applied in the form of classifiers, e.g., by the use of neural network, statistical, or model-based classifiers.

However, the use of machine intelligence only for the classification process leads to a problem. With the application of a detection threshold, a significant amount of information is discarded, including those object signatures which are weak or ambiguous. Our belief is that increased detection capability can be achieved by applying machine intelligence techniques prior to the application of detection thresholds.

A framework for applying machine intelligence techniques at the earliest levels of signal (image) processing is provided by the Experimental Target Recognition System (XTRS) [1], a general-purpose machine intelligence approach to ATR developed at Lincoln Laboratory. Specific techniques of knowledge-based signal processing, fuzzy set theory, and pixel-level maps of spatial evidence are all part of this approach. Based on XTRS, a Machine Intelligent Gust Front Algorithm (MIGFA) has been constructed for use with both TDWR and ASR-9 WSP imagery. Of the two radar systems, the ASR-9 presents the greatest challenge to gust front detection because of its lower sensitivity and less reliable Doppler measurements in clear air. Thus, this article will focus on the ASR-9 WSP version of MIGFA to demonstrate best the algorithm's effectiveness.

### Gust Fronts

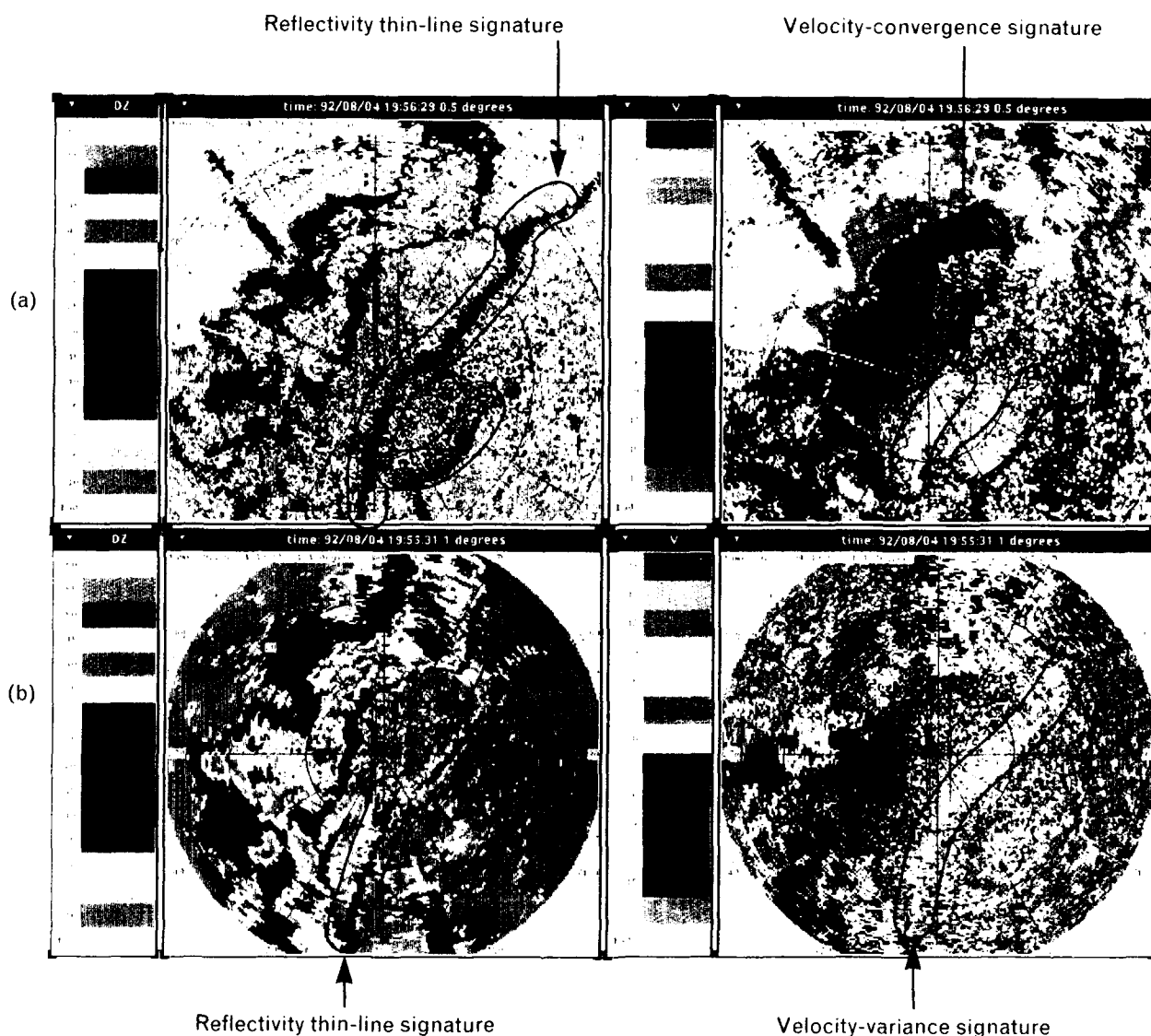
An intense thunderstorm downdraft can arise from various processes such as evaporative cooling and frictional drag between water droplets and the air. Upon impact with the ground, the downdraft is deflected horizontally (Figure 1), producing a local region of divergent winds. The downdraft feeds an outflow of outwardly expanding cool air. At the leading edge of the outflow exists a boundary where cool outflow air collides (converges) with the warmer ambient air. This leading-edge boundary, called a gust front, can grow to be many kilometers long and can propagate far away from the generating storm.

The turbulence within a gust front can be severe enough to present a danger to aircraft during takeoff and landing. And, because the prevailing winds behind a gust front can persist for a long time, the passage of a gust front over an airport often necessitates a change of active runway. When unanticipated, a gust front can delay airport operations as aircraft are

rerouted to a different runway. Aside from issues of cost and inconvenience, delays can increase the risk of potentially fatal human errors as the distance between aircraft that are taking off or landing decreases and the work load on air traffic controllers increases. With sufficient warning, though, controllers can incorporate in their plans a change in active runway at the anticipated time of a gust front's arrival, thereby minimizing the hazards and costs associated with delays.

Gust fronts can be detected in Doppler radar imagery on the basis of three physical properties: velocity convergence, thin lines, and motion. Figure 2 shows a typical gust front in both TDWR and ASR-9 WSP images.

The air within and behind a gust front converges with the ambient air ahead of the gust front. In a Doppler velocity image, this activity is observable as a boundary between regions of converging velocities. When viewed along a single radial, the convergence

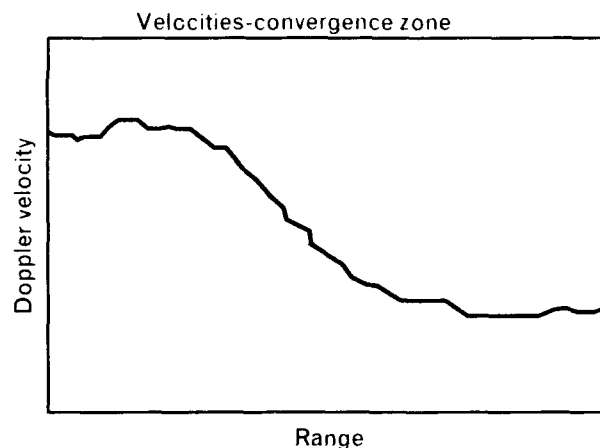


**FIGURE 2.** An example gust front in (a) TDWR and (b) ASR-9 WSP images. The left radar plots are reflectivity images with units in dBZ. The right radar plots are Doppler images with units in m/sec. The different signatures (see main text) of the gust front have been indicated by a human interpreter.

signature is characterized by a relatively sharp decrease in radial Doppler values with distance (Figure 3). Because Doppler radars can measure only the component of the wind that is directed along the beam, Doppler velocity measurements can often underestimate the true wind speed. In the extreme, the convergence signature of a gust front disappears completely when the direction of motion is perpendicular to the radar beam azimuth. The TDWR velocity image shown in Figure 2(a) demonstrates this problem. In the figure, the portion of the front closest to the radar site has a direction of motion that is nearly radially aligned, resulting in a pronounced convergent boundary for that area. However, at the ends of the gust front, where the direction of motion is more azimuthal, the boundary is more difficult to detect.

The thin-line signature is generally thought to be produced by the concentration of scatterers (dust, insects, rain droplets) along the leading edge of the thunderstorm outflow. Some gust fronts produce a distinctive cloud formation along the gust front, which can also contribute to the thin-line reflectivity. The thin line varies in width but seldom exceeds 3 km. Typical maximum reflectivities reported by the ASR-9 along gust fronts are in the range of 10 to 20 dBZ. But significant portions of many thin lines can have reflectivities as low as  $-5$  dBZ, which is near or below the threshold of detectability for the ASR-9. (Note: The basic unit of measurement for radar reflectivity is dBZ. Reflectivities of 50 dBZ or more are typical of intense thunderstorms with heavy rain. Background typically has reflectivity values between  $-15$  and 0 dBZ.) Because of ground-clutter obscuration, the quality of a thin-line signature often degrades at close range, and the signature can even vanish as the gust front passes over the radar. As the front moves out of the cluttered region, the signature often reestablishes itself. This type of degradation is especially troublesome for the ASR-9 because of the radar's on-airport location, which makes it more prone to detection loss when a gust front is affecting the airport.

A final key gust front signature is motion. When sequential radar scans are compared, convergence and thin-line signatures of a gust front will move conspicuously in a direction perpendicular to the orientation of the convergence boundary and reflectivity



**FIGURE 3.** Example velocity-convergence signature associated with a gust front.

thin line. Signatures that do not move are either not gust fronts—e.g., they could be false alarms from range-ambiguous echoes (discussed in the subsection "Feature Detection"), edges of storm regions, or ground clutter—or they are gust fronts that are not operationally significant. Within limits, gust fronts tend to move uniformly as outwardly expanding curved boundaries; i.e., the propagation speed tends to be consistent along the front's length and across time. Of course, when gust fronts collide, the motion may become more erratic.

If these signatures were 100% reliable, detection would be a trivial task. For some gust fronts, however, one or more signatures may be weak, ambiguous, or entirely absent. For example, convergence signatures disappear when the radar beam is perpendicular to the wind velocity. Reflectivity thin lines and thin-line motion can disappear when a gust front is obscured by storm regions. To complicate matters further, none of these signatures are unique to gust fronts. Vertical shears, often present in severe thunderstorms, can bias low-altitude velocity estimates, producing apparent convergence signatures. Range-ambiguous echoes, ground clutter, flocks of birds, and elongated patches of low-intensity precipitation can all appear as reflectivity thin lines. Motion can be associated with anything (e.g., clouds or airborne dust) that follows the ambient wind. In short, each signature can be missing and each signature can be mimicked by other observable phenomena. Consequently, suc-

cessful discrimination requires knowledge of the circumstances for which these signatures are reliable as well as knowledge of gust front behavior. Only by weighing the quality of several signatures simultaneously can an automated system detect gust fronts with near human performance.

The task is difficult enough with TDWR data. And yet the TDWR is a pencil-beam radar, designed for weather sensing, with enough sensitivity to generate reliable Doppler values in relatively clear air and enough resolution in elevation to provide three-dimensional images of weather phenomena. In contrast, the ASR-9 is a surveillance radar that was not originally intended for weather imaging. With a fan-beam design, the ASR-9 vertically integrates signals into a single two-dimensional representation. Because the transmitted energy is distributed over a wider arc of elevation, the energy returned from a low-altitude, low-reflectivity gust front will be small relative to the energy filling the remainder of the sample volume. With this reduced sensitivity, gust front detection is much more difficult. Almost all convergence signatures are eliminated for the ASR-9 because the Doppler values are unreliable since the reflectivity returns from clear air are below the threshold of detectability for the radar. Even for cases in which gust fronts pass through regions of high reflectivity, convergence cannot be used reliably for gust front detection. For example, the signal contribution from overhanging precipitation near the edges of storms can bias the low-level wind-velocity estimate when there is vertical wind shear. Without convergence signatures, thin line and thin-line motion become the primary signatures for detecting gust fronts in ASR-9 WSP imagery. In the example ASR-9 WSP reflectivity image of Figure 2(b), the gust front is visible. But note that while the TDWR thin line is quite strong, the ASR-9 WSP thin line shows less contrast, is somewhat more fragmented, and does not extend as far as is apparent in the TDWR data.

Although a convergence signature is missing from the ASR-9 WSP velocity image of Figure 2, the gust front is still visible. The accuracy of velocity estimations degrades markedly over the range of signal-to-noise values associated with low reflectivity returns. For this reason, gust fronts are observable in ASR-9

WSP velocity images as bands of low-variance Doppler values, with high variance in the low signal-to-noise regions ahead and behind the gust front. This velocity-variance thin line is an alternative signature used in the ASR-9 WSP version of MIGFA. In addition, implicit zones of convergence can be identified. Doppler values within the gust front thin line are used to estimate winds behind the gust front. The environmental low-level wind velocity ahead of the storm can be measured by some other means—for example, from a network of anemometers at the airport. A comparison of these two wind-velocity estimates can be used to confirm that convergence exists somewhere between the gust front and the anemometer site.

### Background

Automated radar gust front detection algorithms have been under development and evolution for almost ten years. H. Uyeda and D. Zrnić [2] first described an automated detection algorithm, developed for the Next Generation Weather Radar (NEXRAD), that was based solely on detecting velocity convergence along radials. The algorithm was successful in locating and tracking the strong gust fronts that commonly occur in Oklahoma during the spring.

An improved version of the initial algorithm reduces false alarms by requiring vertical association of gust front signatures from two different low-altitude elevation scans. The improved algorithm, known as the Gust Front Detection Algorithm (GFDA), also incorporates a technique for estimating horizontal winds ahead and behind detected gust fronts [3, 4]. As with its predecessor, GFDA detects velocity convergence along radials. GFDA is the algorithm currently intended for use in the initial operational deployment of TDWR systems.

Briefly described, GFDA begins with a search in each radial for runs, or segments, of decreasing radial velocity, indicating convergent shear. Segments in which the maximum shear exceeds a predetermined threshold are logically grouped into features on the basis of end-point-proximity and segment-overlap tests. The feature attributes are then tested against a number of thresholds and are kept, discarded, or combined with other features. After separately pro-

cessing each of the two full-circle scans from different altitudes, the algorithm tests for vertical continuity of the features between the scans. Features that exhibit vertical continuity and that exceed a minimum-length threshold are declared to be gust fronts. The reported location of the detected gust front is determined by fitting a curved line through the peak shear of each segment in the gust front feature. Sequential detections are associated over time to build detection histories for each gust front upon which propagation speeds are estimated and forecasts generated.

Lincoln Laboratory, in conjunction with the National Severe Storm Laboratory (NSSL), has since developed the Advanced Gust Front Algorithm (AGFA) [5, 6], which contains several enhancements, including reflectivity thin-line detection. AGFA detects thin lines by finding local maxima of reflectivity values that are consistent with the widths and intensities associated with gust fronts. Thin-line segments are generated twice: once by constructing segments over all range gates along a radial and once by constructing segments across radials along arcs of constant range. The final thin-line features consist of lists of the points connecting the centers of each of the segments. Convergence and thin-line features are fused on the basis of end-point proximity and orientation. AGFA does not use motion as a signature for detecting gust fronts. Motion is used only in heuristics that reject false features after they have been extracted.

During field testing in 1990 and 1991, a customized version of AGFA was used for gust front detection on an ASR-9 WSP [7, 8]. Because of the lack of reliable velocity-convergence features, the ASR-9 version of AGFA was configured to operate in a thin-line-only detection mode. Although the algorithm was successful in detecting gust fronts that had thin-line signatures of good quality, it had some difficulty detecting gust fronts when the reflectivity was weak or fragmented. Lacking convergence signatures to confirm the existence of gust fronts, the algorithm was prone to false alarms triggered by elongated low-reflectivity weather echoes that are sometimes associated with stratiform rain. Installing suboptimal detection thresholds to reduce the false-alarm rate further reduced the detection probabilities.

In the above study, the scoring was done against

human interpretations of the same images used as input to the algorithm. The discrepancy between human and AGFA performance appears to be partially due to AGFA's not making full use of a variety of additional information that is available in the ASR-9 WSP data, including velocity thin lines and thin-line motion. Moreover, both GFDA and AGFA rely on sequentially applied thresholds to discriminate gust fronts from background. When the relevant signals are weak or ambiguous, the use of thresholds in the early stages of processing can result in the elimination of potentially relevant information, thus setting unnecessary limits on detection performance. GFDA and AGFA also rely on one-dimensional signal processing operations to locate gust fronts. The extraction of chains of points across the second dimension is done at a higher, heuristic level of processing. In contrast, two-dimensional signal processing operations can directly establish the shape of gust fronts without relying on heuristics. Finally, these early gust front algorithms have no systematic means of conditionally fusing information from various sources by taking into account the different reliabilities of the sources. Different signatures can have varying reliability depending on the situational context.

### **Low-Level Machine Intelligence**

The conventional wisdom in computer vision/object recognition research has been to use general image processing operations, ideally devoid of object- and context-dependent knowledge, at the initial stages of processing. Such operations might include edge detection, segmentation, cleaning, and motion analysis. And yet the ideal has never really been achieved in practice. For example, some knowledge of the sensor and the expected scene contents must be implicitly encoded in the form of thresholds or other similar parameters to detect edges effectively.

From the results of such general operations, image characteristics are extracted and represented symbolically. Machine intelligence is then applied, as if by definition, only on the symbolic representations at higher levels of processing.

MIGFA has inherited the development environment, control structure, knowledge-based signal processing, and several other important attributes of

XTRS. In contrast to more conventional approaches to object recognition, sensor-, object-, and context-dependent knowledge is applied in the earliest levels of processing, i.e., at the image processing stage. As used in MIGFA, low-level machine intelligence applies knowledge in three ways.

First, knowledge of the current environment is used to choose from a library those feature detectors which are *selectively indicative* of the object being sought. Using multiple independent feature detectors, MIGFA can adapt to different contextual circumstances. At the beginning of the processing of each scan, a rule-based expert examines contextual information to select a set of feature detectors known through experience to be the most effective for a given set of circumstances. In the extreme, this process would enable MIGFA to adapt itself dynamically to changes in the environment. Currently, the only rule used by MIGFA selects between two fixed alternative sets of feature detectors, one set customized for the TDWR and the other customized for the ASR-9 WSP. Because of the redundancy inherent in the use of multiple feature detectors, MIGFA tends to be robust: the malfunction of a feature detector or even the absence of one data source does not necessarily halt processing and may have only minor effects on detection performance.

Second, knowledge is also incorporated within feature detectors through the design of matched filters that are customized to the physical properties of the sensor, the environment, and the object to be detected. A new technique of knowledge-based signal processing, called *functional template correlation* (FTC), allows the construction of customized signal processing operations that are more effective than standard operations (see the box, "Functional Template Correlation"). The output of FTC is a map of numeric values in the range [0,1] that indicate the degree of match between the pattern of pixels in an image region and the feature or object encoded in the functional template.

Finally, knowledge of the varying reliabilities of the selected feature detectors is used to guide data fusion and extraction. Conditional data fusion is simplified by using "interest" as a common denominator [9]. An interest image is a spatial map of evidence for the

presence of some feature that is selectively indicative of an object being sought (the output of FTC is an interest image as long as the functional template encodes an indicative feature). Higher pixel values reflect greater confidence that the intended feature is present at that location. Using interest as a common denominator, MIGFA fuses data by combining interest images derived from various pixel-registered sensory sources. Using simple or arbitrarily complex rules of arithmetic, fuzzy logic, or statistics, MIGFA can assimilate pixel-level evidence from several coregistered sources into a single combined interest image. Clusters of high values in such combined interest images are then used to guide selective attention and can serve as the input for object extraction. If done effectively, the combined interest image provides a better representation of object shape than is evident in any single sensor modality. Using these techniques, MIGFA performs a significant amount of knowledge-based processing before the application of the first discriminating threshold. Most traditional perception systems apply one or several thresholds early in the processing as a way of quickly reducing the amount of data to be processed. However, especially with ambiguous data, each applied threshold closes off options for detecting an object. A better strategy—a strategy attempted in XTRS and MIGFA—is to apply thresholds only after evidence from many sources of information have been meaningfully fused into a single map of evidence.

### MIGFA Design

The system block diagram in Figure 4 is an overview of MIGFA as configured for ASR-9 WSP data. In preparation for processing, input images V (Doppler velocity) and DZ (reflectivity) from the current radar scan are converted from polar to Cartesian representation and scaled to a useful resolution. Image SD is a map of the local standard deviations of V values. The SD and DZ images are then passed to multiple simple independent feature detectors that attempt to localize those features which are selectively indicative of gust fronts. The outputs of each of these feature detectors, most of which are based on some application of FTC, are expressed as interest images that specify evidence indicating where and with what confidence a gust front may be present. The different interest images



## FUNCTIONAL TEMPLATE CORRELATION

FUNCTIONAL TEMPLATE correlation (FTC) [1, 2] is a generalized matched filter that incorporates aspects of fuzzy set theory. Consider, as a basis for understanding, the basic image processing tool *autocorrelation*. Given some input image  $I$ , an output image  $O$  is generated by matching a kernel  $K$  against the local neighborhood centered on each pixel location  $I_{xy}$ . The match score assigned to each pixel  $O_{xy}$  is computed by multiplying each element value of  $K$  by the superimposed element value in  $I$  and summing across all products. If the shape to be matched can vary in orientation, then the pixel  $I_{xy}$  is probed by  $K$  at multiple orientations. The score assigned to  $O_{xy}$  is the maximum across all orientations.

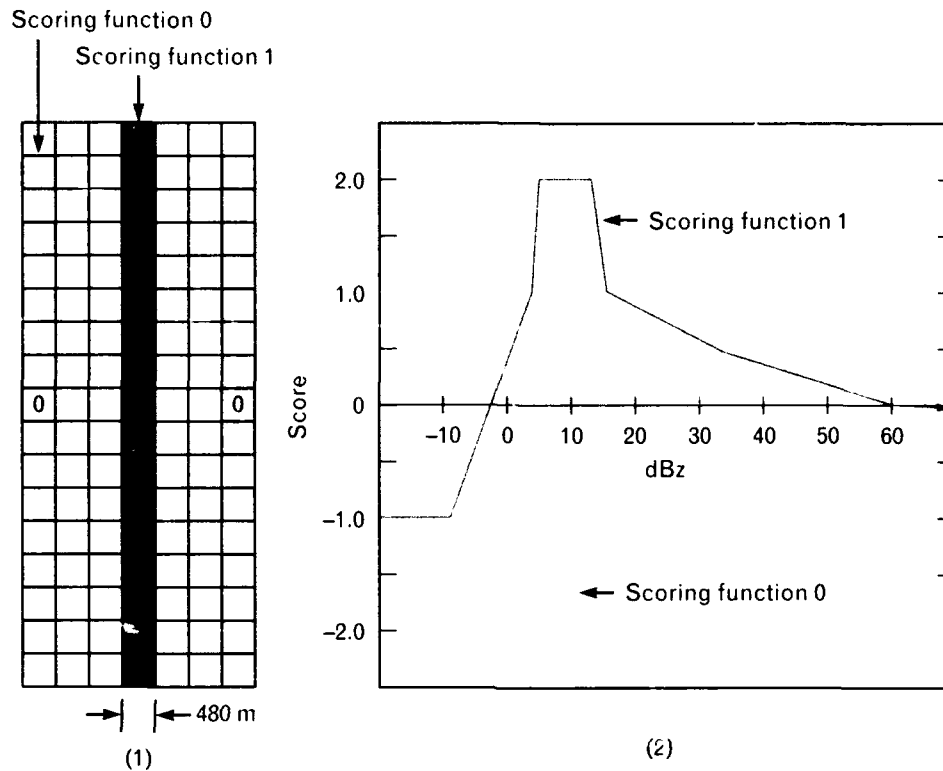
FTC is fundamentally the same operation with one important exception: whereas the kernel used in autocorrelation is an array of image values (the array is essentially a subimage of the image to be probed), the kernel used in FTC is an array of *scoring functions*. The scoring functions return scores that indicate how well the image values match the expectations of the values at each element of the kernel. The set of all returned scores are averaged and "clipped" to the continuous range [0,1]. (In the clipping process, those averaged scores which are less than zero are assigned a

value of zero while those averaged scores which are greater than one are assigned a value of one.) The output of FTC is a map of these values, each of which reflects the degree that the shape or object implicitly encoded in the functional template is present at that image location.

Consider as an example the functional template implementation of a simple matched filter designed to detect gust fronts in reflectivity data (Figure A). Gust fronts are observed as thin lines of moderate reflectivity (approximately 0 to 20 dBZ) that are flanked on both sides by low reflectivity (approximately -15 to 0 dBZ). Figure A(1) shows the template kernel consisting of integers that correspond to the two scoring functions shown in Figure A(2). Elements of the kernel that do not correspond to either of the scoring functions form guard regions in which image (i.e., reflectivity) values are ignored and have no effect on match scores. Scoring function 0, corresponding to the flanking regions of low reflectivity, returns a maximal score of 1.0 for image values in the interval of -20 to -5 dBZ, a gradually decreasing score for image values in the interval -5 to 10 dBZ, and a score of -2.0 for image values larger than 10 dBZ. Scoring function 1, corresponding to the center of the

kernel where moderate reflectivity values are expected, returns maximal scores in the interval 5 to 12.5 dBZ and gradually decreasing scores for both higher and lower image values. Note that although very low image values can generate scores of -1.0, a slower decline in score with a minimum score of 0.0 is returned for image values above the maximal scoring interval. This asymmetry is an attempt to mitigate the obscuring effects of storm regions and other patches of high reflectivity.

In general, by increasing or decreasing the intervals over which affirming scores (i.e., scores > 0.5) are returned, scoring functions can encode varying degrees of uncertainty with regard to which image values are allowable. In addition, knowledge of how a feature or object appears in sensor imagery can be encoded in scoring functions. The interfering effects of occlusion, distortion, noise, and clutter can be minimized by the use of various design strategies [3]. As a consequence, matched filters customized with FTC for specific applications are generally more robust than classical signal processing operations. In the thin-line matched-filter example shown in Figure A, the filter does not simply find thin lines, but selects those thin lines which have reflectivity values within a particu-



**FIGURE A.** Example functional template for thin-line feature detection: (1) index kernel and (2) corresponding scoring functions. By increasing or decreasing the intervals over which affirming scores (i.e., scores  $> 0.5$ ) are returned, scoring functions can encode varying degrees of uncertainty with regard to which image values are allowable. In addition, knowledge of how a feature or object appears in sensor imagery can be encoded in scoring functions.

lar range. Furthermore, the matched filter can display differential tolerances to image values that are higher or lower than the expected range of values. In the automatic target recognition (ATR) systems developed at Lincoln Laboratory, FTC has been used primarily as a direct one-step means of three-dimensional object detection and extraction. In the Machine Intelligent Gust Front Algorithm (MIGFA), FTC is used more as a signal processing tool for edge detec-

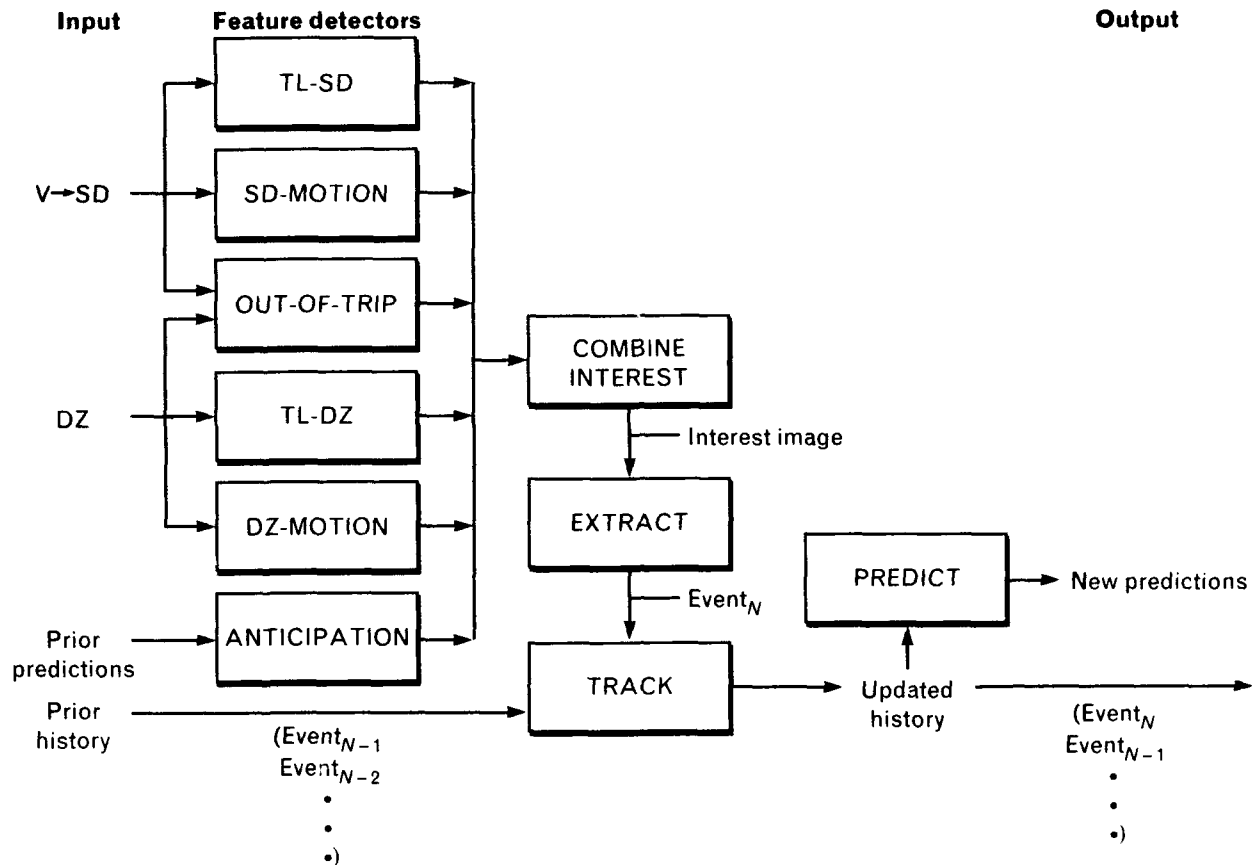
tion, thin-line filtering and smoothing, shape matching, skeletonizing, and erosion.

If FTC were implemented literally as described here, the computational expense would be prohibitive for most useful tasks. But FTC is actually faster than autocorrelation if the input data are scaled to a fixed integer range (e.g., 0 to 255) and the scoring functions are implemented as a precomputed two-dimensional lookup table that is indexed by a scoring-function number and

an image value.

#### References

1. R.L. Delaney, J.G. Verly, and D.E. Dudgeon, "Functional Templates and Their Application to 3-D Object Recognition," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, Mar. 1992.
2. R.L. Delaney and J.G. Verly, "Computer Apparatus and Method for Fuzzy Template Shape Matching Using a Scoring Function," U.S. Patent No. 5,222,155 (June 1993).
3. R.L. Delaney, J.G. Verly, and D.E. Dudgeon, "Pixel-Level Fusion Using Interest Images," *Technical Report 979*, MIT Lincoln Laboratory (26 Apr. 1993).



**FIGURE 4.** Block diagram of the Machine Intelligent Gust Front Algorithm (MIGFA). For a description of the different feature detectors, see the subsection "Feature Detection" in the main text.

are fused to form a combined interest image, thus providing an overall map of evidence indicating the locations of possible gust fronts.

From the combined interest image, fronts are extracted as chains of points. The chains extracted from a radar scan, collectively called an event, are integrated with prior events by establishing a point-to-point correspondence. Heuristics are then applied to reject those chain points which have an apparent motion that is improbable. The updated history is used to make predictions of where points along the front will be located at some future time. Such predictions are used in the processing of subsequent images, specifically in the feature detector called ANTICIPATION. In the output of ANTICIPATION, high interest values are placed wherever fronts are expected to be, thereby selectively sensitizing the system to detect gust fronts at specif-

ic locations. ANTICIPATION is tuned so that it will not automatically trigger a detection by itself but, when its output is averaged with other interest images, it will support weak evidence that would otherwise be insufficient to trigger a detection. Figure 5 is a summary of the processing steps for an example ASR-9 WSP scan.

#### Image Preparation

As discussed earlier, velocity convergence is an unreliable signature for detecting gust fronts in ASR-9 WSP data. Gust fronts, nevertheless, are visible in velocity images. Because of the tendency for high-pass clutter-filtered pulse-pair Doppler estimates in a velocity image to have high variance in regions of low signal-to-noise ratios (SNR), the local velocity variance is higher for an area of clear air than for an area associated with slightly higher reflectivity values. This information is

translated into a usable form by transforming the velocity image  $V$  into a map of local standard deviations (the SD image). At each pixel of  $V$ , the standard deviation was computed in the surrounding  $5 \times 5$  pixel neighborhood and assigned to the corresponding pixel in SD.

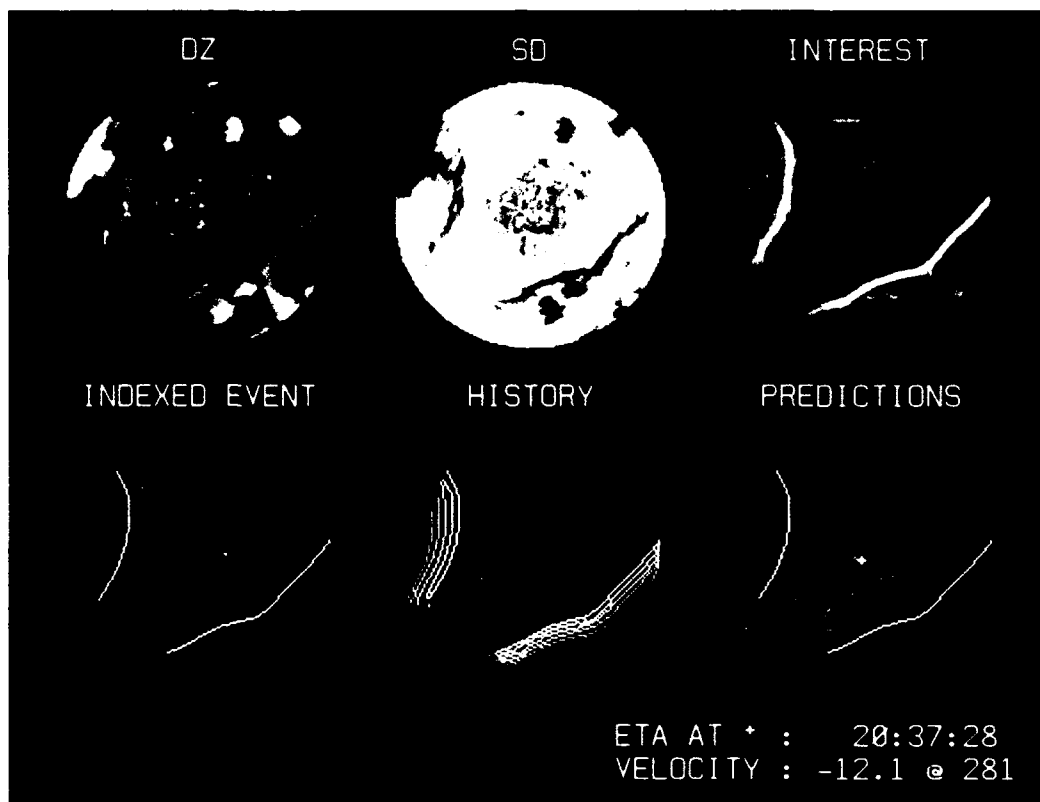
Pixel values for all images are scaled to the interval 0 to 255 to support subsequent FTC operations on the input imagery. Each image is tagged with the scaling factor and offset necessary to translate scaled values back to the original physical values.

Finally, the DZ and SD images are converted from polar arrays (240 range bins  $\times$  256 radials) to Cartesian arrays (130  $\times$  130). Mapping is done by

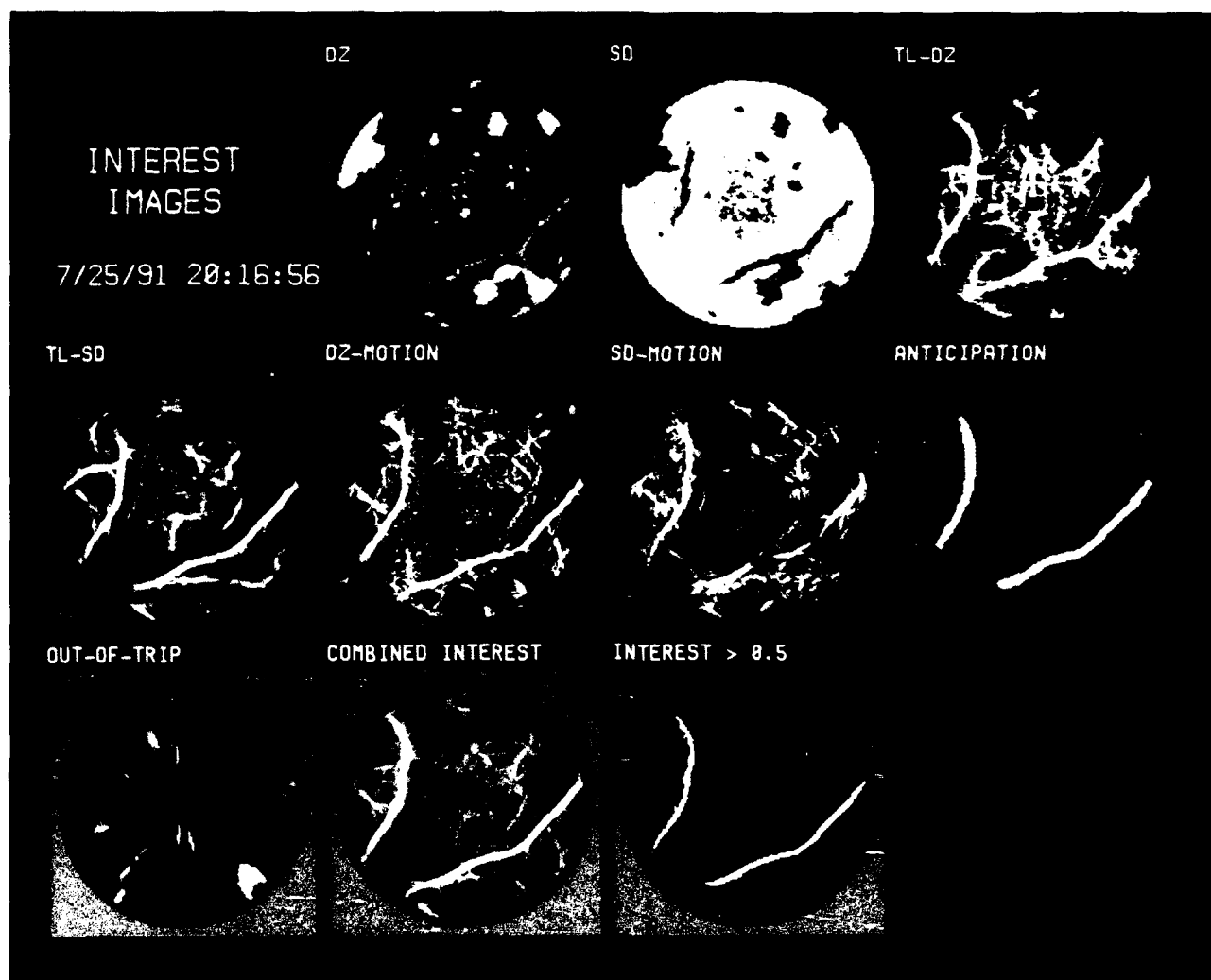
computing for each element of the Cartesian array the range bin and radial at which the corresponding value is to be found in the polar array. During the mapping process, an implicit subsampling of the data occurs. From an initial radial resolution of 120 m per range bin and pixel size in the azimuthal dimension decreasing from 680 m at 28 km, the final Cartesian image has a pixel resolution of 480 m per pixel.

#### *Feature Detection*

Given contextual information of the sensor being used, the location of that sensor, and the environmental conditions, a rule-based expert selects an appropriate set of feature detectors for application to the input



**FIGURE 5.** Processed scan summary. In the first row are the DZ (reflectivity) image, SD (standard deviation of velocity) image, and the combined interest image that has been computed from the DZ and SD images. The second row begins with the extracted indexed event. White pixels are those points which have been declared as part of a gust front. Gray pixels are those points which have not been tracked long enough to establish sufficient confidence. In the history frame, the current chain is shown in white and the preceding scans are shown in shades of gray (darker shades indicate more distant events in time). In the predictions frame, fat gray pixels indicate the 10- and 20-min forecasts of where the fronts are expected to be. Also shown at the bottom right corner are the estimated time of arrival of the next gust front to cross the radar site, the speed of the winds measured inside the front (in m/sec), and the direction (in degrees) from which the front is coming.

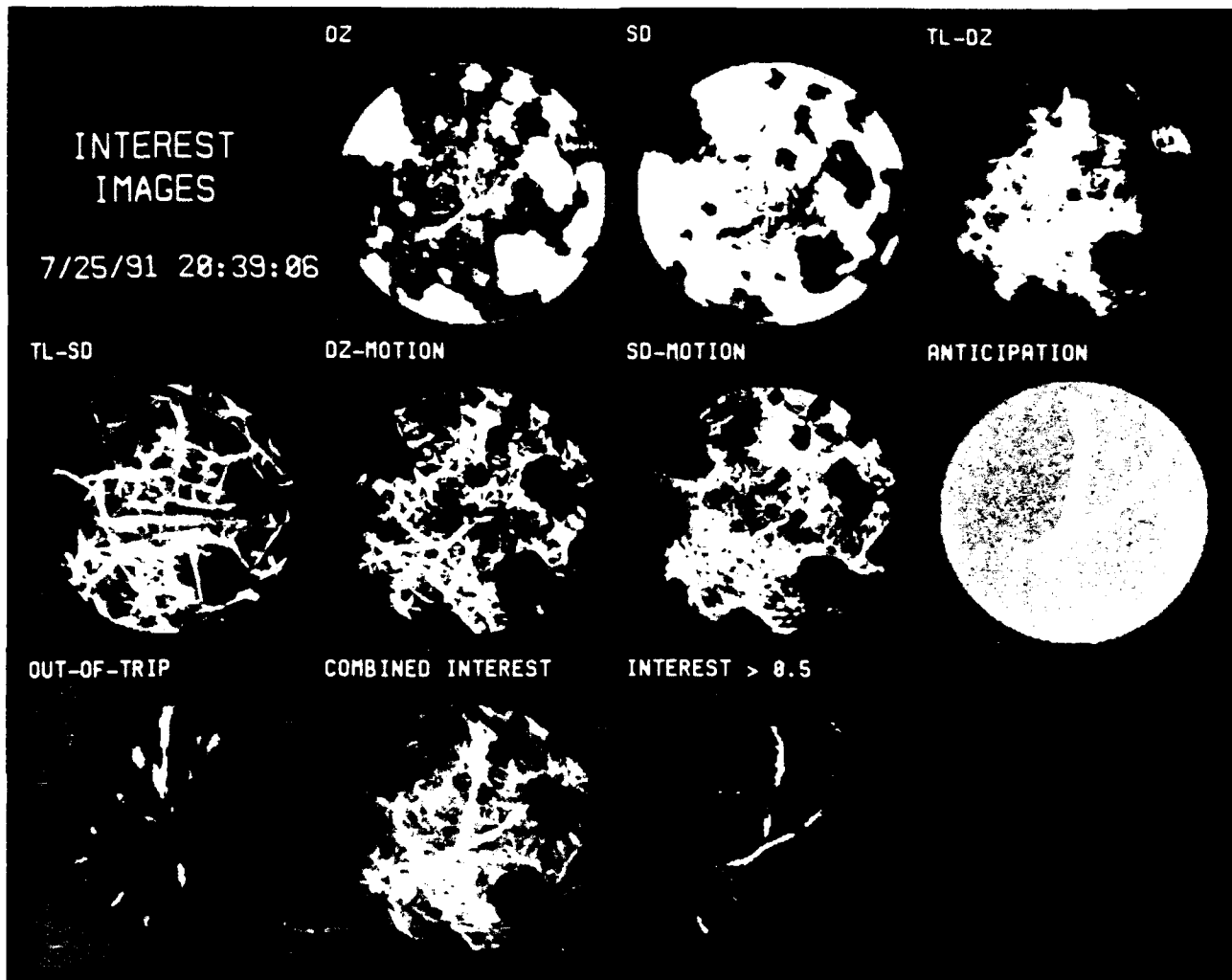


**FIGURE 6.** Combining interest images—satellite video. An example DZ and SD image of two coast fronts are shown, along with the output images of the different feature detectors (cf. Figure 4). Regions of test pixels indicate areas where specific feature detectors have not expressed an opinion regarding the presence of a coast front (detected or not). The regions are generated by other feature detectors. The last two frames (the center and right frames of the third row) show the images that resulted from combining the different interest images.

data. For the moment, the only rule that is available chooses between two sets of feature detectors: one set customized for the ASR-9 WSP, the other for the TDWR. These two sets alone may be sufficient. As MIGA-Arresting is expanded, however, additional rules adapting the feature detector set for sites other than central Florida and seasons other than midsummer may be added as required. Although we do not anticipate the need, the set of feature detectors could be designed to adapt dynamically from one set to the next as changes in weather conditions are detected. The following is an example of the output of the

were tested for the ASR-9 (p. 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100).

TL-DZ and TL-SD (Figure 6) are two simple thin-line feature detectors that are used for the input images DZ and SD, respectively. The TL-DZ feature detector is based primarily on the application of the functional template illustrated in Figure 5 of the box "Functional Template Correlation." An example DZ image and the interest image generated by TL-DZ and other feature detectors are shown in Figure 6a and 6b. Although the thin-line template is normally applied to SD, as the example of the interest image TL-DZ shows, it can also be applied to the DZ image.



**FIGURE 7.** Combining interest images: weak evidence. This figure is similar to Figure 6 except that here, the two gust fronts are not clearly visible in any of the single interest images except for the anticipation image. In the combined interest images, however, the gust fronts are much more apparent, illustrating how the fusion of weak evidence from multiple sources enhances gust front detectability.

associated with gust fronts have low Doppler velocity standard-deviation values within the front and high values ahead and behind the front. Consequently, the scoring function for the center strip returns maximal scores for low values while the scoring function for the flanking regions returns maximal scores for high values.

DZ-MOTION and SD-MOTION are two thin-line motion detectors that are very similar to the basic thin-line detectors TL-DZ and TL-SD. The detection of motion is based on simple differencing. For example, in DZ-MOTION the DZ image from a

previous scan produced approximately 4 min before the current scan is subtracted from the current DZ image. In the differenced DZ image, gust fronts appear as white lines (positive values at the front's position in the current scan) that are trailed by parallel dark lines (negative values at the front's position in the previous scan). Although functional templates that can scan for parallel white and dark thin lines simultaneously are feasible, these types of templates have so far proven to be too computationally expensive to operate within the real time constraints of the available computer resources. Thus the existing

DZ-MOTION simply looks for thin lines of positive values. The functional template used has a kernel that is identical to the one shown in Figure A of the box, "Functional Template Correlation," but the scoring functions are somewhat different because of the consequences of differencing. The feature-detector SD-MOTION, is similar to DZ-MOTION in that SD-MOTION also applies to the difference of two sequential images a thin-line filter with customized scoring functions. With this approach, thin lines that do not move are given low interest values, reflecting the belief that a stationary thin line is either not a gust front or is a gust front that may be ignored. Because the background in differenced images is reduced to values near zero, DZ-MOTION and SD-MOTION tend to be more sensitive than TL-DZ and TL-SD.

One disadvantage of DZ-MOTION and SD-MOTION is that they tend to produce false alarms when moving storms are present because the leading edge of the storm may appear in the differenced image as a thin line of positive values. For reducing the likelihood of such false alarms, an image of storm regions is generated with a round functional template whose kernel has a diameter of 13 pixels (6.25 km). Wherever storm regions are detected with this template, interest values are decreased in DZ-MOTION and set to nil (i.e., no opinion) in SD-MOTION.

A fifth feature detector, OUT-OF-TRIP, highlights range-ambiguous echoes. Range-ambiguous echoes occur when signals are reflected by weather more distant than the maximum unambiguous range. Because the signals have traveled farther, they arrive back at the radar receiver at the same time as signals that are transmitted later and reflected from nearer weather (hence the name OUT-OF-TRIP). For these range-ambiguous echoes, the apparent range extent is maintained while the azimuthal extent is reduced proportional to the range; thus the signals have a distinctive appearance as reflectivity thin lines that are radially aligned and that are associated with high local variance in the Doppler data. Because of their thin-line appearance, range-ambiguous echoes are often inappropriately given high interest values by both TL-DZ and DZ-MOTION.

The detection of out-of-trip signals is performed by applying two functional templates simultaneously.

One template looks for radially aligned thin lines in the DZ image, while the other requires that the corresponding SD values are high. The result is an interest image that highlights out-of-trip signals. After the combination of all other interest images, the out-of-trip interest image is subtracted from the combined interest image, thus selectively suppressing evidence for the presence of gust fronts where out-of-trip signals are found. Example outputs of the OUT-OF-TRIP feature detector are shown in Figures 6 and 7.

The ANTICIPATION feature detector provides a mechanism, based on situational context, for spatially adjusting the detection sensitivity of MIGFA. High anticipation values get averaged with interest values from other feature detectors to increase the likelihood of detection at specific locations. Similarly, low anticipation values suppress the likelihood of detection.

The most important use of anticipation is as a replacement for *coasting*. Simply defined, coasting is the continued tracking of a target on a radar screen for some time interval after the target has disappeared (i.e., after the target's signal has fallen below some detection threshold). Coasting assumes that the loss of a target's signal is not due to a change in the target's behavior (e.g., a change in velocity or perhaps the disappearance of the target). Gust fronts, however, do change their behavior, as in cases in which gust fronts collide. Consequently, the blind coasting of a signal after the signal's loss is a potential source of false alarms. As an alternative to blind coasting, anticipation provides a mechanism for progressively increasing the sensitivity of a detection system, supporting weak evidence that would otherwise fall below detection thresholds.

In MIGFA, prior history of the behavior of a particular gust front is used to predict where that front is expected to be in the current scan. The predictions are used to create a band of elevated interest values, typically not so high as to trigger a detection by themselves, but high enough to raise collocated weak signals above threshold. In general, as the length of time a gust front has been tracked increases, the anticipation interest values can also be increased. If absolute coasting is desired, interest values can be increased to a level high enough to trigger a detection

without any other supporting evidence. Examples of anticipation interest images are shown in Figures 6 and 7.

Anticipation can also be used to adjust the sensitivity of gust front detections on the basis of contextual knowledge. Some examples follow:

1. Many gust fronts are not observable in radar data when the fronts are directly over the radar site because of obscuration by intense ground clutter. Even with anticipation of where a gust front is expected to be, the radar system can often lose the front as the front crosses over the radar site. To prevent such a loss, absolute coasting over the radar site can be accomplished by setting interest values within 2 km of the radar site to nil (i.e., missing values) for all interest images *except* the anticipation image. Consequently, the anticipation interest image will be the only image allowed to have an opinion of what exists directly over the radar site.
2. Gust front false alarms often occur from thin, elongated bands of low-reflectivity stratiform rain. In central Florida at least, gust fronts are seldom associated with the stratiform rain that often follows intense storm activity. Hence, under such conditions, the ANTICIPATION feature detector suppresses the background anticipation interest values.
3. False alarms are rare in the absence of any precipitation. Thus, when no precipitation is visible on the radar screen, the background anticipation interest values may be safely raised, thereby increasing the likelihood of detecting an incoming gust front that is generated by a more distant storm.

### *Combining Evidence*

During the feature-detector selection process, a *rule of combination* is also chosen to govern the combining of evidence—an example of data fusion. In principle, the rule of combination can be as simple as the averaging of pixel values across all interest images. However, for the set of ASR-9 WSP feature detectors described earlier, a somewhat more complicated rule has been used.

The four interest images generated by TL-DZ,

TL-SD, DZ-MOTION, and SD-MOTION are averaged together. During the process, any missing values are ignored. The resulting averaged interest image and the anticipation interest image are combined as a weighted average: the average of the first four interest images is given a weight of 0.75 while the anticipation image is given a weight of 0.25. Finally, elements of the out-of-trip interest image are multiplied by 0.25 and subtracted from the elements of the weighted average. The resulting image is called the combined interest image.

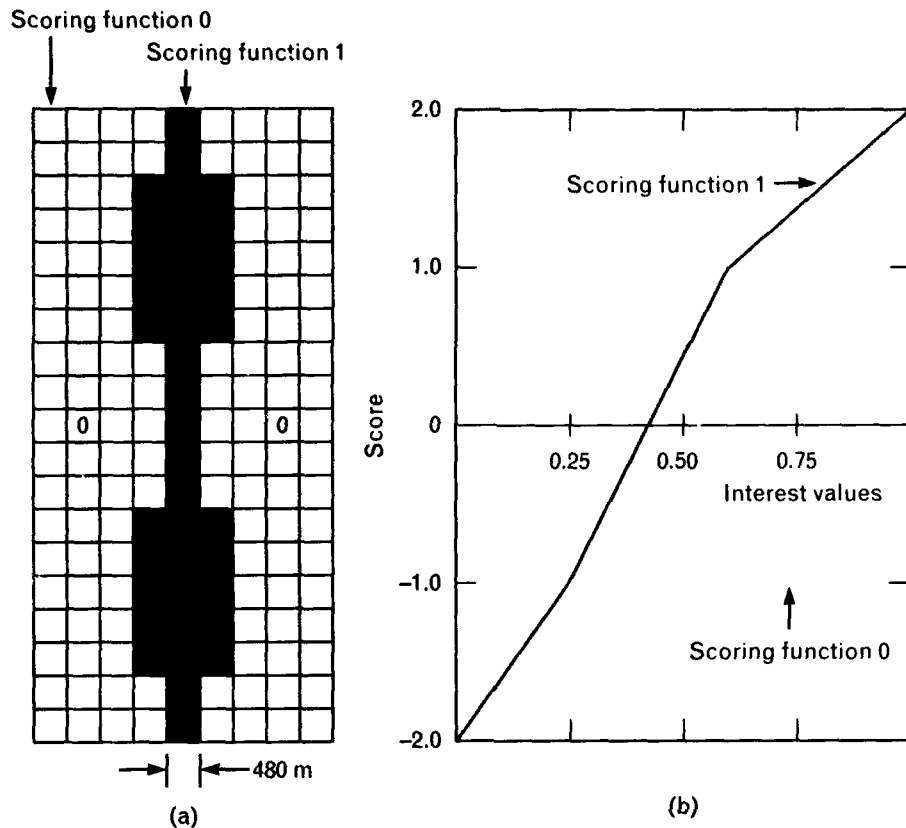
Figure 6 shows an example ASR-9 WSP DZ image, the outputs of each feature detector, and the final interest image. In this case, strong evidence for the two fronts is visible in each of the component interest images (except, of course, for the out-of-trip image). Clearly, any one of the feature detectors acting alone would have been adequate. Now consider Figure 7, which summarizes the evidence for the presence of the two gust fronts in a later scan in which detection has become more difficult as accumulating storm regions have occluded the fronts. Note that although different parts of the gust fronts are highlighted in different interest images, the gust fronts are not unambiguously visible in any single interest image (except the anticipation image). In the combined interest image, however, the gust fronts are much more apparent. This example illustrates how evidence derived from multiple feature detectors can be combined so that the various detectors mutually support and compensate for one other.

In MIGFA, no one feature detector is meant to be a perfect, or even necessarily a good, discriminator of gust fronts and background. When used together, however, several weakly discriminating feature detectors can achieve robust performance depending on how the detector outputs are combined.

### *Extraction*

Algorithms, such as AGFA, that track gust fronts as entities must identify gust fronts prior to tracking. The algorithms rely on the assignment of unique labels that permit the establishment of correspondence across time. Gust front statistics, such as propagation speed and location, are computed for the front





**FIGURE 8.** The bow-tie functional template used for thin-line smoothing: (a) index kernel and (b) corresponding scoring functions. (For an explanation of functional templates, see the box, "Functional Template Correlation.")

as a whole. This approach is adequate for simple cases. Inevitably, however, complex rules are required to handle the labeling, correspondence, and tracking for cases in which a single front breaks up into disjoint fragments or for cases in which multiple fronts merge or collide. Given the variable nature of gust front behavior, the construction of a fully comprehensive set of rules that are correct for all possible circumstances is a difficult task.

The problem is bypassed in MIGFA by making the goal of extraction the identification of all points (collectively called an *event*) that lie in *any* gust front. Certainly, some chains of points are spatially segregated or have different velocities. For purposes of reporting, such chains can be inferred to belong to separate gust fronts even though there is no concerted attempt to label or track gust fronts as entities. Instead, individual points are tracked across time; that a point belongs to one gust front or another is irrelevant to processing. MIGFA predictions are elastic in

that the variable velocities of different points along the gust fronts are each used to make predictions of what the gust front appearance will be at some time in the future.

Thin lines in the combined interest image can be fragmented for gust fronts that intersect with out-of-trip weather or for fronts obscured by storm regions. To bridge gaps between collinear fragments and to suppress random unaligned high-interest values, MIGFA uses thin-line smoothing of the combined interest image. Figure 8 shows the bow-tie functional template used as the basis for thin-line smoothing. The template, inspired by the receptive field of the cooperative cell of the Boundary Contour System developed by S. Grossberg and E. Mingolla [10], has a bow-tie shape that weights the influence of the end regions over that of the center by placing more kernel elements at the ends. Consequently, the template generates high output interest scores for an image point between two collinear high-interest segments, even if

that middle point itself has a low input interest value. Because of the scoring-function design, the bow-tie filter suppresses those collinear interest values which are below the level of ambiguity (0.5), and amplifies those values which are above the level of ambiguity. With this design, the boundaries between gust fronts and background are sharpened, resulting in cleaner shapes for subsequent processing. An example of an input image of combined interest and an output smoothed image are shown in Figure 9.

A threshold of 0.5 is then applied to the smoothed image to create a binary image of candidate fronts. The lengths of resulting elongated shapes are then computed, and the elements of those binary shapes which are too short (<6 km for the ASR-9 WSP) are set to 0. The result of this process is shown in the frame labeled "match > 0.5" in Figure 9.

The bow-tie functional template also generates a map of orientations. In the orientation image, each element indicates the orientation that is associated with the highest-scoring bow tie rotated at 10° increments from 0° to 170°. Black pixels correspond with best matches at 0°; white pixels correspond with best

matches at 170°. An example orientation image is shown in Figure 9.

The elongated binary shapes of the "match > 0.5" image can be thinned down to a single-pixel-width skeleton by using an FTC implementation of a modified version of S. Levialdi's homotopic thinning [11]. The result of thinning is shown in the frame labeled "marked thinned" in Figure 9.

The chains of points resulting from thinning are then extended along ridges of relatively high interest by using what is essentially a road-following algorithm. At each end point, the pixels immediately surrounding that point are examined by looking outward from the rest of the chain for the maximum-interest pixel with an orientation (found in the orientation image) that is within a specified angle from that of the initial end point. When the maximum interest score of a new point falls below 0.2 or when no new point has an orientation consistent with the initial end point, extending halts. The result of the extending process is shown in the frame labeled "extended" in Figure 9.

After the chain-extension process has been com-



**FIGURE 9.** Extraction steps. Candidate gust fronts are extracted from a combined interest image. For a description of the different steps involved, see the subsection "Extraction" in the main text.

pleted, the resulting image may be highly branched and it may contain loops. For further refinement of the image, chain segments are assigned scores based on the sum of the corresponding interest values found in the smoothed interest image. In each disjoint network of chain segments, the single most interesting (usually, but not always, the longest) non-looping combination of chain segments is extracted as the candidate gust front. Once the most interesting chain has been extracted, the process is repeated on the remaining unextracted chain segments to find the next most interesting combination of chain segments. The extraction process is repeated until the most interesting remaining chain is below an empirically determined interest threshold. In Figure 9, the frame labeled "selected chains" shows the set of above-threshold combined chain segments that were extracted from the "extended" image.

#### *Tracking/Heuristics*

As stated earlier, each point in the extracted event is tracked individually. The tracking of a particular point requires that the corresponding point in the event immediately prior to the current event be found. Correspondence can be difficult to establish when several gust fronts collide; in such cases, the point in the prior event that is closest to a point in the current event might not necessarily be the correct corresponding point. Consequently, the corresponding point is chosen to be the closest point in the immediately prior event for which the orientation and speed are consistent with the given point in the current event. If no such point in the prior event is found, then the corresponding point is assumed to be the closest point. Once correspondence for a point is established, the point is indexed by creating a pointer linking that point to the corresponding point in the immediately prior event. If the distance between the two corresponding points is too large or if the distance is inconsistent with prior history, then the point is unindexed (i.e., the link is broken). Through the index links, a point can be tracked backwards in time to its first recorded instance. The number of prior events through which a point can be tracked is called the point's depth. (A depth of

0 means that the point is unindexed.) Once indexed, each point is assigned the following attributes: coordinates, distance moved, direction moved, depth, Doppler value, interest value, and propagation speed.

After indexing, each extracted chain of points is edited:

1. If the direction a single point moves is opposite (approximate difference of  $180^\circ$ ) from its neighbors, the direction of the point is reversed.
2. Single chains may be divided into two subchains if a persistent discontinuity in velocity or a persistent change in orientation is detected at some point along the chain.
3. Various parameters such as propagation speed, Doppler value, and direction of motion are smoothed along the length of each chain.
4. Heuristics are applied that, when satisfied, unindex individual points in a chain. If more than half of any chain's points become unindexed, all points in the chain are unindexed.

The heuristics mentioned in item 4 above are based on knowledge of how false alarms can be distinguished from real gust fronts. For example, if the direction a point moves is inconsistent with the measured Doppler value, the point is unindexed. Or, if the point is approaching the radar site and moving in the same direction and no faster than the winds measured by anemometers at the radar site (i.e., there is no convergence), the point is unindexed.

In the final stage of tracking, a binary decision is made for each chain as to whether the chain should be declared a gust front. A chain's summed interest score and the depths of its constituent points are used to make the decision. For chains with high summed interest scores (reflecting a higher degree of confidence), points with lower depths may be included. On the other hand, chains that have low summed interest scores are less likely to be gust fronts and are thus required to accumulate higher depths before being included in the announced gust front detections. The frame labeled "indexed event" in Figure 5 shows the set of all extracted points. White pixels represent those points which have the sufficient depths and interest scores to be reported. Gray pixels represent those

points which will not be reported due to a lack of confidence. In the frame labeled "history," the reported points are shown in context with previously reported events.

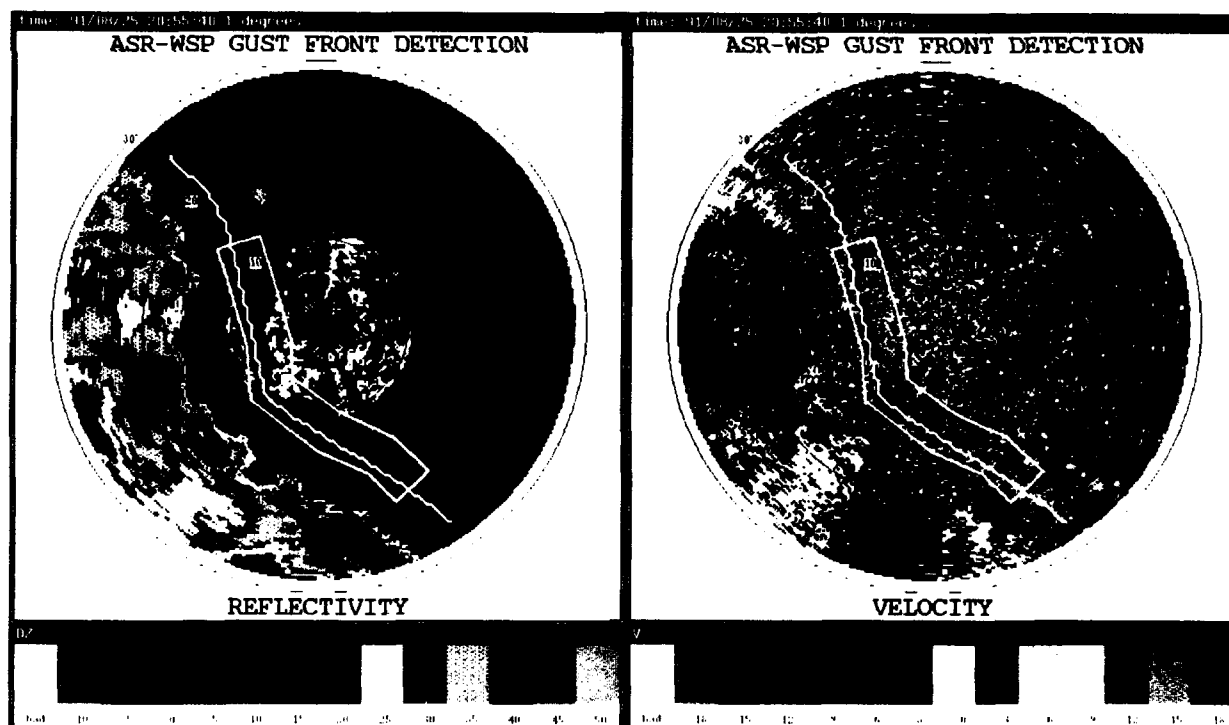
### *Prediction*

The current extracted event, indexed into the prior history, is used to predict the future locations of those points which have the sufficient depths and interest scores. Given the direction moved, the propagation speed, and the current coordinates of a point, a new coordinate is computed for some specified time in the future. Gaps can arise between the projected future coordinates of two adjacent gust front points when the orientations and velocities of the points are not identical. In such cases, the gaps are filled in. An example showing the reported chains and their expected locations after 10 and 20 min is shown in the frame labeled "predictions" in Figure 5.

### **Results**

The performance of MIGFA has been scored against human interpretations of the same input radar data. Implicit in this statement is the assumption that human interpretations are 100% accurate. As we will see later, this assumption is not always correct.

The human interpreter had access to both Doppler and reflectivity images for an entire sequence of ASR-9 WSP scans, which could be viewed separately or in sequence as a movie. For each scan, a description of "truth" (i.e., the interpretation of the scan by a human) was stored in a table as a list of coordinates marking the gust front end points and an intermittent sampling of points in between. For categorization of results, the estimated maximum wind shear in the zone of convergence was also stored. This scoring exercise was intended to measure MIGFA's detection performance, not the end-to-end gust front detection



**FIGURE 10.** Human versus MIGFA interpretation of ASR-9 WSP data. The 5-km-wide box denotes a region where a human interpreter has detected a gust front. The single line represents a detection by MIGFA. Note that the human interpreter did not include the extreme ends of the front because the ends were nearly radially aligned and had weak reflectivity values—characteristics of out-of-trip weather. However, because the extended thin line moved consistently with the center of the front and because the variance of Doppler velocity values associated with the thin line was too low to be out-of-trip weather, MIGFA probably gave the more likely interpretation. The reflectivity is given in dBZ, and the velocity in m/sec.

capability for the ASR-9 WSP. Consequently, the human interpreter was restricted to including in the truth set only those gust fronts which had some visible signature, however subtle. Other data sources, such as matching TDWR data and anemometer measurements of winds over the radar site, were used to confirm or deny the existence of gust fronts that had an ambiguous appearance in ASR-9 WSP data. The interpreter, however, did not use these other data sources to define gust fronts in the absence of visible ASR-9 WSP signatures. For cases in which MIGFA detections in ASR-9 WSP data were scored against a human interpreter looking at TDWR data, the same procedures were used to generate the TDWR truth tables.

An automatic scoring procedure, described in detail by D. Klinge-Wilson et al. [12], compares computed gust front detections with human-generated truth (see Figure 10). Briefly described, the scoring algorithm draws lines that connect the sequence of coordinates encoding the human-estimated limits of a gust front. The lines are then expanded to a 5-km-wide region that is called, in this article, a truth box. Computed gust front detections overlapping with some portion of the truth box are counted as successful detections while those not overlapping are counted as false alarms. A probability of detection (POD) is computed by dividing the number of successfully detected fronts by the number of fronts identified by the human interpreter. The probability of a false alarm (PFA) is the number of false alarms divided by the total number of algorithm-generated detections. (Note: In this article, POD and PFA values will be expressed

as percentages.) In addition to the hit-or-miss POD and PFA scores, scoring is also done in terms of the percent overlap of computer-generated detections and truth boxes. The percent length detected (PLD) is the number of points in an algorithm-generated detection that fall within a truth box divided by the length of that truth box (in pixels). The percent false length detected (PFD) is the number of points in an algorithm-generated detection that fall outside any truth box divided by the total number of algorithm-generated gust front points.

One improvement to this method is the use of a MAYBE category of truth. Often gust fronts or parts of gust fronts are only marginally detectable, forming a gray area in which the human observer is undecided or uncertain. If an algorithm detects a weak gust front associated with an ambiguous signature, the detection should not count as a false alarm. Similarly, if the algorithm misses a gust front that is too weak to have any operational significance, the miss should not affect the POD and PLD scores. Radar image features that are categorized as MAYBE are omitted from scoring.

Table 1 compares the performance of MIGFA against the latest version of AGFA, which uses more conventional methods of signal processing and computer vision. The test set of ASR-9 WSP data collected in Orlando, Florida, during field testing in 1991 contained nine different moderately strong gust fronts tracked through 15 hours (372 images). A human interpreter looking at the same data detected 280 instances of the nine gust fronts. The first two columns of Table 1 indicate that MIGFA increased

**Table 1. AGFA and MIGFA Performance\* on ASR-9 WSP Data**

	Gust Fronts		Gust Front Length	
	Probability of Detection (POD)**	Probability of a False Alarm (PFA)**	Percent Length Detected (PLD)	Percent False Length Detected (PFD)
Baseline (AGFA)	56.7	4.6	38.9	12.9
MIGFA	88.1	0.6	86.2	33.4

\* As scored against human interpretations of ASR-9 WSP data

\*\* Expressed as a percent

**Table 2. AGFA and MIGFA Performance\* on ASR-9 WSP Data**

	<i>Gust Fronts</i>		<i>Gust Front Length</i>	
	Probability of Detection (POD)**	Probability of a False Alarm (PFA)**	Percent Length Detected (PLD)	Percent False Length Detected (PFD)
Baseline (AGFA)	42.6	3.2	21.0	4.2
MIGFA	75.1	0.0	58.7	6.4

\* As scored against human interpretations of matching TDWR data

\*\* Expressed as a percent

by more than 50% the number of fronts detected by AGFA, while decreasing the false-alarm rate. Similarly, the PLD scores (column 3) indicate an improvement in detection performance. The increase in PFD (from 12.9% to 33.4%), however, appears to suggest that MIGFA is not as good as AGFA at discriminating the extent of individual fronts.

For a better understanding of why MIGFA was extending fronts beyond what the human interpreter believed appropriate, we examined several cases in which the PFD was high. In most of those cases, we found the extra points that MIGFA included in the gust front detections were believable. For example, Figure 10 shows a gust front truth box that overlays a MIGFA-generated detection. The human interpreter was reluctant to include the extreme ends of the front because the ends were nearly radially aligned and had weak reflectivity values—characteristics of out-of-trip weather. However, because the extended thin line moved consistently with the center of the front and because the variance of Doppler velocity values associated with the thin line was too low to be out-of-trip weather, MIGFA probably gave the more likely interpretation of the scene.

To substantiate such anecdotal observations, we took the gust fronts that MIGFA and AGFA had detected in ASR-9 WSP data and scored the fronts against human interpretations of TDWR data that had been taken at the same time. Although the resulting scores (Table 2) support the general trend of the first three columns of Table 1, the PFD for MIGFA (6.4%) is now roughly the same as that for AGFA (4.2%). Because gust fronts are more readily observ-

able in TDWR imagery, we assume that the TDWR truth (i.e., the TDWR data as interpreted by a human) is more accurate than the ASR-9 WSP truth (i.e., the ASR-9 WSP data as interpreted by a human). Thus the difference between the PFDs as scored against ASR-9 WSP and TDWR truths crudely approximates the percentage of detected gust front points missed by the human interpreter. For MIGFA, this difference ( $33\% - 6\% = 27\%$ ) added to the PLD scored against the ASR-9 WSP truth (86%) is 113%; i.e., MIGFA's performance was 13% better than that of the human interpreter. For AGFA, the comparable result is  $13\% - 4\% + 39\% = 48\%$ .

MIGFA was installed at the ASR-9 WSP site at Orlando International Airport in the spring of 1992 and was part of a formal operational test from 8 July to 20 September. During this time, gust front detections and predictions were relayed to air traffic controllers for their use in planning air traffic operations. During the early part of the summer, several minor problems and algorithm deficiencies were identified, and several fixes and enhancements were added during the middle of July. Careful interpretation, or "truthing," of the ASR-9 WSP data by a human was done from 1 August to 20 September.

As with the off-line testing described earlier, the on-line performance was scored against human interpretations of the same data. Table 3 shows the performance statistics for the test period. In general, the on-line test results substantiate the off-line results. Not surprisingly, the POD (75%) and PLD (81%) were somewhat lower than for the off-line test results shown in Table 1. Most of this differ-

**Table 3. MIGFA Results\* on ASR-9 WSP Data**

	<i>Gust Fronts</i>		<i>Gust Front Length</i>	
	Probability of Detection (POD)**	Probability of a False Alarm (PFA)**	Percent Length Detected (PLD)	Percent False Length Detected (PFD)
MIGFA	75.4	1.8	80.8	21.1

\* Results are scored against human interpretations of the same ASR-9 WSP data

\*\* Expressed as a percent

Note: The data are for the period 1 August to 20 September 1992 in Orlando, Florida

ence can be explained by two problems.

First, several gust fronts had reflectivity values at or below the sensitivity limits of the ASR-9. Of course, those fronts with reflectivity values below the ASR-9 limits were missed by both MIGFA and the human interpreter. But there were a few cases of marginal contrast in which the human could detect a gust front while MIGFA had not accumulated enough confidence to declare an alarm. Note that, unlike MIGFA, the human interpreter had the opportunity to examine the sequence of radar images repeatedly and could use information from scans late in the sequence to confirm or deny the existence of the gust front in early scans. Not much can be done to overcome the sensitivity limits of the ASR-9. In most (but not all) cases, however, gust fronts with marginal reflectivity levels were associated with weak wind shears. Because these weak fronts had a minimal impact on airport operations, a failure to detect such fronts was not a significant liability.

The second problem was that several gust fronts were missed due to obscuration. In these cases, storm regions or out-of-trip weather were extensive enough to hide or fragment the thin-line signatures so that some gust fronts were detected late, dropped early, or sometimes missed altogether.

The PFA (1.8%) represents 19 false detections out of 1080 total detections generated by MIGFA in more than 14,000 scans processed. The high PFD (21.1%) is almost entirely the result of MIGFA's extending gust fronts beyond the ends delimited by the human interpreter. With the use of anticipation based on prior tracking data, MIGFA was able to extend the

detected gust front lengths through areas where the signatures appeared ambiguous. As was seen with the off-line testing described earlier, a case-by-case analysis indicates that most of these extensions were in fact justified even though they were inappropriately scored as false lengths. Rescoring the results against TDWR data should improve the PFD score.

Another way to assess detection performance is to score only those gust fronts which had an impact on airport operations. From 20 July to 20 September, 14 convergent wind shears of greater than 15 kn were recorded on the anemometer network at the airport. Two of the wind shears were the result of short-lived localized winds beneath storm regions that were directly over the airport. The cause of a third wind shear could not be determined for certain, but was probably due to a microburst that was reported at the south end of the airport just as the wind shear was recorded. In none of these three instances could human interpreters find evidence of gust fronts in the ASR-9 data.

Of the 11 remaining wind shears, which were all verified later as gust fronts by human interpreters, MIGFA correctly tracked eight at least up to (but not always over) the airport. In the eight cases, air traffic controllers were given initial warnings from 18 to 79 min prior to the arrival of the front. Of the three missed gust fronts, one was occluded by fast-moving storm regions that were trailing the front. The second missed gust front had a very weak fragmented thin-line signature that was missed both by MIGFA and the human operators at the radar site who were logging weather and system activity. The third missed

case was a young gust front that had been generated by a large microburst only 5 km away from the runways. Because of its youth, the gust front had not yet developed a thin-line signature. Human interpreters who studied the radar scans after the testing was completed could find no evidence of this particular gust front in the ASR-9 WSP data, but could see a small zone of convergence without a corresponding thin-line signature in the data from TDWR. In summary, although MIGFA correctly detected and tracked (up to the airport) 8 out of 11, or 73%, of the gust fronts that had an impact on airport operations (wind shear  $> 15$  kn), human operators working at the radar site were able to log 9 out of 11, or 82%, of the same gust fronts.

False gust front detections that are reported to be approaching an airport can also adversely affect airport operations. If a false alarm were trusted, inappropriate changes in airport operations planning might be made and the resulting delays could be just as bad as when a gust front is missed. During the test period, three incoming events—covering a combined time of 24 min (12 scans)—were scored as false alarms. Only one event generated a false wind-shear hazard alert (wind shear  $> 15$  kn). All three were probably the result of thin lines from stratiform rain. None of these false alarms should have influenced airport operations planning because in each case tracking was dropped when the estimated time of arrival at the airport was more than 40 min.

### **Evaluation**

Using the same input ASR-9 WSP data, we have shown by direct comparison that MIGFA provides a substantial improvement over AGFA in detection performance. We have also provided indirect evidence suggesting that, given the same input data, MIGFA may be nearly as good as human interpreters. However, the absolute reported POD scores for MIGFA (88% when scored against ASR-9 truth and 75% when scored against TDWR truth) are potentially misleading and should be regarded with caution because the dataset used for comparison testing was relatively small and from only one season at one site. Thus the off-line test probably did not contain a good representative sampling of gust fronts. The test did,

however, provide a reasonable basis for comparing MIGFA against the older algorithm.

The results for the operational test period should be more representative of MIGFA performance. In the on-line testing, the POD and PLD scores remained high (in fact, the scores were only somewhat lower than those reported for the off-line testing), but an apparent problem in the relatively high PFD score (21%) persisted. Again, as was shown in the initial off-line testing, many of the false detections were in fact weak gust fronts or parts of gust fronts that the human interpreter had overlooked. Although these results have not been rescored against TDWR truth, the existences of gust fronts were established for several cases by the examination of matching TDWR or anemometer data.

An analysis of results accumulated during the 1992 operational test period has identified three main classes of failure modes for the ASR-9 WSP version of MIGFA. The failures within the first class are a direct result of the limited sensitivity of the ASR-9. Some gust fronts that were visible in TDWR data and that had an impact on the Orlando airport with moderate wind shear had reflectivity returns below the sensitivity of the ASR-9. Like MIGFA, experienced human observers using ASR-9 data did not see such gust fronts, although with the benefit of hindsight the observers could sometimes detect above-threshold fragments of what must have been the approaching front. In general, gust fronts with thin-line signatures that have reflectivity levels at or below the sensitivity limits of the ASR-9 usually (but not always) exhibit weak wind shears, making them operationally less significant.

The second failure mode was due to a lack of reliable Doppler estimates of velocity in clear air. Because of the unreliability of these values, the ASR-9 version of MIGFA had to rely on thin-line signatures for detecting gust fronts. As discussed earlier, however, not all thin lines are caused by gust fronts. For example, elongated low-reflectivity storm echoes associated with extensive areas of stratiform rain moving with the ambient wind were a source of false alarms in the operational testing. Because the reflectivity levels of light-rain echoes overlap with the range of reflectivity levels exhibited by gust fronts, the



thin-line feature detectors produced high interest values. In most of these cases, the thin-line features associated with the stratiform rain were transient and did not accumulate enough confidence through time for the system to declare a gust front. Some false alarms could be dismissed because of the lack of implicit convergent wind shears, which were computed by comparing the radar-measured winds in incoming candidate gust fronts with the winds measured by the airport anemometers surrounding the radar site. In at least one case, however, a false alarm could not be rejected with this criterion. The winds at the airport were variable and not representative of the winds immediately in front of the feature, which was 15 km away from the airport.

The third failure mode was caused by obscuration. During the 1992 operational test period, several gust fronts were either detected late, prematurely lost, or not detected at all due to obscuration by patches of high reflectivity that were caused by storms, range-ambiguous echoes, or ground clutter. Even in places where the *thin-line features were visible*, such patches of high reflectivity had sometimes fragmented the features into short segments. One missed gust front is known to have had an impact on the airport with a wind shear greater than 15 kn.

Experience gained from the operational test period has led to the implementation of a partial solution to the obscuration problem. The solution uses anticipation and the system's ability to detect obscuring weather patterns. Given a sequence of images, there often exists some time interval when a significant part of the gust front is not obscured and tracking can be initiated. Once sufficient confidence has accumulated, the system begins to anticipate where the gust front ought to be in the next scan. In normal operation, the thin lines of increased interest in the anticipation interest image are used to boost weak signals that would otherwise be below threshold for detection. (During the operational testing, obscuration suppressed all interest, eliminating any signals for anticipation to confirm.) In the modified system, when obscuring weather is found to overlap the anticipated gust front locations, the anticipation interest values can be increased to a level at which detection is triggered regardless of how weak the other evidence is. In

other words, when obscuration is detected, the anticipation interest image becomes absolute, resulting in spatially restricted coasting.

### Summary

The identifying signatures for gust fronts—thin lines of increased reflectivity, boundaries of converging Doppler values, and motion perpendicular to the thin lines and convergence boundaries—are conceptually easy to define and exploit as the basis of detection algorithms. And yet, although several research groups have worked collectively for nearly 10 years to develop reliable automatic gust front algorithms, none of the algorithms has demonstrated performance comparable to the ideal of human performance.

The problem is that automatic gust front detection, like other applications in computer vision, is deceptively much more difficult than the task of simply finding one or more signatures. Human observers use a variety of perceptual skills that have been notoriously and surprisingly difficult to implement in computer-vision systems. For example, humans have a talent for dealing with uncertain, ambiguous, and even contradictory evidence. Humans use specific knowledge of the object being sought and the context of observation as well as the object's spatial and temporal context. Unlike most other computer-vision and automatic target recognition (ATR) methodologies, the Experimental Target Recognition System (XTRS) and the Machine Intelligent Gust Front Algorithm (MIGFA) do not rely on machine intelligence only at the higher symbolic levels of processing. XTRS provides a framework for applying knowledge at the level of raw data by using specialized techniques for knowledge-based signal processing and pixel-level processing of evidence. The fact that MIGFA performance is competitive with that of human observers is at least partially due to this use of low-level machine intelligence.

### Acknowledgments

The authors would like to thank the staff of the FL-3 ASR-9 radar site in Orlando, Florida, who were responsible for running MIGFA, recording data, and logging results during the operational test period in 1992. These people include Wes Johnston, Craig

McFarland, Jeff Boisseau, and Cindy Meuse. The authors would also like to thank Joe Cullen for generating "truth" from ASR-9 WSP data, against which all algorithm results have been scored. XTRS, the prototype object recognition system upon which MIGFA is based, has evolved over several years of collaboration with Jacques Verly and Dan Dudgeon.

This work was sponsored by the Federal Aviation Administration.

## REFERENCES

1. J.G. Verly, R.L. Delanoy, and D.E. Dudgeon, "Machine Intelligence Technology for Automatic Target Recognition," *Linc. Lab. J.* 2, 277 (1989).
2. H. Uyeda and D.S. Zrnić, "Automated Detection of Gust Fronts," *J. Atmos. Oceanic Tech.* 3, 36 (1986).
3. S. Smith, A. Witt, M. Eilts, L. Hermes, D. Klinge-Wilson, S. Olson, and J.P. Sanford, "Gust Front Detection Algorithm for the Terminal Doppler Weather Radar Part I: Current Status," *Proc. 3rd Intl. Conf. on the Aviation Weather System, Anaheim, CA, Jan. 1989*, p. 31.
4. L. Hermes, A. Witt, S. Smith, D. Klinge-Wilson, D. Morris, G. Stumpf, and M. Eilts, "The Gust Front Detection and Wind Shift Algorithms for the Terminal Doppler Weather Radar System," *J. Atmos. Oceanic Tech.* (to be published).
5. M. Eilts, S. Olson, G. Stumpf, L. Hermes, A. Abrevaya, J. Culbert, K. Thomas, K. Hondl, and D. Klinge-Wilson, "An Improved Gust Front Detection Algorithm for the TDWR," *Proc. 4th Intl. Conf. on the Aviation Weather System, Paris, June 1991*, p. J37.
6. M.W. Merritt, D. Klinge-Wilson, and S.D. Campbell, "Wind Shear Detection with Pencil-Beam Radars," *Linc. Lab. J.* 2, 483 (1989).
7. T.A. Noyes, S.W. Troxel, M.E. Weber, O.J. Newell, and J.A. Cullen, "The 1990 Airport Surveillance Radar Wind Shear Processor (ASR-WSP) Operational Test at Orlando International Airport," *Project Report ATC-178*, MIT Lincoln Laboratory (July 1991), AD-239852 (NTIS only).
8. M.E. Weber, "Airport Surveillance Radar (ASR-9) Wind Shear Processor: 1991 Test at Orlando, FL," *Project Report ATC-189*, MIT Lincoln Laboratory (June 1992), AD-252246 (NTIS only).
9. R.L. Delanoy, J.G. Verly, and D.E. Dudgeon, "Pixel-Level Fusion Using Interest Images," *Technical Report 979*, MIT Lincoln Laboratory (26 Apr. 1993).
10. S. Grossberg and E. Mingolla, "Neural Dynamics of Perceptual Grouping: Textures, Boundaries, and Emergent Segmentations," *Perception and Psychophysics* 38, no. 2, 141 (1985).
11. S. Levialdi, "Parallel Pattern Processing," *IEEE Trans. Syst. Man Cybern.* 1, 292 (1971).
12. D.L. Klinge-Wilson, M.F. Donovan, S.H. Olson, and F.W. Wilson, "A Comparison of the Performance of Two Gust Front Detection Algorithms Using a Length-Based Scoring Technique," *Project Report ATC-185*, MIT Lincoln Laboratory (May 1992).



**RICHARD L. DELANOY** is a staff member of the Machine Intelligence Technology Group. His work spans the fields of computer vision, machine learning, and construction of object-recognition systems. From 1980 to 1983, he was a research scientist at the University of Virginia Department of Psychology, where he investigated the biochemical correlates of learning and the effects of stress-related hormones on electrophysiological models of memory. Before joining Lincoln Laboratory in 1987, he worked for GE Fanuc Automation N.A., Inc., as a software engineer developing numerical and programmable controllers for manufacturing automation. Dick received a B.A. degree in biology from Wake Forest University in 1973, a Ph.D. degree in neuroscience from the University of Florida College of Medicine in 1979, and an M.S. degree in computer science from the University of Virginia in 1987. He was a National Science Foundation Predoctoral Fellow and a National Institute of Mental Health Postdoctoral Fellow.



**SETH W. TROXEL** received a B.S. degree in meteorology from San Jose State University, California, in 1983 and went on to work as a meteorologist in the Atmospheric Lidar Group of the NOAA Wave Propagation Laboratory in Boulder, Colorado. In 1987, he joined Lincoln Laboratory as a software engineer and meteorologist with the Weather Sensing Group. Since coming to Lincoln Laboratory, he has been involved in the testing and development of hazardous-weather-detection capabilities for the Airport Surveillance Radar (ASR-9) and Terminal Doppler Weather Radar (TDWR). Seth's primary research interests are in the areas of remote sensing, software engineering, and design of algorithms for aviation weather products.

---

# Extracting Target Features from Angle-Angle and Range-Doppler Images

Su May Hsu

■ For diffuse targets, features such as shape, size, and motion can be determined from a time series of images from either angle-angle passive telescopes or range-Doppler radars. The extracted target features can then be used for automated target recognition and identification.

An algorithm that uses scene-analysis techniques has been developed to perform the feature extraction. The algorithm first processes the images to suppress noise, then applies a two-dimensional slope operation for edge detection to determine the target boundaries. Next, Hough transforms are used on the target edges to detect straight lines and curves, which are subsequently refined with line and curve fits. Groups of the fitted lines are then examined to form cylinders and cones representing typical target components. After these shapes have been identified, the target configuration, size, location, and attitude can be estimated. The target motion can then be inferred from a time series of attitudes that have been extracted from a sequence of images.

**F**OR A TARGET with rough surfaces, electromagnetic signals are reflected and returned from scatterers that are distributed over the entire target surface. The resulting imagery, whether angle-angle (passive telescopes) or range-Doppler (radar), will show a diffuse object with surface returns that became apparent along the sensor line of sight (LOS). From such images, the target shape, size, and orientation can be determined from pattern recognition and identification techniques. And, with a time sequence of images, the motion of the target can be estimated from its orientation history.

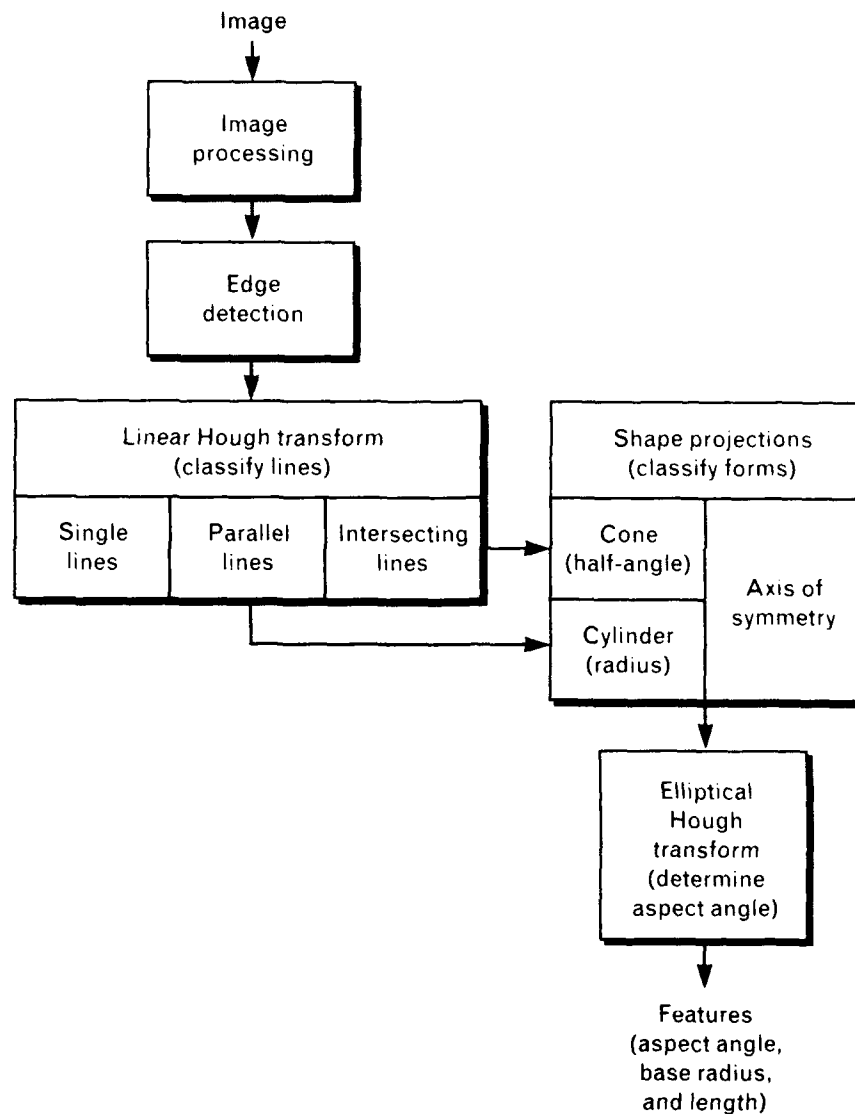
In the feature-extraction algorithm developed at Lincoln Laboratory, the images are first processed to suppress noise and to smooth the image surface. Edge detection is then performed to determine the target boundaries, and the detected edge points are integrated into line segments to form target shapes. Next, target dimensions and orientations are measured from

the line representation of the target, which is assumed to be axisymmetric.

In this article, examples of target feature extraction are demonstrated for both angle-angle and range-Doppler images. For angle-angle images, target orientation is obtained from the projected elliptical shape of circular components and the projected length of a symmetric body axis. For range-Doppler images, target dimensions are first determined from a sequence of range and Doppler extents that have been extracted from the images. The target aspect-angle history can then be derived from that same sequence of range and Doppler extents (as for angle-angle images) by using the estimated target dimensions.

## Image Scene Analysis

The goal of target feature extraction is to obtain shape information. From such information, the size, orientation, and position of a target can be estimated.



**FIGURE 1.** Feature-extraction process.

Shapes are composed of lines and curves that are collections of boundary edge points. Such edge points can be located with edge-detection techniques. The detected edge points can then be associated with lines and curves by using predetermined fitting constraints. For the current application, cylindrical and conical shapes are considered, and Hough transforms are used to detect the presence of lines and curves [1].

Figure 1 illustrates the feature-extraction process. First, image processing is performed to enhance the edge-detection process. The image processing includes the application of median filters to remove isolated

noise, the averaging of multiple image frames to enhance the image signals, and the spatial smoothing of the image surface. Thresholds are then applied to remove the image background. Next, a linear Hough transform is used to detect and collect lines, which are later examined to form the boundaries of cylinders and cones. The position and orientation of the axis of symmetry of each detected shape are then determined. For angle-angle imagery, an elliptical Hough transform is generally used to detect and fit the target-base curve to allow for the subsequent calculation of the target aspect angle. For range-Doppler imagery,

an elliptical Hough transform is used to determine the target dimensions, and, hence, to obtain the target aspect angle.

The three main steps of scene analysis for the extraction of target features are edge detection, line identification, and curve extraction. These three steps are described in the following subsections.

### Edge Detection

The boundaries of objects in an image exist at locations where the image values change abruptly. These abrupt changes can be detected by a spatial difference operator [2]. The design of such an operator depends closely on the quality and complexity of the image, and on the desired level of feature extraction. The current application uses spatial gradient operators [3].

The operator center is placed at an image location, and the result of the convolution operation on the image values represents the local gradient at that image location and in the operator's direction. The edge value at the image location  $I(x, y)$  can be expressed as

$$G[I(x, y)] = \sqrt{G_x^2 + G_y^2},$$

where  $G_x$  and  $G_y$  are the image gradients in the horizontal and vertical directions, respectively. The edge is in the gradient direction, which can be obtained as

$$\theta = \tan^{-1} \frac{G_y}{G_x},$$

where  $\theta$  is measured with respect to the  $x$ -axis.

Gradient operators of size 5, as shown in Table 1, are used. In general, the operator size should be decreased for coarser image resolutions because, as the resolution becomes coarser, the object boundaries become closer to each other.

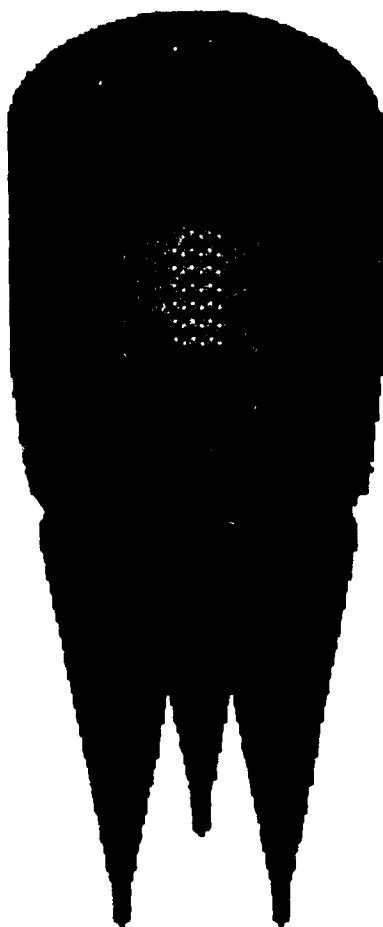
The calculated edge values are small for a smooth image surface and large for a discontinuous surface. Thus edge points can be detected by applying a threshold to the edge values. The threshold can be varied, depending on the level of detail desired for edge extraction. The examples presented in this article use a threshold equal to 10% of the dynamic range of the edge values, i.e.,  $G_{\min} + 0.10(G_{\max} - G_{\min})$ . The choice of the threshold, however, can be optimized with respect to the histogram of the edge values. Because of the spatial extent of the operator, the edge values obtained after the application of a threshold generally are thickly populated near or at target discontinuities. Thus a procedure for non-maxima suppression can be applied for the further thinning of these edges [4]. In the procedure, the edge value at a point is set to zero if the value is not the local maximum in the direction perpendicular to the edge direction. Figure 2 shows an example of the detected edge points superimposed over a gray-scaled image. The edge points include the outline of the cones and the silhouette of the cylinder against the background.

### Line Identification

After the edge points have been detected, they must be associated with lines and curves. For simple closed contours, the edge points can be chain coded [5], and

**Table 1. 5 × 5 Gradient Operators**

<i>x-Direction</i>					<i>y-Direction</i>				
-2	-2	-2	-2	-2	-2	-1	0	1	2
-1	-1	-1	-1	-1	-2	-1	0	1	2
0	0	0	0	0	-2	-1	0	1	2
1	1	1	1	1	-2	-1	0	1	2
2	2	2	2	2	-2	-1	0	1	2



**FIGURE 2.** Gray-scaled image with detected edge points.

the shape description can be performed with syntactic pattern grammars [6]. The current application considers images of complex objects comprising a combination of shapes, from which broken edges and edges inside other object boundaries are permitted. Thus a layered approach for shape formation is required.

Hough transforms can be used to detect the lines and curves in an image. In Figure 3(a), a line in two-dimensional (2-D) space is represented by the directed orthogonal distance  $h$  to the origin and the angle  $\theta$  that  $h$  forms with the  $x$ -axis. Any point  $(x, y)$  on the line will satisfy the following equation:

$$h = x \cos \theta + y \sin \theta.$$

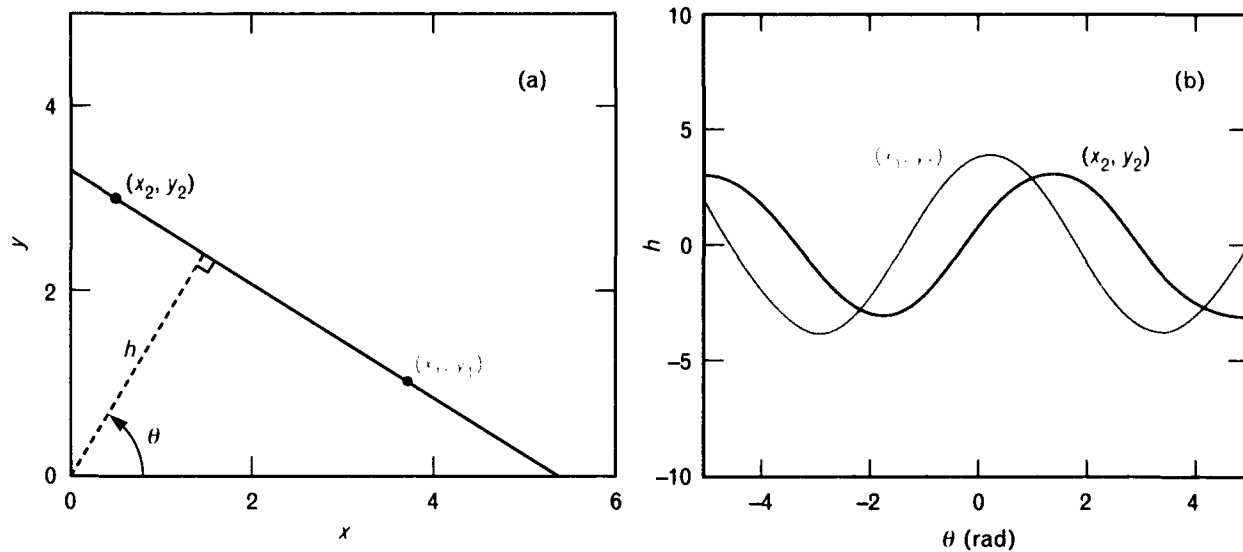
The set of lines passing through a given point  $(x, y)$  can be plotted in  $h$ - $\theta$  space. The result will be a sinusoidal curve, as shown in Figure 3(b). Note that

the  $h$  curves of two points in  $x$ - $y$  space will intersect periodically in  $h$ - $\theta$  space, and the points of intersection (spaced  $\pi$  radians apart) will correspond to the line defined by the two points in  $x$ - $y$  space. Thus, if the  $h$  curves of all edge points in an image are plotted in  $h$ - $\theta$  space, the points at which the  $h$  curves intersect will correspond to prominent lines in the image.

In practice, the  $h$ - $\theta$  space is divided into accumulator cells, and the cell value of a particular cell is increased by 1 each time a curve passes the cell's location. Because the direction of each edge point is known from the edge-detection process (discussed earlier), only the portion of the  $h$  curve for angles around that direction needs to be searched. A threshold is chosen at the half-length (in pixels) of the shortest line expected to be extracted from the image. Only those accumulator cells whose cell values are greater than the threshold are selected for further investigation. The selected cells represent a set of lines detected from the image.

For each detected line, a collection of points that lie on the line is gathered from the edge points. To represent the location of a line in the overall image, the mean of the collected points, i.e., the mean  $(x, y)$  value, is used. For cases in which several line segments from different image locations happen to be collinear, the line segments will be represented by just single line values of  $h$  and  $\theta$ . In such cases, additional lines—with the same line values  $h$  and  $\theta$  but with different mean  $(x, y)$  values—are added to the line set to represent the different line segments. The line set is examined further to eliminate those lines which have mean  $(x, y)$  values and  $h$  and  $\theta$  values that are close to other lines. The line set is then used as a set of seeds for growing connected line segments with a relaxation process, as shown in Figure 4.

During each iteration of the relaxation process, each edge point in the image is classified in association with one of the detected lines. The edge point is then added to the point collection of the associated line. When a point is not close enough to any of the lines (the allowable orthogonal distance for point classification should be less than half the distance of the closest distinctive feature lines in the image) or when the edge direction does not agree with the line direction (the allowable angular deviation should be less



**FIGURE 3.** Hough transform for straight lines: (a) line defined in  $x$ - $y$  coordinates and (b) Hough transform in  $h$ - $\theta$  space. A line in two-dimensional (2-D) space can be represented by the directed orthogonal distance  $h$  to the origin and the angle  $\theta$  that  $h$  forms with the  $x$ -axis, as shown in part a. For any given point, the set of lines passing through that point can be plotted in  $h$ - $\theta$  space. The result will be a sinusoidal curve, as shown in part b. Note that the  $h$  curves of two points in  $x$ - $y$  space will intersect periodically in  $h$ - $\theta$  space. The points of intersection (spaced  $\pi$  radians apart) correspond to the line defined by the two points in  $x$ - $y$  space.

than half the smallest angle formed by intersecting lines in the image), the point is not classified. At the end of each iteration, the current classification is compared with the results of the previous iteration. If any difference exists, the line set is recalculated from the current point collection and, with the updated line set, the point classification is reiterated. The relaxation process stops when the current classification has not changed from the previous iteration. After the relaxation process ceases, the lines are identified from the edge points, and the position and orientation of each line are calculated from the point collection with a least-squares-error fit. This information is then used for simple shape formation: parallel lines are identified for cylinders, and intersecting lines are chosen for cones. Finally, the axis of symmetry is determined for the position and projected orientation of each chosen shape.

#### Curve Extraction

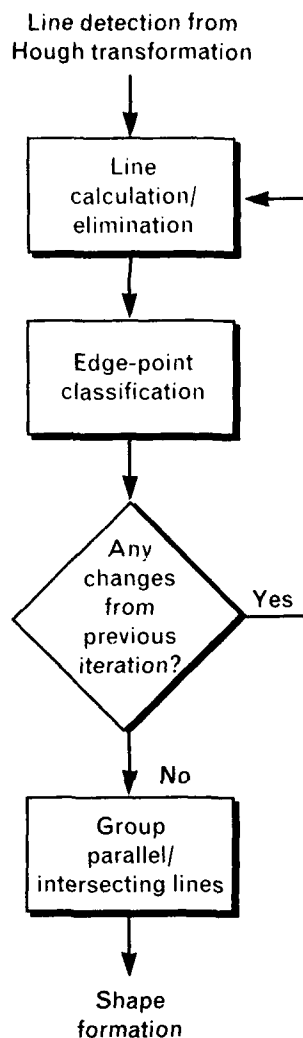
The circular base of a cylinder or cone generally appears elliptical in angle-angle imagery because the aspect angle between the object body axis and the

receiver LOS is usually nonzero. From geometry, the ratio of the minor and major axes of the projected ellipse is the cosine of the aspect angle. The aspect angle, together with the projected body-axis orientation, can be used to determine the body attitude with respect to the sensor in 3-D space. In extracting the elliptical base, it is assumed that the straight boundaries have already been identified. Thus the ellipse center will lie on the axis of symmetry of the shape. For cylinders, the major axis of the base ellipse will be half the distance between the two parallel edges that define the shape. For cones, the major axis will be the distance from one conical edge to the ellipse center. Thus the ellipse-extraction scheme first determines the base center along the axis of symmetry of the shape and then fits an elliptical curve at the base region to estimate the minor axis.

Figure 5(a) depicts a cone shape with an elliptical base. The ellipse function with a coordinate rotation  $\beta_r$  is expressed by

$$\frac{(x - x_0)_r^2}{a^2} + \frac{(y - y_0)_r^2}{b^2} = 1, \quad (1)$$





**FIGURE 4.** Relaxation process for line identification.

where  $(x_0, y_0)$  represents the ellipse center on the axis of symmetry having orientation  $\beta_r$ , and  $a$  and  $b$  are the minor and major axes of the ellipse, respectively. Equation 1 can be rewritten for the minor axis  $a$ :

$$a = \frac{|(x - x_0)_r|}{\sqrt{1 - \frac{(y - y_0)_r^2}{b^2}}}$$

The above equation indicates that the minor axis of the ellipse is a function of the location of the ellipse center  $(x_0, y_0)$  along the axis of symmetry, as shown in Figure 5(b). Thus the Hough transform space  $[a, (x_0, y_0)]$  can be used for ellipse detection in which

values of  $a$  are plotted for all edge points in the base region. The maximally accumulated cell in Hough transform space will then determine  $a$  and the location of  $(x_0, y_0)$ .

In addition, the edge detection of a point on the ellipse curve should be consistent with the curve tangent derived by differentiating Equation 1, as follows:

$$\tan \theta = \frac{d(x - x_0)_r}{d(y - y_0)_r} = -\frac{a^2}{b^2} \frac{(y - y_0)_r}{(x - x_0)_r}$$

Thus each edge point is checked with respect to location and edge orientation before being added to the collection of points for an extracted curve. The minor-axis value is then refined from the best curve fit of the collected points. The results of line and curve identification are shown and discussed in the following section.

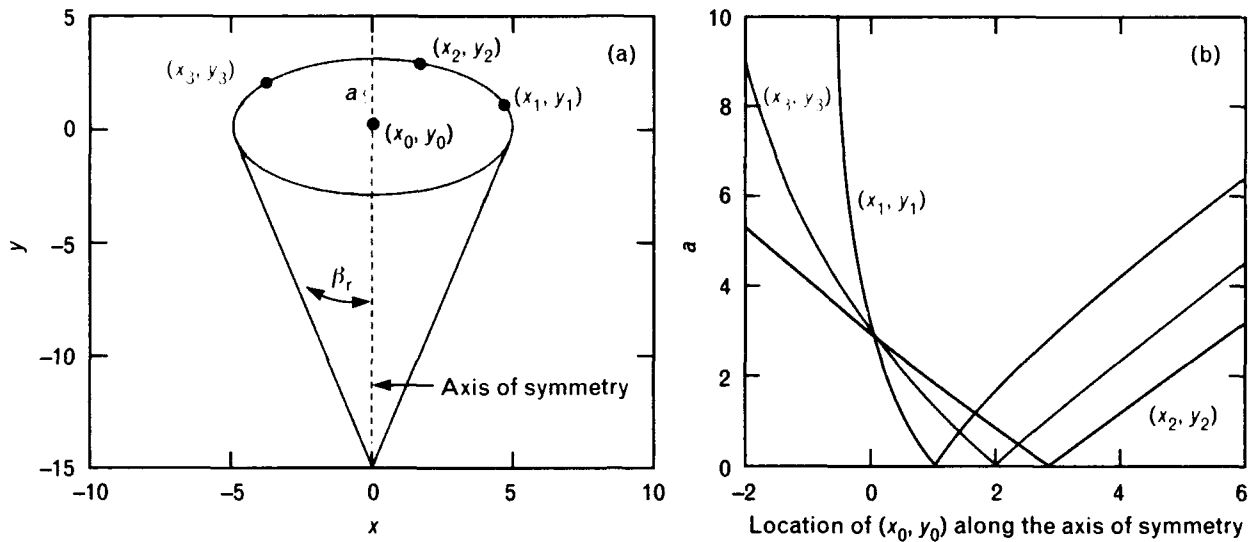
### Feature Extraction for Angle-Angle Images

For angle-angle imagery, targets are projected from 3-D space to a plane perpendicular to the sensor LOS, as shown in Figure 6. Thus spheres are projected as circles, and the circular bases of cones and cylinders become ellipses. In the images, the physical radii of the circular forms of targets are generally preserved without transformations: the major and minor axes of the elliptical projection are respectively the circular radius itself and the radius with a  $\cos \alpha$  factor,  $\alpha$  being the aspect angle (i.e., the angle between the target body axis and the sensor LOS). The boundary lines of targets are projected into lines in images, but their dimensions are generally transformed with a factor of  $\sin \alpha$ . With the use of such projection relationships between target 3-D space and image 2-D space, the size, shape, and orientation of a target can be inferred from its image features.

Two examples of simulated angle-angle images have been analyzed to demonstrate the extraction of target features from such imagery. First, a simulated image of a complex object is used for size and shape estimation. Then a sequence of cone images is employed for motion extraction.

### Size and Shape Estimation

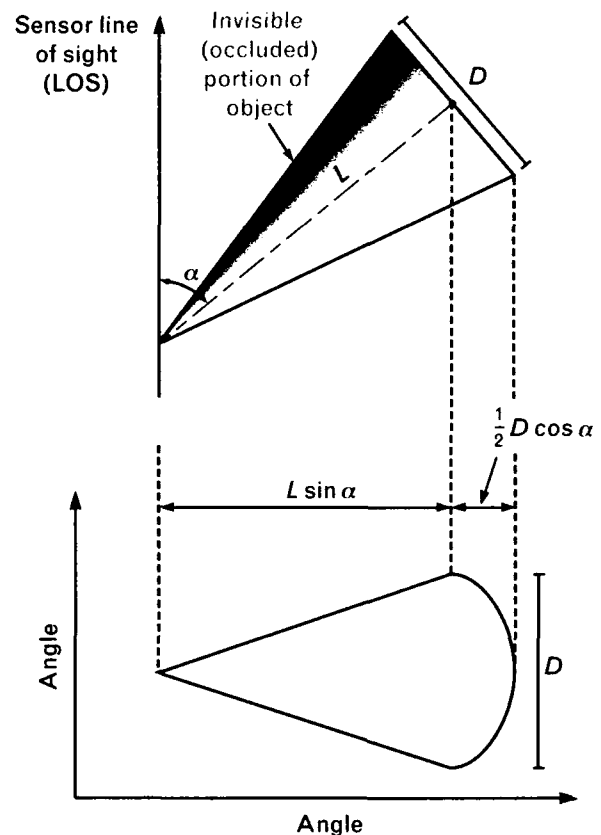
The model used in the example has a cylindrical main



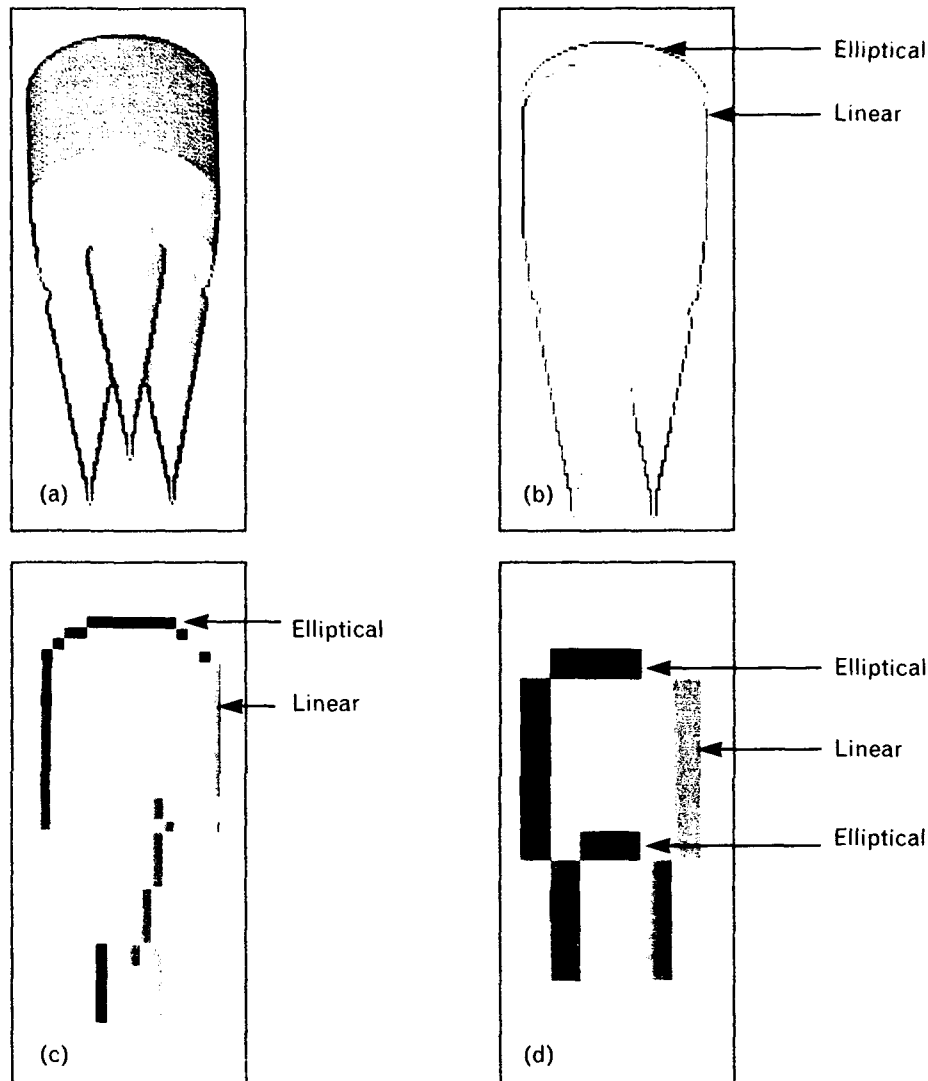
**FIGURE 5.** Elliptical Hough transform: (a) cone shape with elliptical base in  $x$ - $y$  coordinates after a coordinate rotation  $\beta_r$ , and (b) Hough transform in which  $a$ , the minor axis of the ellipse, is plotted on the vertical axis and the location of the ellipse center  $(x_0, y_0)$  along the axis of symmetry is plotted on the horizontal axis. For a detailed description of the mathematics involved, see the main text.

body with three cones mounted at the forward end. Figure 7(a) shows a simulated angle-angle image of the object at a  $50^\circ$  aspect angle, and Figures 7(b), (c), and (d) show the results of a scene analysis that was performed with pixel sizes of 2, 8, and 20 cm, respectively. For each of the scene-analysis images, the edge points have been detected and lines identified. (Note: The different lines are coded in different colors.) Ideally, two major parallel lines would be selected for the cylindrical body, and three pairs of intersecting lines would be chosen for the cones. Then an elliptical curve could be fitted for the cylindrical base to determine the aspect angle. Because the cylindrical body is large compared to all three of the pixel sizes used, the two parallel lines representing the shape are easily discernible in Figures 7(b), (c), and (d). But the cones, because of their smaller size, appear distorted in the images, particularly in Figure 7(d). Nonetheless, the cones are recognizable at pixel sizes of 2 and 8 cm.

Figure 8 shows the quantitative results. For all pixel sizes, the estimated cylinder radius (Figure 8[a]) agrees well with the model. Estimations for the cone radius (Figure 8[b]) and projected cone angle (Figure 8[c]) are good for pixels smaller than 8 cm, and the aspect



**FIGURE 6.** Geometry of angle-angle imagery of a cone.



**FIGURE 7.** Size and shape extraction for a simulated object consisting of a cylindrical main body with three cones mounted at the forward end: (a) angle-angle input image after edge detection, (b) line classification at a pixel size of 2 cm, (c) line classification at a pixel size of 8 cm, and (d) line classification at a pixel size of 20 cm.

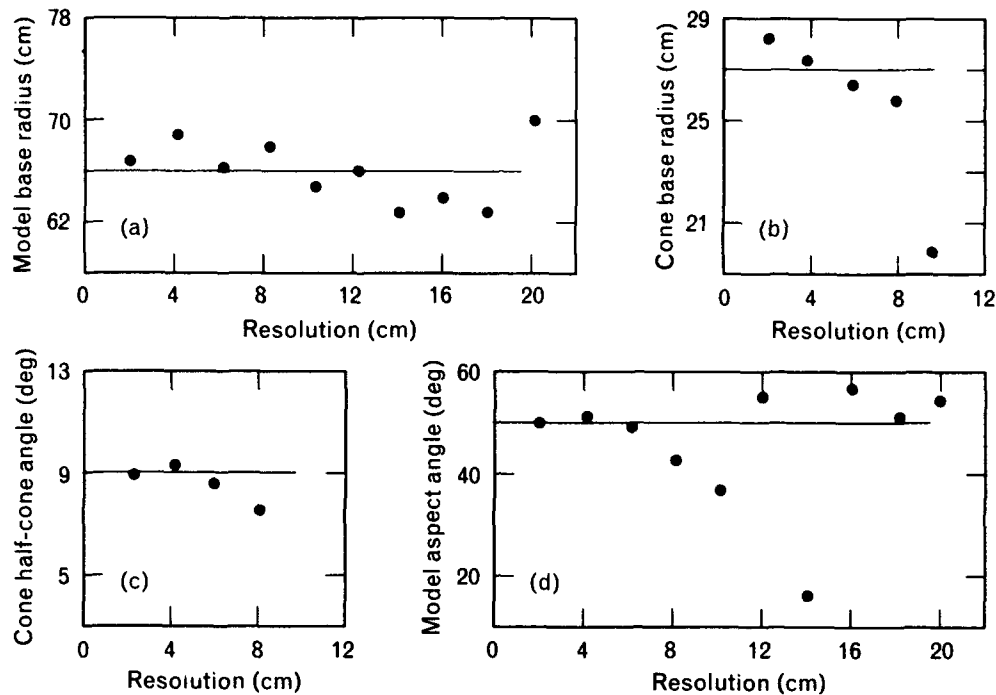
angle obtained from the elliptical-curve fit (Figure 8[d]) is also in agreement with the model for pixels smaller than 8 cm.

In general, curve fitting requires finer image quality than line fitting. Lines can usually be detected within one pixel to the true edge. The pixel size, which represents the sampling size at the image focal plane, is determined by the type of focal plane and the angular resolution, the diffraction or resolution limit of the imaging system, and the signal-to-noise

ratio for the target at the receiver.

#### *Motion Extraction*

Target motion can be used to aid the target-classification process, as has been demonstrated in range-Doppler imagery from millimeter-wave (MMW) and other microwave radars. With a time series of target attitudes extracted from such imagery, target motion can be inferred. The observation generally is more straightforward in a time series of high-resolution images.



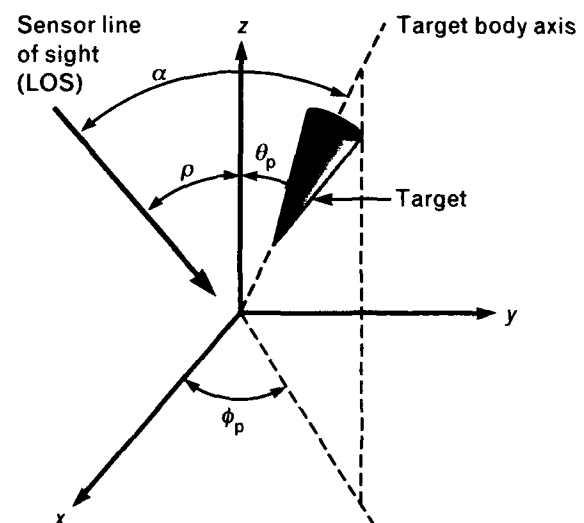
**FIGURE 8.** Feature-extraction performance at various pixel sizes for the simulated model of Figure 7: (a) calculated base radius of the model, (b) calculated base radius of a cone, (c) calculated half-cone angle of a cone, and (d) calculated aspect angle of the model. (Note: For comparison, the input values used in the simulation are indicated with straight lines.)

For purposes of demonstration, a free-body spin-precession motion is considered. Figure 9 shows a coordinate system with a spinning target and the body axis of the target precessing around the  $z$ -axis. The parameters  $\theta_p$ ,  $\phi_p$ , and  $\rho$  represent the precession half-cone angle, the precessing azimuthal angle, and the angle between the LOS and the  $z$ -axis, respectively. Suppose that  $\rho$  and  $\theta_p$  are constant during an observation. Then the aspect angle  $\alpha$  between the body axis and the LOS will be a function of  $\phi_p$ . For  $\rho$  in the  $z$ - $x$  plane,

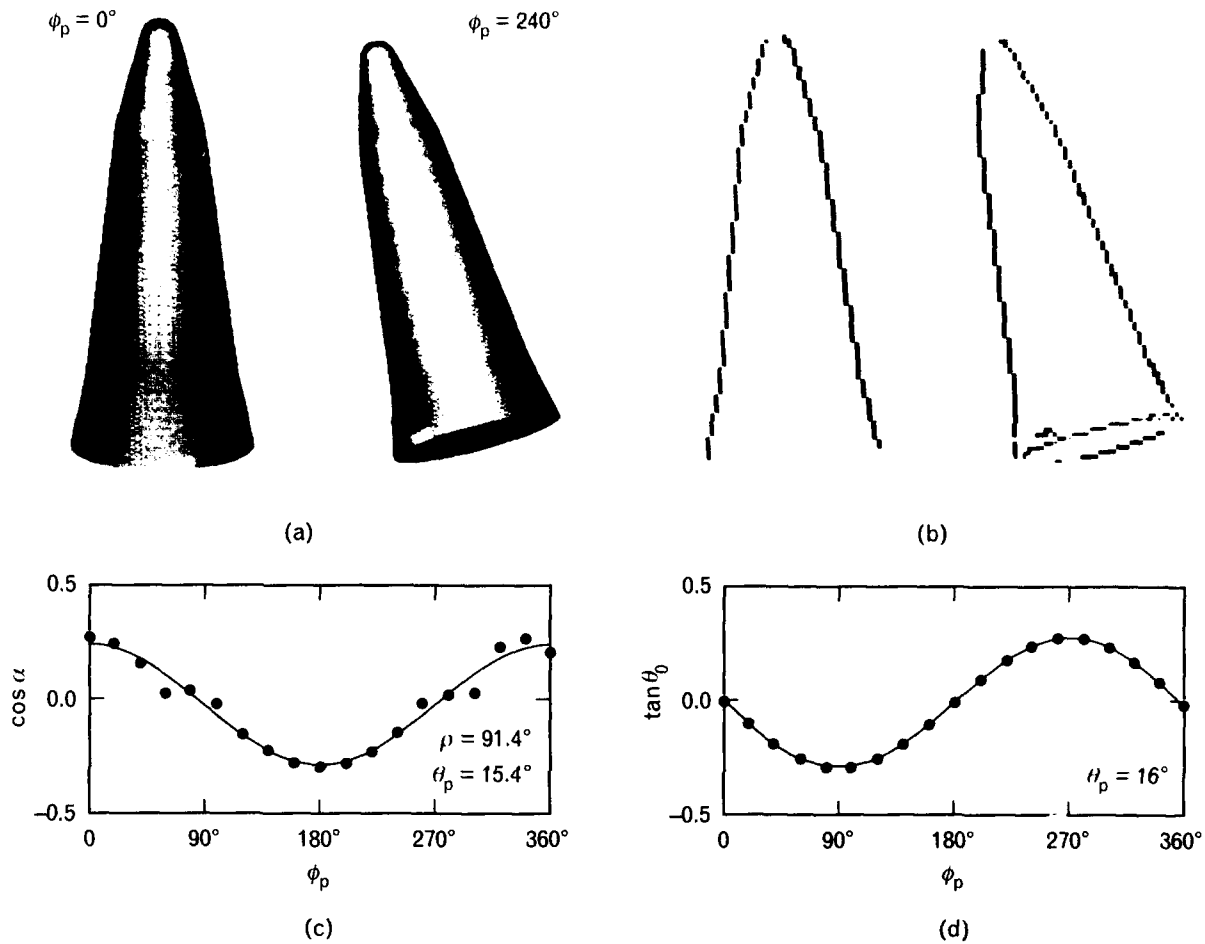
$$\cos \alpha = \sin \rho \sin \theta_p \cos \phi_p + \cos \rho \cos \theta_p. \quad (2)$$

If  $\cos \alpha$  can be estimated for different values of  $\phi_p$ , then Equation 2 can be used to solve for the coefficients  $\sin \rho \sin \theta_p$  and  $\cos \rho \cos \theta_p$ . The sum and difference of these coefficients are  $\cos(\rho - \theta_p)$  and  $\cos(\rho + \theta_p)$ , respectively, and, from these two quantities,  $\rho$  and  $\theta_p$  can be obtained.

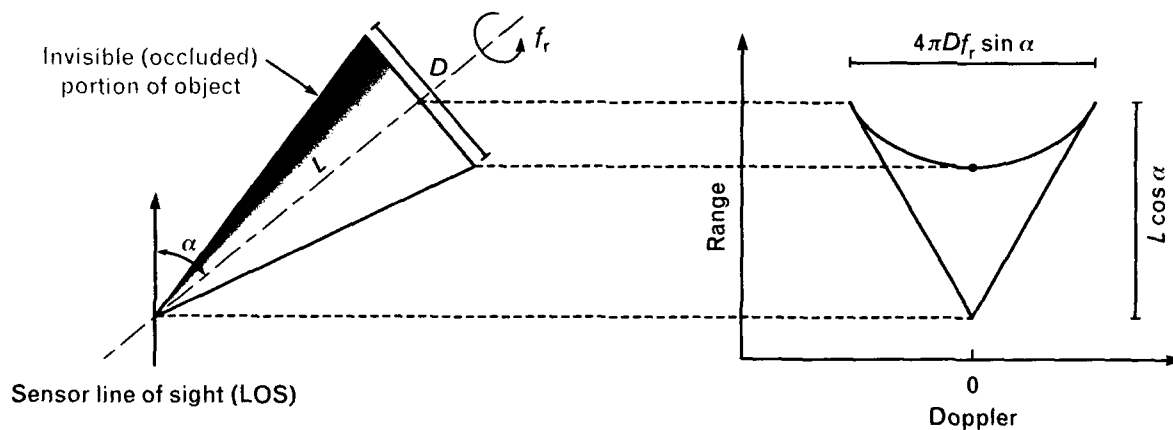
The target projected orientation  $\theta_0$  in angle-angle images is also a function of  $\phi_p$ :



**FIGURE 9.** Geometry of a target that is spinning (about the target body axis) and precessing (about the  $z$ -axis) simultaneously. The parameters  $\theta_p$  and  $\phi_p$  represent the precession half-cone angle and the precessing azimuthal angle, respectively, and  $\alpha$  and  $\rho$  represent the aspect angle and mean aspect angle, respectively.



**FIGURE 10.** Feature extraction for angle-angle images of a simulated spinning triconic body undergoing precessional motion: (a) input images, (b) edge extraction, (c) fitted data for  $\cos \alpha$ , and (d) fitted data for  $\tan \theta_0$ . The rotation rate chosen was  $80^\circ/\text{sec}$ ,  $\theta_p$  was  $16^\circ$ , the precession rate for  $\phi_p$  was  $10^\circ/\text{sec}$ , and  $\rho$  was  $90^\circ$ . The images were simulated with a 2-cm pixel size and a 2-sec frame time (i.e., a 2-sec interval between frames), with 19 frames generated over a complete precessional cycle.



**FIGURE 11.** Geometry of range-Doppler radar imagery of a spinning cone.

$$\tan \theta_0 = \frac{\sin \theta_p \sin \phi_p}{\cos \rho \sin \theta_p \cos \phi_p - \sin \rho \cos \theta_p} \quad (3)$$

At broadside viewing ( $\rho = 90^\circ$ ), Equation 3 reduces to  $\tan \theta_0 = -\tan \theta_p \sin \phi_p$ . Similar to the case of aspect-angle data, the orientation data can also be fitted with Equation 3 to determine  $\rho$  and  $\theta_p$ .

As an example, a series of simulated angle-angle images has been generated for a spinning triconic body undergoing precessional motion (Figure 10[a]). The rotation rate chosen was  $80^\circ/\text{sec}$ ,  $\theta_p$  was  $16^\circ$ , the precession rate for  $\phi_p$  was  $10^\circ/\text{sec}$ , and  $\rho$  was  $90^\circ$ . (Note: The LOS was assumed to be outside the precession cone.) The images were simulated with a 2-cm pixel size and a 2-sec frame time (i.e., a 2-sec interval between frames), with 19 frames generated over a complete precessional cycle. The triconic boundaries were identified by computer for each image frame (Figure 10[b]) so that the aspect angle  $\alpha$  and orientation  $\theta_0$  could be determined. Fitting these angles to Equations 2 and 3 then allows  $\rho$  and  $\theta_p$  to be obtained. For the aspect-angle history, Figure 10(c) shows that the best fit occurs for  $\rho = 91.4^\circ$  and  $\theta_p = 15.4^\circ$ . For the image-plane body-axis orientation (given  $\rho = 90^\circ$ ), Figure 10(d) shows that the best fit occurs for  $\theta_p = 16^\circ$ .

### Feature Extraction for Range-Doppler Images

For range-Doppler radars, targets with angular dynamics are imaged along the radar LOS in the two dimensions of range and Doppler. The projection of  $L$ , the length of a target measured along its body axis, onto the LOS is  $L \cos \alpha$  in the range dimension, with  $\alpha$  being the aspect angle, as shown in Figure 11. In the case of a rotating target, the rotational angular velocity  $V$  projected onto the LOS is measured in the Doppler dimension as  $V \sin \alpha$  [7]. With these relationships, the dimensions of a target can be derived from its image range and Doppler extents.

Application of this technique is demonstrated in this section with an example of simulated range-Doppler images of a diffuse cone. In the example, image processing and scene analysis are applied, as has been demonstrated earlier for angle-angle imagery. A target line model is then used to guide the formation of a target line representation in the imagery. Target di-

mensions and orientations are then determined from the range and Doppler extents of a sequence of the images.

### Estimation of Target Dimensions

For the cone-shaped target of Figure 11, the range and Doppler extents are related to the target aspect angle  $\alpha$  and the rotation rate  $f_r$  by

$$R \cdot dr = L \cos \alpha \quad (4)$$

and

$$V \cdot dv = 4\pi D f_r \sin \alpha \quad (5)$$

where  $R$  and  $V$  are the range and Doppler extents (number of image pixels),  $dr$  and  $dv$  are the range and Doppler cell sizes, and  $L$  and  $D$  are the target physical length and base diameter, respectively. Combining the two equations to eliminate functions of the aspect angle gives the following ellipse equation for a rigid target undergoing steady rotation with constant  $L$ ,  $D$ , and  $f_r$ :

$$\left(\frac{dr}{L}\right)^2 R^2 + \left(\frac{dv}{4\pi D f_r}\right)^2 V^2 = 1 \quad (6)$$

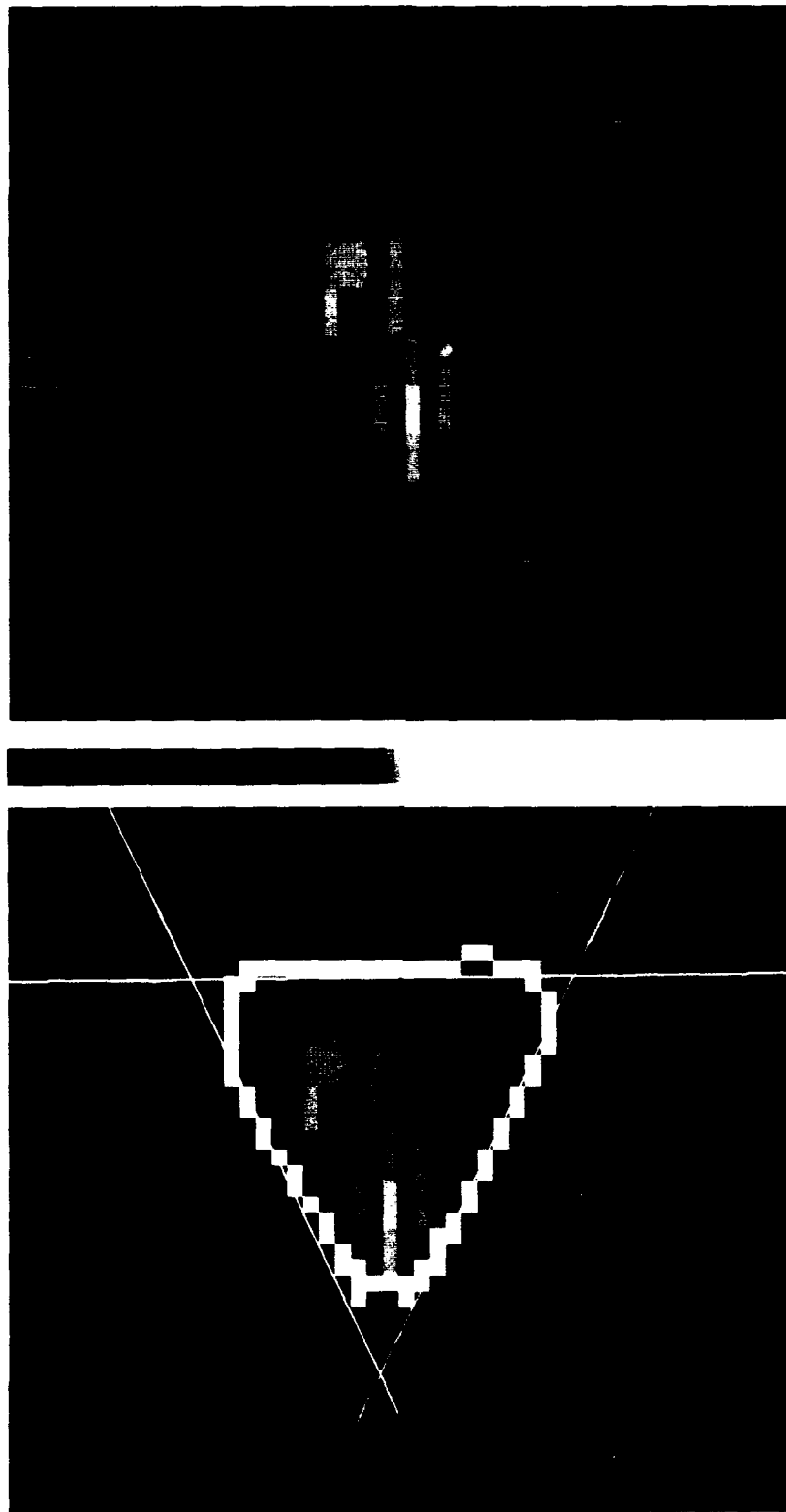
The major and minor axes of the above ellipse can be estimated for a dataset of  $(R, V)$  pairs by the least-squares-error method. The physical length  $L$  and the Doppler velocity can thus be calculated by

$$L = \sqrt{\frac{R^4 V^4 - (R^2 V^2)^2}{R^2 V^4 - V^2 R^2 V^2}} dr$$

and

$$4\pi D f_r = \sqrt{\frac{R^4 V^4 - (R^2 V^2)^2}{R^4 V^2 - R^2 R^2 V^2}} dv.$$

For cases in which the rotation frequency  $f_r$  is known, the base diameter  $D$  can also be determined from the Doppler velocity. Theoretically, the above derivation can be performed for any dataset with more than two pairs of  $(R, V)$  measurements, provided that the data noise or measurement errors are



**FIGURE 12.** Simulated range-Doppler image of a cone: (top) before processing and (bottom) after edge points have been detected and lines fitted. The color bar indicates increasing (from black to white) radar cross section.

much less than the true data difference corresponding to different aspect angles. For a large dataset, the measurements of range and Doppler extents can be median filtered separately to remove noise (while still retaining the temporal trend resulting from changing aspect angles).

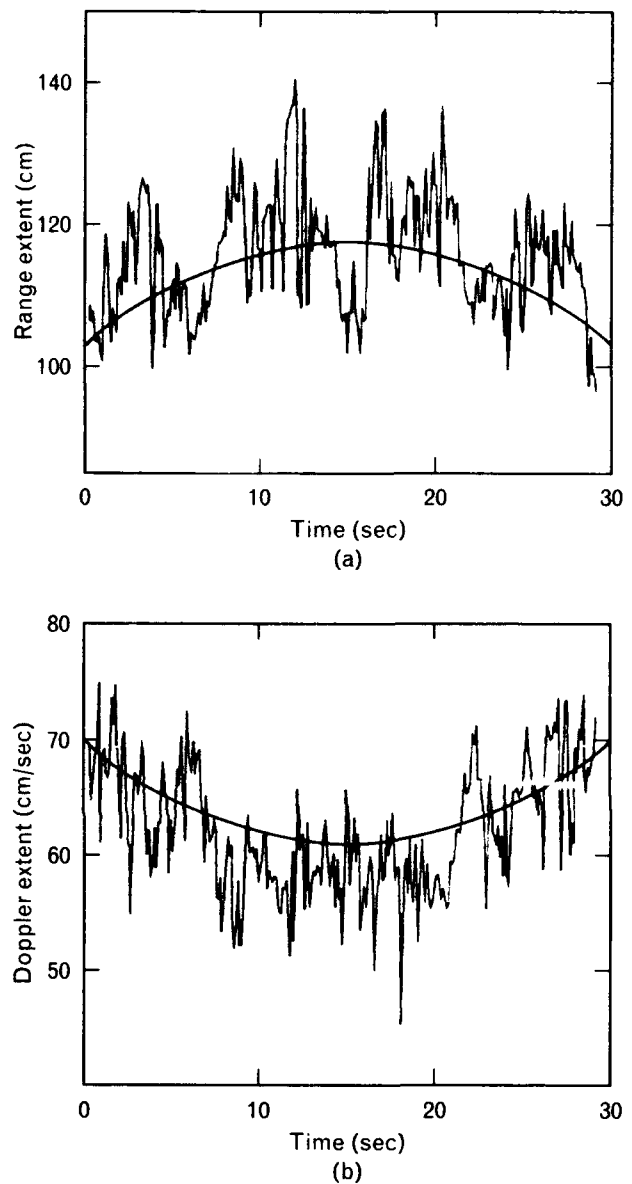
With both  $L$  and  $Df_r$  derived from a given dataset, the target aspect-angle history can be obtained from either Equation 4 or 5, or from the ratio of the two equations. It should not matter which equation is used to obtain the aspect angle because both of the equations were used to derive Equation 6, which was used to estimate both  $L$  and  $Df_r$ . More precise results, however, may be obtained by using the equation corresponding to better cell resolution.

#### *Motion Extraction for a Simulated Cone*

For purposes of demonstration, range-Doppler images have been simulated for a cone (length of 150 cm and a base radius of 19.7 cm) having a diffuse target surface. A total of 280 images was generated over 28 sec with the cone undergoing spin (spin period of 4.5 sec) and precession [8]. Each image was  $53 \times 53$  pixels, with a range cell representing 5 cm and a Doppler cell representing 2.93 cm/sec. Figure 12 (top) shows one of the range-Doppler image frames.

Image processing, edge detection, and line fitting were then performed on the images. For the simple known target shapes in the current application, the line fitting was simplified with a piecewise linear fitting of the target boundaries: the edges at the base were collected and fitted linearly, and the cone was defined by two lines fitting the cone edges and the base line. Range and Doppler extents of the target could then be estimated from the target line model. Figure 12 (bottom) shows a frame of the image with the edge points detected and the lines fitted. For such figures, the bisection line of the cone is the target-body centerline, the distance from the nose to the base center in range is the target range extent, and the Doppler spread between the intersections of the base line and the cone's left and right sides is the Doppler extent.

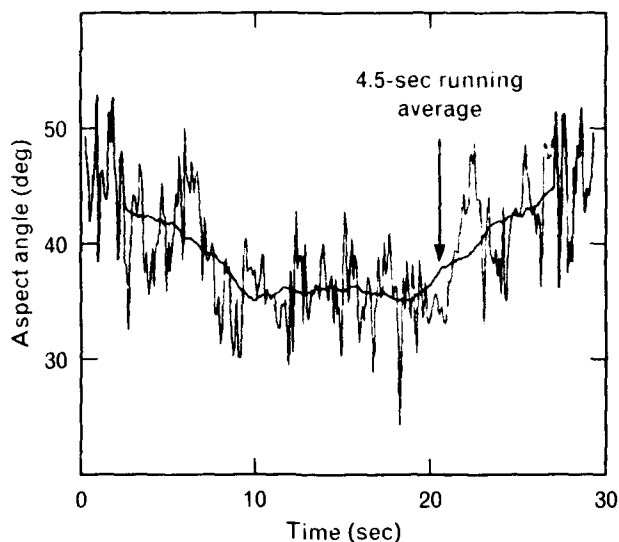
Figure 13 contains plots of the extracted range and Doppler extents from the image sequence. Note that, in spite of the considerable amount of local noise



**FIGURE 13.** The extracted (a) range and (b) Doppler extents from a sequence of simulated images (see Figure 12).

present in the imagery, the range and Doppler histories show functions of  $\cos \alpha$  and  $\sin \alpha$ , respectively, for the aspect angle  $\alpha$ , which changes because of the precession of the target. Target dimensions can be obtained by the elliptical fitting of the range and Doppler data, as described earlier in the subsection "Estimation of Target Dimensions." With such techniques, a value of 153.5 cm was calculated for the target length, and 17.3 cm for the base radius. Note that the derived target dimensions agree well with the





**FIGURE 14.** Aspect-angle history derived from the Doppler extents (see Figure 13[b]).

corresponding input model dimensions of 150 cm and 19.7 cm. Both image-resolution-related noise and target-surface-scattering noise could have contributed to the differences in the derived and input target dimensions. The quantitative impacts of such noise are currently under evaluation.

Figure 14 shows the aspect-angle history that was derived from the Doppler extents. A running averaging of 4.5 sec was applied to smooth the aspect-angle curve, which changes from  $38^\circ$  to  $47^\circ$  with a period of about 28 sec. With the assumption that the radar LOS was outside the target precession cone, the target precession half-cone angle can be calculated as  $4.5^\circ$ , which is comparable to the input angle of  $5^\circ$ .

### Summary

The extraction of target features—shape, size, orientation, and dynamics—from both angle-angle and range-Doppler images has been demonstrated. The target shape, size and orientation can be determined from single-frame angle-angle images, and the target angular dynamics can be estimated from a time series of such images. For the case of range-Doppler imagery, multiple frames of images are required to obtain multiple measurements of range and Doppler extents to enable the derivation of the target dimensions. A time series of range-Doppler images can also be used to estimate the target angular dynamics.

For simple known targets, the feature-extraction process can be simplified greatly by obtaining the target image description with a piecewise linear approximation of the target boundaries. For cases in which some size/shape parameters such as cone half-angle, length, and/or base diameter are available, the target size and aspect angle can be estimated from data extracted for each frame of range-Doppler images. Convergence of such estimations over multiple frames of images can then be used to evaluate the algorithm performance and to verify measurements for the known target.

The algorithm is undergoing extensive testing with simulated and field data, and both the precision and efficiency of the algorithm are expected to improve over time. Preparation for real-time implementation of the range-Doppler algorithm is currently in progress.

### Acknowledgments

The author would like to express her sincere appreciation to Kent Edwards for his suggestions, guidance, and support for this article. Special thanks are due to Marc Bernstein and Thierry Copie for their interest, encouragement, and continuous supply of simulated and field data. Without the simulated data, the assessments of algorithm performance would not have been possible. Without the field data, the algorithm could not have been made realistic and practical. The author also thanks Marianne Pietrzyk for her assistance in the development of the algorithm for the extraction of target features from angle-angle imagery.

This work was sponsored by the U.S. Army Strategic and Space Defense Command.

## REFERENCES

1. D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition* **13**, 111 (1981).
2. R.C. Gonzalez and P. Wintz, "Digital Image Processing," 2nd ed. (Addison-Wesley Publishing Co., Reading, MA, 1987).
3. P.R. Beaudet, "Rotationally Invariant Image Operators," *Proc. Intl. Conf. on Pattern Recognition* (Kyoto, 7-10 Nov. 1978), p. 579.
4. L.S. Davis, "Two-Dimensional Shape Representation," *Handbook of Pattern Recognition and Image Processing*, eds. T.Y. Young and K.S. Fu (Academic Press, Inc., Orlando, FL, 1986).
5. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations," *IEEE Trans. Elec. Computers* **EC-10**, 260 (1961).
6. K.S. Fu, ed., *Syntactic Pattern Recognition, Applications* (Springer-Verlag, Berlin, 1977).
7. A.M. Aull, R.A. Gabel, and T.J. Goblick, "Real-Time Radar Image Understanding: A Machine-Intelligence Approach," *Lincol. Lab. J.* **5**, 195 (1992).
8. T.B. Copie, private communication (Oct. 1992).



SU MAY HSU

received the following degrees in electrical engineering: a B.S. from the National Taiwan University and a Ph.D. from Purdue University, where her research focus was in syntactic pattern recognition. She later joined the Corporate Research and Development Center of General Electric Co. in Schenectady, New York, where she became involved in image processing and scene analysis in application to nondestructive part inspection for quality assurance. In 1981 she joined Lincoln Laboratory, where she is currently a staff member in the Signature Studies and Analysis Group, and her focus of research has been in scene analysis and signal processing.