REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	Dec. 31, 1996	Final Report, Sept. 1, 1993 to Dec. 31, 1996

4. TITLE AND SUBTITLE

Phenomenology and Learning-Based Configuration and Control of ATR Systems; ATR Design via Adaptive Configuration and Control

5. FUNDING NUMBERS

DAAH04-93-C-0050

6. AUTHOR(S)

Wing Au, Barry Roberts

7 PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Honeywell Technology Center
Honeywell Inc.
3660 Technology Drive
Minneapolis, MN 55418

8. PERFORMING ORGANBIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 32370.2-MA

11. SUPPLEMENTAL NOTES

The views, opinions, and/or findings contained in is report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12A. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The aim of the three year effort is the development of a three component system for Configuration and Control of ATR systems. The components perform three functions: 1) Intelligent ATR system Configuration and Training, 2) Intelligent Control and Scheduling of ATR processing, and 3) Calibration of model of IR image formation.

The components are named the *Learning Classifier, Cognitive Executive, and Phenomenlogical Model Calibration* systems respectively. This contractual effort is in support of the Reconnaissance, Surveillance, and Target Acquisition (RSTA) component of DARPA's Unmanned Ground Vehicle (UGV) program.

14. SUBJECT TERMS

Adaptive Automatic Target Detection, FLIR Image Processing Learning Classifier System Bayes Nets, Adaptive Configuration and Control

15. NUMBER OF PAGES

16

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

ATR Design via Adaptive Configuration and Control*

Wing Au, Barry Roberts
Honeywell Technology Center
3660 Technology Drive, Minneapolis, MN

Abstract

This paper presents an ATR design paradigm that self configures and adapts to the diverse scenarios encountered during a mission. The technical approach is described, and the results of a self-evaluation are summarized.

Today's ATR is constructed via inefficient and sub-optimal system configuration and training, whose process is very labor intensive, subjective and inaccurate. The resulting ATR is only capable of a limited amount of adaptation to changes in the environment. Moreover, the operation of such ATR systems require a user with expert algorithmic knowledge.

Addressing the above-mentioned problems, the Honeywell effort is producing a self-adaptive ATR system. The system employs a Genetic Algorithm to autonomously and optimally perform configuration and training; the system also includes a specific knowledge capture mechanism, the Context Capture tool, which ties the context of the mission with the optimal configuration. Lastly, the system employs Selective Perception software to dynamically configure and control the ATR system based on the changing context during an ATR mission.

*. This effort is supported by the Army Research Office under contract DAAH04-93-C-0050 with funding provided by Defense Advanced Research Projects Agency.

1 Introduction

The military community has not accepted Automatic Target Recognition (ATR) systems as being viable and fieldable due to two key deficiencies: (1) the lack of reliability or robustness of such systems and (2) limitations in the ***usability*** (ease of use) of the systems.

Today's ATR is not highly robust because it does not work under diverse scenarios. During a targeting mission, the scenarios and conditions of the battlefield are constantly changing. For example, the terrain could be flat or rolling; the ground cover could be sparse to dense; the time of day could be morning, noon, or night; the weather and environmental conditions could be anywhere between clear to cloudy, dry to humid, cold to hot, etc. A robust ATR system must recognize targets of unknown operational history (i.e., clean to dirty, hot to cold) under all these conditions. There have been numerous attempts to develop robust ATR systems but all such attempts have met with limited success. In the authors' opinion, the reason for the limited success lies with the limited amount of adaptability that the extant systems have with regard to the scenario and conditions of the mission.

Current ATR algorithms are only capable of limited adaptation to changes in context during a mission. In other words, each ATR algorithm, as conceptually illustrated in Figure 1, works well under a specific set of scenarios, but fails miserably in others. If an ATR system could execute the proper ATR algorithm for each given scenario, then

a truly robust *ATR system* could be achieved. In addition, today's typical ATR algorithm requires the correct setting of multiple parameters. Each setting corresponds to an optimal operating point for a specific image context; poor setting of these parameters causes the failure of the ATR algorithm. Thus, to properly change these parameter values to achieve good performance requires expert knowledge in algorithm operation and image understanding technology, which the typical ATR user does not have. Moreover, ATR users often have many other critical activities to which they must devote their attention rather than the tuning of the parameters of the ATR system.

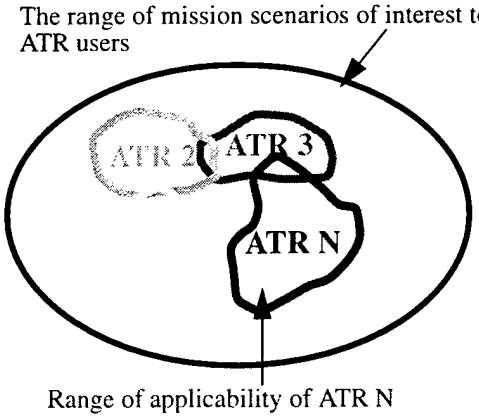


Figure 1: A static ATR algorithm only performs well under a limited range of imaging scenarios.

This poor usability prevents the acceptance of ATR system by the potential user.

The poor reliability and usability are inherent in the traditional ATR design philosophy. Today, an ATR is constructed using inefficient and sub-optimal system configuration and training. The process of configuring an ATR is currently very labor intensive, subjective and inaccurate, as is the process of training an ATR for a particular mission. Neither process ensures that an optimum ATR system has been produced. This problem is attributed to the large search space (the numerous algorithms and the uncountable parameter values) to be examined to find the optimum ATR system solution. Also, the amount of adaptability in ATR systems is currently limited to the modification of the values of key algorithm parameters within one static algorithm suite.

To improve the reliability and usability of an ATR system, this paper presents a new paradigm in ATR design. What is needed to solve the ATR problem is not more algorithm development, but rather the intelligent selection and application of algorithms (and their parameter values) to best meet the scenario at hand.

Figure 2 illustrates the authors' view of an ATR system as a collection of many vision modules working in concert throughout the phases of ATR

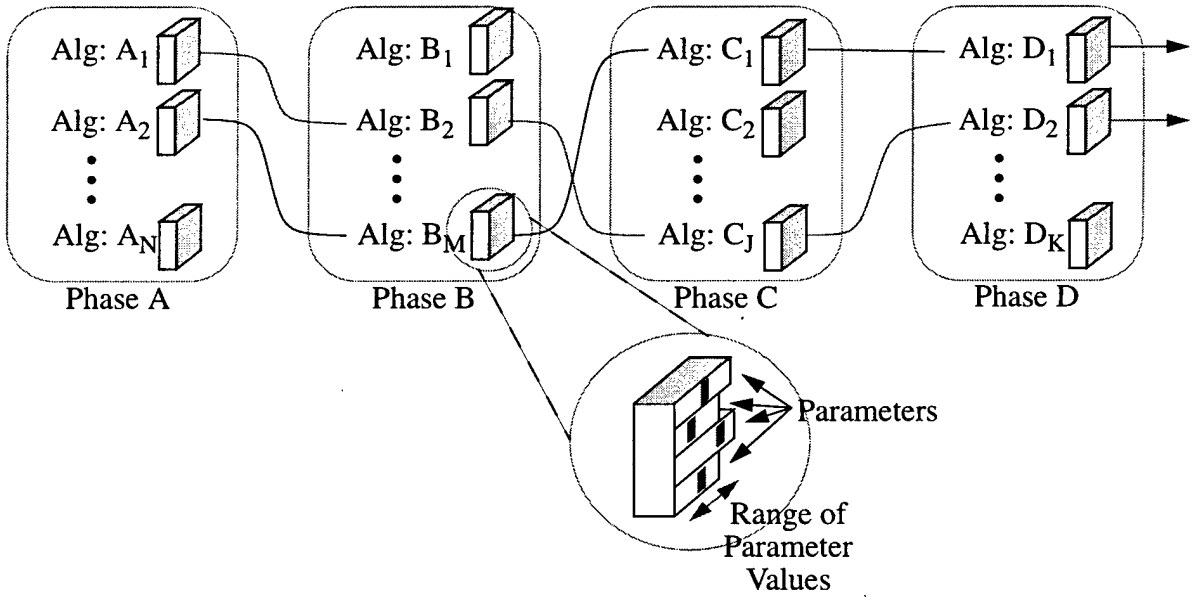


Figure 2: Block diagram of the operation of Configuration and Training as achieved with a *Learning Classifier* system and its *Genetic Algorithm*.

operation that are required to detect, classify, and identify targets. Within each module, multiple algorithms can be employed to implement the module; each algorithm has its own set of parameters that can be applied to optimize its own performance based on the context/characteristics of the input data (ancillary and sensor). Selection of the algorithms and their parameter values (called configuration and training, respectively) under the context of diverse scenarios, becomes a combinatoric problem. Under a particular battlefield scenario, one configuration will likely work better than all of the rest, but no one configuration will be the best for all scenarios.

To improve the adaptability of an ATR system and thereby improve accuracy, robustness, and user acceptance, ATR needs an automated configuration and training tool that can be employed for (and/or applied to) the various mission scenarios. The knowledge gained from using such a tool should then be stored (using a second automated tool) in a knowledge representation that is indexed by the "context" of the mission scenario. Hence, with a third tool for probing the knowledge representation using the context of a current mission, an ATR configuration tailored to the mission (as determined by the configuration and training tool) can be selected and applied to the current sensor data. Hence, with this triad of tools, a self-adaptive context/mission-dependent ATR system is formed that holds the promise of robust performance that is acceptable to the user community.

In the remainder of this paper, the self-adaptive ATR system concept and its three fundamental components are described. The progress to date and the future directions are also presented.

2 System Overview

Our work, which develops these tools (Figure 3) for self-adaptive, configuration and control of ATR algorithms and their parameter values, provides optimal ATR performance over diverse scenarios while saving significant time in performing configuration and training.

The first tool in the triad is the Configuration and Training tool implemented by a *Genetic Algorithm* (GA). GA produces/learns new configurations of ATR algorithms and trains them to produce optimal levels of target recognition performance

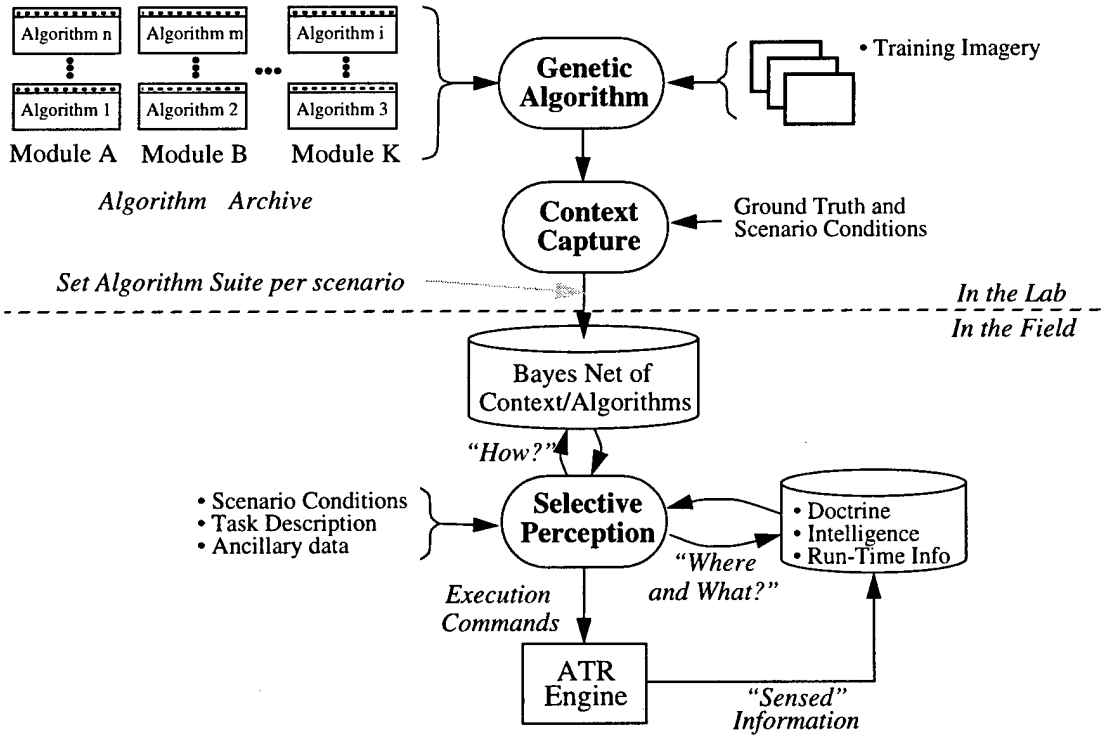


Figure 3: High-level diagram of the Self-Adaptive ATR *system* being developed.

under given scenarios. The GA operates on a library of "building blocks". The building blocks used in the Genetic search are executable, component algorithms of an ATR algorithm suite and simple descriptions of each algorithm identifying its syntax and functional category (e.g., preprocessing, segmentation, clutter rejection, etc.). Note that source code is not needed, so proprietary algorithmic information is preserved.

The second tool, which performs the knowledge (i.e., algorithm and context) capture, is the *Context Capture* (CC) tool. For self-adaptation, the ATR system needs to learn what optimal configuration to execute under a specific condition. Therefore, CC, first down-selects the key context types from a pool of available types. The context consists of the conditions of the scenarios (i.e., ancillary data and sensor data metrics). The down-select is based on the linear discriminant analysis or Karhunen-Loeve (K-L) transform. Next, CC associates the optimal configurations with the context extracted from the input images and ground-truthed information. The output of the knowledge capture process is a database (Bayes Net) that maps the key context and specific tasks to ATR algorithm suites and their parameters.

Note that these two tools perform the functions that correspond to the traditional training process. Therefore they operate in an off-line, pre-mission environment. Still our design allows distributed processing on a network of computers, thereby speeding up the training process.

The third tool, known as *Selective Perception* (SP), is responsible for configuration and control of the ATR engine in field operation using "context" acquired during the mission to decide which algorithm suite (plus parameter values) is best for the current situation. Selective perception chooses tasks using a priori doctrinal and mission information, information acquired as the mission proceeds, and knowledge of configured and trained ATR algorithms under various operational scenarios. The choice is based on Bayesian probabilistic cost and benefit analysis. The analysis produces an action (what visual action to perform next, where in the scene to apply it, and how to implement it) to be executed by the processing engine.

Honeywell's adaptive system has been designed for ease of use from the beginning. For example, it is very easy to describe to the GA system the algorithm components on which it is to operate. This ease of use facilitates the transfer of the system and the growth of the system via additions to the Algorithm Archives (see Figure 3). Transfer of the GA system involves the transfer of C++ source code and its associated documentation. Operation within the Khoros environment by both components of the self-adaptive ATR system, helps to simplify the technology transfer process.

Simple interfaces allow easy addition of visual capabilities (i.e., image understanding and ATR algorithms), such as those developed by others in the RSTA community, into the population for Genetic search and/or into the Selective Perception module. The same interfaces facilitate the transfer and integration of the system at other sites.

Note that our approach works for any ATR system regardless of its sensor modality; thus our approach is applicable to SAR-, FLIR-, and LADAR-based ATRs. The context of these different ATR types would be different due to the different sensing phenomenology. However, the Context Capture is able to acquire various types of context.

3 Genetic Algorithm System Overview

The process of configuring an ATR is currently very labor intensive, subjective and inaccurate, as is the process of training an ATR for a particular mission. Neither process, performed manually, will ensure that an optimum ATR system has been produced. This problem is attributed to the large "solution" space that must be examined to find the optimum ATR system solution. The large solution space is a result of three key dimensions of ATR algorithm suite variation:

1. many vision modules working in concert throughout the phases of ATR operation,
2. multiple algorithms that can be used to fulfill the requirements of each module, and
3. the range of values that can be taken by the parameters associated with each algorithm.

It is the combinatorics of this search problem that make the human-generated solution

impractical in producing a robust ATR under diverse mission conditions.

Honeywell's RSTA effort has produced a software system for autonomous configuration and training of ATR algorithm suites. The system employs a *Genetic Algorithm* (GA) [Booker et al, 89; Holland, 75], which automates the search process through the large space of ATR solutions. The GA is needed to intelligently and autonomously search the space of ATR modules, algorithms, and parameter values to find the optimum (i.e., best performing) ATR for the available training imagery.

Inherently, the GA system is exceptionally well suited to this search problem for multiple reasons:

- it can handle the combinatoric of the problem,
- its processing can be performed on a network of distributed processors,
- it does not require algorithm-specific knowledge or source code,
- it does not require domain-specific knowledge or any type of performance measure surfaces,
- it is not deceived by local (i.e., non-global) maxima in the ATR performance measure.

The GA system efficiently searches the enormous hyperspace of module, algorithm, and parameter value combinations using a collection of points within the search space known as a *population*; each solution point within the population is called an *individual*. The operation of a GA system is characterized by three core concepts:

- a "Darwinian" notion of fitness or strength which determines an individual's likelihood of affecting future generations through reproduction,
- a reproduction operation which produces new individuals by combining the best members of the existing population, and
- genetic operators which create new individuals based on the structure of their parents.

The operation of the GA system is iterative; during each iteration, called a *generation*, each individual is *computed/executed* and *evaluated*. The process of evaluation involves the computation of an overall quality or utility, called *fitness*, for each individual. On the basis of the fitness of the individuals in the population, *reproduction* takes place. The entire evolutionary cycle (i.e., execution, evaluation, reproduction, and population maintenance) is illustrated in Figure 3.

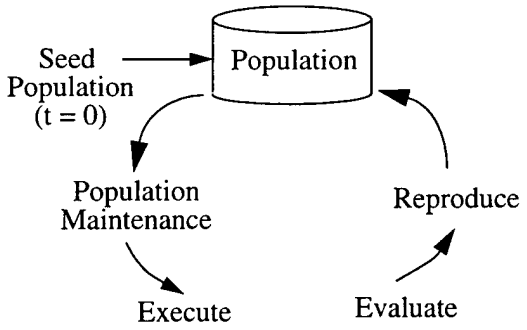


Figure 4: High-level, conceptual diagram of the iterative operation of the *Genetic Algorithm*

The inherent power of a GA system lies in its ability to exploit, in a highly efficient manner, information about a large number of individuals. In reproduction, individuals with high fitness are allowed to combine with other high fitness individuals through the processes of *cross-over* and *mutation*. The "offspring" from this activity, replace the low fitness individuals within the population and are subsequently evaluated in the next iteration. The process of cross-over interchanges components of the "parents" to create two offspring which have components from each parent. By allocating more reproductive occurrences to above average individuals, the net effect is an upward shift in the average fitness of the population. In this manner, the GA system is focussing on productive regions of the search space and thereby making intelligent use of processing resources. The mutation process randomly modifies the components within individuals. This reintroduces diversity into the population (by producing outliers) to keep it from stagnating within one area of the solution space. By keeping the population distributed throughout the solution space, the genetic search effectively eliminates the problem of converging to local (i.e., non-global) maxima. A detailed block diagram of the operation of the GA system is provided in Figure 5. The GA system requires

- knowledge of the modules of the ATR and their interconnection,
- knowledge of all the algorithms available for use within each module, and
- knowledge of the parameters (and their respective range of values) used by each algorithm.

It is a simple matter to define these items of information for the GA system. This is accomplished by producing a text file containing the *Interface Specification* whose simple format is described in [IUW94, BR]. It is important to note that the GA system does not require source code for the algorithms or an understanding of the operation of the algorithms (hence proprietary information need not be disclosed by the provider of algorithms). Also, the GA system can be adapted to operate on any ATR architecture (i.e., any combination, connection, or order of modules).

Program execution of an individual is carried out through the use of an *Output Interface* which, for now, creates a UNIX™ shell file that can subsequently be executed by the system. Recall that an "individual" is a complete ATR algorithm suite with a particular set of parameter values. Within the GA system, the command line that can execute each algorithm within an individual is constructed from the internal description of the algorithms in the algorithm archive (see [IUW94, BR]). The concatenation of the respective command lines is used to form the shell file.

The *Evaluation* of each individual involves the computation of a performance metric that characterizes the results produced by the individual. The operation of the GA system is not dependent on the choice of performance metric(s). The computation of any performance metric is performed by an executable function, which is called by the GA system; the system will operate with any desired metric(s). Undoubtedly, the metric or metrics (e.g., the probability of detection and false alarm rate) will rely upon the ground-truth associated with the training imagery.

The execution of ATR shell files and the associated evaluation command are distributed to a network of UNIX workstations. Thus, the speed of configuration and training of an ATR suite is increased greatly.

The product of the GA system (after operating on one or more training images which characterize a particular mission scenario) is a trained and configured ATR algorithm suite which performs best in that scenario. Note that it is only necessary to train on the scenarios that are expected to be encountered during a mission.

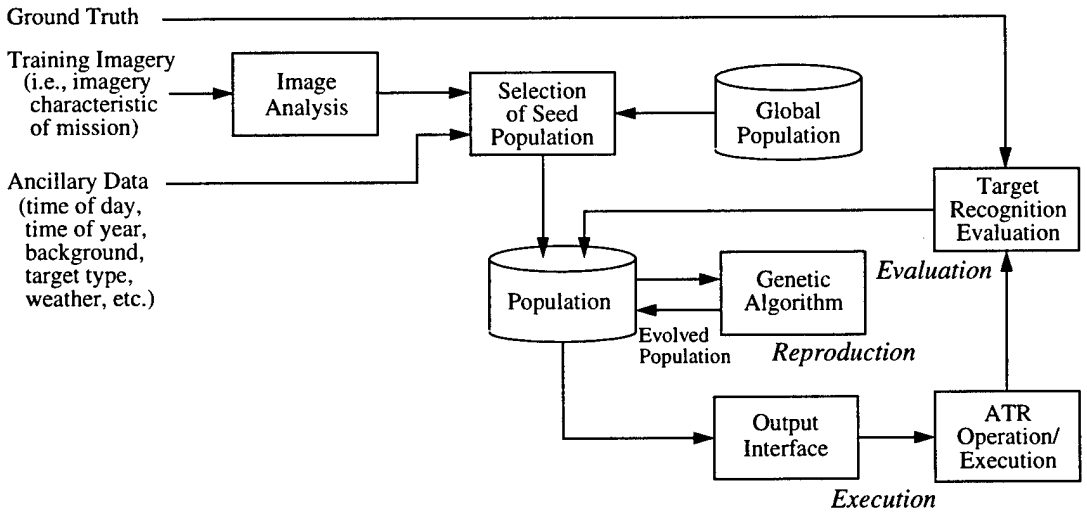


Figure 5: Block diagram of the operation of Configuration and Training as achieved with a *Learning Classifier* system and its *Genetic Algorithm*.

4 Context Capture System Overview

For self-adaptation, the ATR system must learn what optimal configuration to execute under each class of mission conditions. The goal of *Context Capture* (CC) system is to build a knowledge database that the self-adaptive *ATR system* can use to configure and control the ATR processor(s) during in-the-field operation.

CC's objective of providing a software mechanism for knowledge representation (for storage of the relationship between environment and sensor characteristics *and* the best performing ATR algorithms and parameter settings) is accomplished by performing three key tasks:

- Identifying the key context and quantizing the range of context into discrete bins,
- Associating the best ATR configurations with the context of the input scenarios,
- Building the Bayes net structure.

Identifying the key context that the adaptation is based upon is critical to the success of the self-adaptive ATR system. Since the context is used to cue the optimal configuration, an erroneous context description will mislead the Selective Perception process causing it to choose an incorrect algorithm configuration and parameter values.

Initially, a context set was selected that describes the environment and mission description. This context set, which could be easily acquired or measured, includes time of day, background type, (e.g., open field, desert, mountain, rough terrain), vegetation type, temperature, cloud coverage, expected range to target, etc. However, we found that the context had a broad coverage. That is, many input images fit the same context description, but resulted in many different, optimal, ATR algorithm configurations. Hence, the initial context set was not suitable for indexing the optimal algorithm configurations.

Then a set of image metrics were chosen and added to the context set for the ATR system. The image pixel intensities are assumed to be random with a distribution which is well defined by its statistical moments, such as mean, standard deviation, skewness, and kurtosis. The distribution of the energy throughout the image is measured using the Shannon entropy measure. A contrast measure was also included as one of the image metrics. All of the metrics are computed over the entire image. Often images may be characterized by certain local peculiarities due to the image formation process and/or diversities in the captured scene context. It is more appropriate to consider regions of the images, preferably where targets of interest reside. In the metric computation process, certain local metrics are computed using multiple sizes of windows (to account for the possible range of target size). All computed metric values are included in the metric set. It is also important to include in the metric set some of the statistics on the distribution of the metric values that are computed for multiple windows.

Additional measures that are considered are based on relative entropy in different regions of an image. This cross entropy, also known as the Kullback Leibler distance, measures the spread of the energy with respect to neighboring regions in the image. Other measures would be based on the relative contrast measure rather then entropy, using a similar concept.

This large set of context (image metrics) demands intensive computations. It would be desirable to reduce the number of context variables to a key set that still fully captures the input characteristics, and whose values yield the greatest distinction between different input context. One approach is to use the Karhunen Loeve (K-L) transformation. The space into which the K-L transform maps the data is defined by the eigen-vectors of the matrix of second order statistical moments of the training data. Two key characteristics of K-L transform are that (1) the components of the transformed data are un-correlated, and (2) the information contained in the data is compressed/represented into a small number of K-L axes. It is this second characteristic that we would like to exploit for context capture.

Another technique to reduce the number of image metrics is based on Linear Discriminate Analysis. The computation is even simpler than finding the coefficients of the linear discriminate. The method can be optimized in the following way. Given $\mathbf{K}$ images associated with $\mathbf{K}$ algorithms. Each image represents a class of an event with a probability p_k,

and a mean vector equal to the $nx1$ predetermined parameter vector $\hat{\mu}_k$. We must compute within-parameter discrepancy to be compared to a set of correlation thresholds. Define S_b, to be the between-images scatter matrix (S_b represents the correlation matrix of these predefined parameters).

$$S_b = \sum_{k=1}^{K} p_k(\hat{\mu}_k - \hat{\mu}_o)(\hat{\mu}_k - \hat{\mu}_o)^T$$

where

$$\hat{\mu}_o = \sum_{k=1}^{K} p_k \hat{\mu}_k$$

All probabilities are assumed to be equal to one, since we are interested in finding the correlation between the parameters rather than the images. The simplicity of this approach is that there is no longer the need to compute the *Karhunen-Loeve Transform* that would cost $O(n^3)$

By setting a threshold on the correlation measure, a list of m << n parameters is identified which are the least correlated from the entire list of size n.

Once the key context set and the discrete ranges for each member are identified, the context set of the scenario under which the training images were acquired will be computed and stored along with the optimal configurations. A one-level Bayes net is used to store the knowledge. The Bayes net is used during in-the-field ATR system operation as a mapping of key context values to ATR algorithm suites and their parameter values.

5 Selective Perception System Overview

The *Selective Perception* (SP) system provides decision analysis algorithms that choose the proper ATR algorithm, the values for its parameters, and where to apply it in the scene. Hence SP is the inference engine for the self-adaptive ATR system during its field operation.

The intelligent control of vision applications is a key research issue facing the community today. Selective and goal-directed perception is becoming increasingly sophisticated. Rimey's Thesis work [Rimey, Brown, 91] demonstrates data structures and algorithms for knowledge representation using Bayes Nets and also structures and algorithms for continuous decision-making to minimize the cost of information gathering while maximizing its benefit for the current task. Rimey's techniques, perhaps streamlined, will be directly applicable to the ATR problem and can be so applied without major modification. Early pilot studies indicate that this idea is justifiable.

Two sorts of Bayes nets are used in the Selective Perception module; in both cases the nets are restricted to be trees. One type of net incorporates knowledge about *how* a particular RSTA task is to be performed; what probabilistic sub-goals or 'sub' random variables affect the belief in the top-level random variable (proposition, labeling problem). For a multi-task mission, there would be several such "task nets," one for each component task. The *how* question is answered by conditional probability tables that give the probability of each ancestor label, given each label for the descendent.

The other net encodes location information about the scene, telling where in the scene to look to find specific objects associated with random variables in the other net. One such "location net" could support several separate visual tasks. As the task proceeds, locations are usually known with more accuracy and their location, again through conditional probabilities in the location net, can constrain the locations of other objects. Constraining the location and probable identity of objects means that quicker and more robust visual tests can be applied. Vision is faster because a smaller area is to be covered, and can be simpler because the increased probability of identification by ancillary cues is propagated through the Bayes net and the identification problem can take place within a substantial context rather than from a position of complete ignorance.

Selective Perception processing is influenced by expectations about the mission, whether from general knowledge of physics and sensor characteristics, from mission planning information that predicts where the sensing agent will be and under what circumstances, or from knowledge about the mission that is picked up along the way. Mission planning information could be reflected in the perception module (e.g., daylight vision tasks might not be loaded for a night mission), and in the sequence in which they might be expected to arise,

and in the probabilities that govern each one (e.g., certain vehicles might be more or less likely on a route, depending on other aspects of the plan like time of day that the agent arrives at the route).

The Selective Perception software package contains a small Bayes Net package, which is a specialized system for Bayes Net manipulations, probably only a few percent of the size or functionality of currently available free-ware or commercial BN packages. However, the idea is that it will allow real-time operation. The Decision-making package is unique and, coupled with the Bayes net package, will be useful to anyone desiring to develop or run selective perception or goal-directed perception algorithms. Its tech transfer involves the transfer of C and/or C++ source code and its accompanying documentation which is sufficient for anyone to begin to use it productively.

Figure 6 shows two interfaces to the Selective Perception system. ATR algorithm suites are produced by the Genetic search or provided from elsewhere in the community. An estimate of their cost (say time per pixel of image) is needed if cost-benefit analysis is to be meaningful. The output of an algorithm must be translated into probabilities for a random variable representing the interpretation of the sensor's output. Also for each evidence-gathering task that the module can perform, a set of conditional probabilities P(H / E) or P(E / H) is needed that state the likelihood of the random variable having value H when the output of the sensor is E, or vice versa.

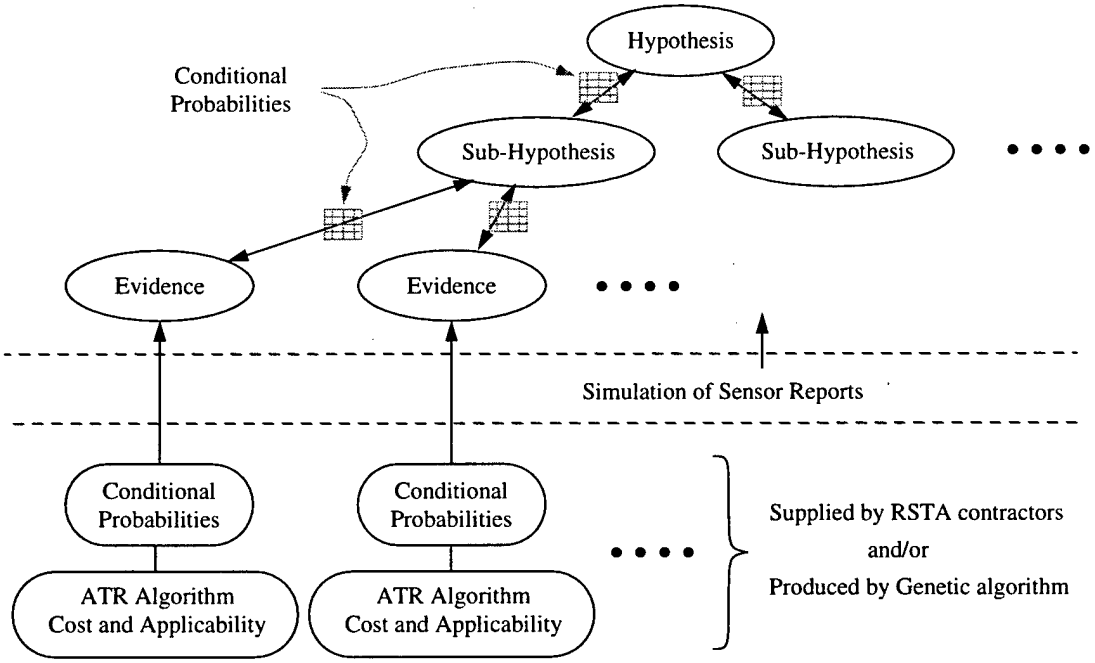


Figure 6: *Selective Perception* and its interface with algorithms/actions.

The dashed lines in Figure 6 represent the separation or independence that exists between the Selective Perception work and the GA work (or the outside world in general). The input to Selective Perception is not the image-level output of a processing step, but is a probability of a detection of an event. Current ATR algorithms often are derived from, and have outputs reflecting, probabilistic considerations (minimum-entropy decision trees, for instance). As a result, the output of the sensor processing can be interpreted as a probabilistic statement about the world. The Selective Perception tool expects output at the level of a probability density function over a hypothesis. Also, this abstraction of the physical sensor data to the level of logical sensor data allows simple simulation of sensor processing output: a simulated sensor reading is simply a probability density function over some hypothesis. Thus Selective Perception strategies can easily be developed with simulated sensors of more or less accuracy, performance, discriminating power, etc.

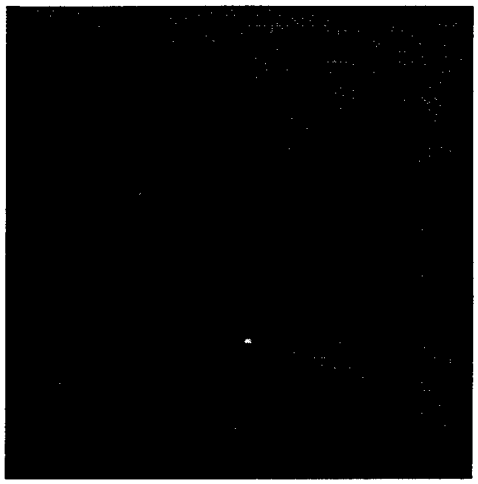
6 Results and Evaluation

The GA, CC and SP components have been implemented. The current version of CC is a preliminary version that accepts all the given context, which is a set of image metrics currently, and stores the context–configuration pairs in a Bayes Net. These three components are integrated into a *self-adaptive ATR system.*

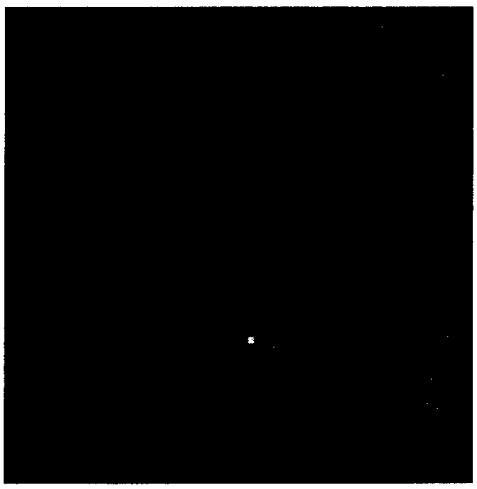
A variety of FLIR-based ATC algorithms are available in-house at Honeywell. For example the system's algorithm archive currently uses the MTAP algorithm suite as developed under Night Vision Lab funding and the Mine Detection algorithm suite as developed under funding from the Marine Corp. In addition, one ATR algorithm suite of a RSTA contractor has been added to our algorithm library. This has been sufficient to demonstrate the capability of our system. Note also, that the system has been designed to make it easy to incorporate other algorithms into our algorithm library (as they become available) such that there will be an even larger library to search in the future.

Figure 7 shows the results of the GA in configuration and training of an ATR system on an image. Figure 7a shows the original image. Figures 7b, 7c, and 7d show the best results of the initial, first and third generations of the ATR

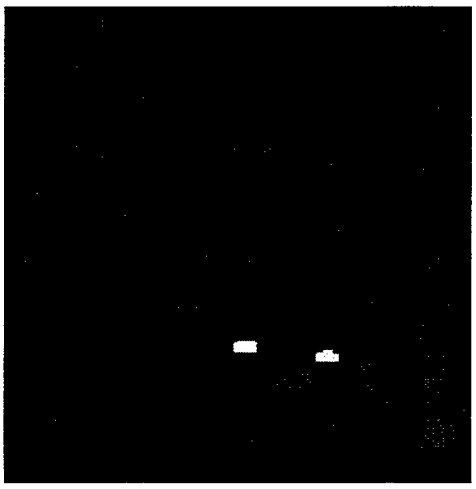
7a: Original Image



7b: best result of initial generation



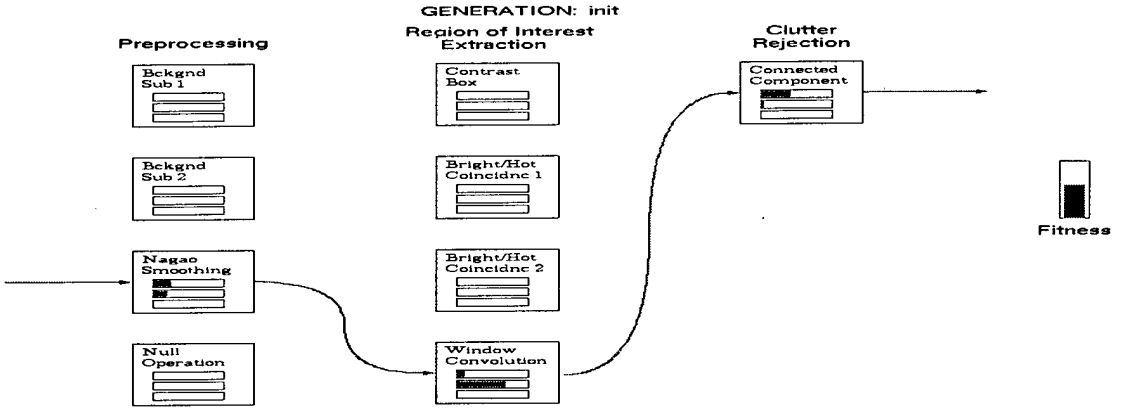
7c: best result of first generation



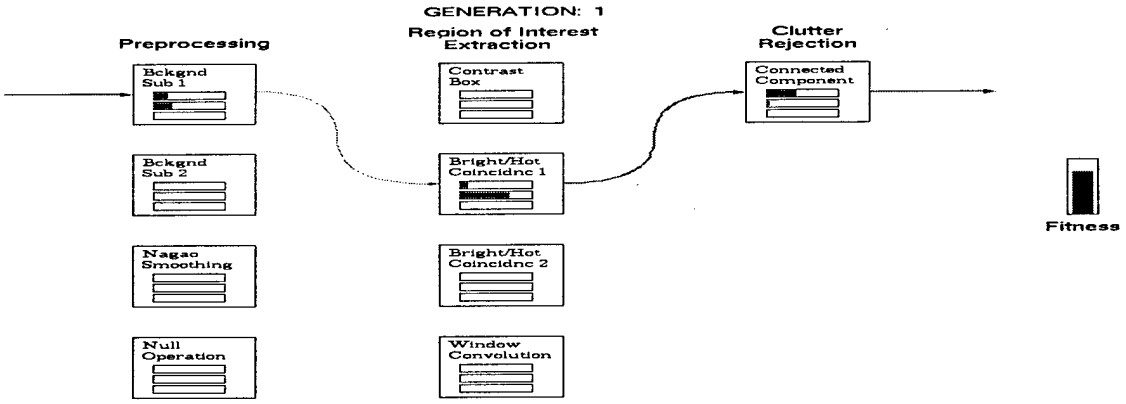
7d: best result of third generation

Figure 7: *Generations of results of Genetic Algorithm in optimal ATR system configuration*

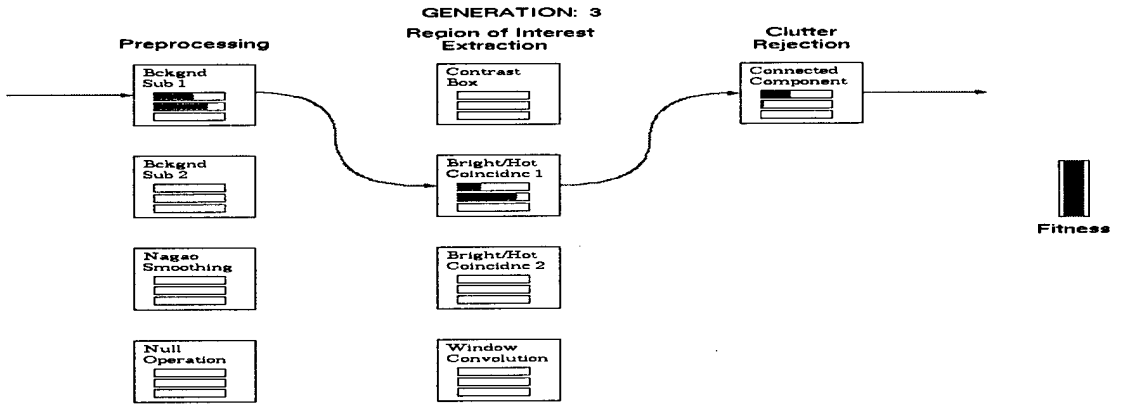
systems. Figures 8a, 8b and 8c show the corresponding ATR systems and the fitness values. Note that the GA selected different algorithm configurations in each generation. Note also that the fitness values of the population increase as the number of generations increased. When the fitness value reaches 1.0, indicating that all targets were detected without false alarm, the GA stops.



8a: Initial ATR configuration



8b: First Generation ATR configuration



8c: Third and last generation ATR configuration

Figure 8: *Progressively Improved Performance in Generations of ATR Configurations by GA*

At Demo C of the UGV program, the self-adaptive ATR system was demonstrated in one of the technology demo sessions. The GA was used to configure and train on a set of sample images. At Demo C, the self-adaptive ATR system was shown to successfully detect targets in a input image (that was collected "live" from a UGV vehicle), without any false alarms. As a comparison, the same image was processed by a baseline ATR suite; the baseline suite only detected one of the two targets that were present.

Subsequently a small scale evaluation was conducted, based on a comparison of the self-adaptive ATR system with two baseline ATR algorithm suites. The algorithm archives of the adaptive ATR system included the two baseline algorithms and another in-house ATR algorithm suite. The end product of the ATR processing during the evaluation was target detection. Three phases (modules) of the ATR processing include (1) preprocessing or preconditioning of the data, (2) region of interest extraction or focus of attention, and (3) clutter rejection. A total of 16 configurations for the ATR and uncountable choices of parameter values are possible. Both the adaptive ATR and the baseline suites were *trained* by the GA tool. The difference was that adaptive ATR was also *configured* by the GA tool.

The training samples consists of 40 images from 7 different data collections (the vertical dotted lines in Figures 9-11 partitions these 7 sets). These data collection included the data from UGV Demo C (Lockheed Martin at Denver), Multi-sensor Function Fusion (MSFF) program, Carson database, Huli9204, Huli9306, and Yuma9202 databases. The testing consists of 90 different images from the same 7 databases. The context of these data varies widely, since they were collected with different sensors (resolution, FOV, and other characteristics), at different locations, and at different time of the day and year.

The fitness function, which is a weighted sum of the probability of detection and the inverse of the false alarm rate, was computed and plotted as part of the evaluation.

Figure 9 shows the configuration and trained results of the self-adaptive ATR system. Figures 10 and 11 show the training results of the baseline 1 and baseline 2 ATRs respectively. In all cases, except one, the self-adaptive ATR yielded better or the same fitness values as compared to that of the two baseline ATRs. That is, the GA configured

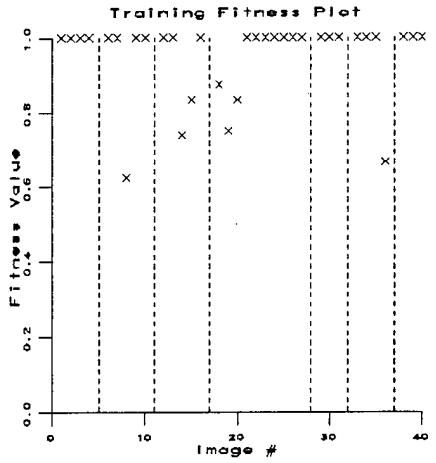


Figure 9: Configuration and training result of the Self-Adaptive ATR system.

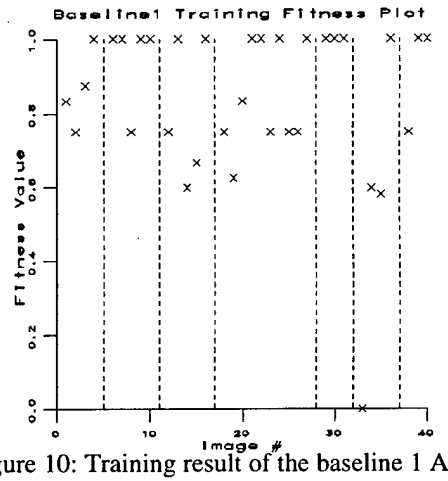


Figure 10: Training result of the baseline 1 ATR system.

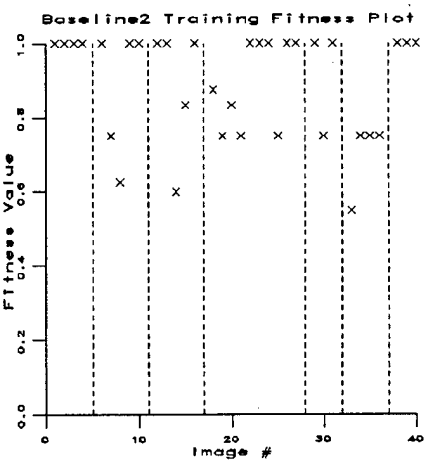


Figure 11: Training result of the Baseline 2 ATR system.

better ATR systems for all 7 data sets, whereas the baseline algorithms performed well only in some data sets.

A large scale self-evaluation was also conducted. Three data sets from Carson, the LMC Sept. 95, and NVL-amber were used. These three data sets covered a diverse scenarios (with various clutter levels, ranges, and time of day). The numbers of training and testing images and the numbers of targets are listed in Table 1.

Table 1: Self-Evaluation Data Sets

Data Set	Training Images	Testing Images	Testing targets	Total images
Carson	15	60	101	75
LMC	92	370	599	462
NVL	122	222	346	344
Total	229	652	1046	881

A context capture experiment was conducted. Thirty six context measures, which included range, time of day, and thirty four image metrics, were measured. From these thirty six measures, twenty four and fifteen most un-correlated metrics were selected. These three context sets were used as indices for the adaptation in three tests. The results indicated that the set of twenty four measures performed the best with comparable detections but less false alarms. We concluded that an optimal context set exists and the correlation and testing method to find the optimal set should work. For the subsequent evaluation, the twenty four measures were used.

For comparison purpose, a baseline algorithm suite was selected. The parameter values were manually-trained to achieve optimal performances on the training data sets. Then the LMC Sept. 94 and Carson test data were evaluated using these parameter values.

Using the same training data sets, the GA and CC tools were applied to produce three Bayes nets: one for Carson, one for LMC Sept94 and one for NVL-amber data. Again these Bayes net couple with the adaptation paradigm were tested on the corresponding test data sets.

The manually-trained test results and the ACC ATR results for the Carson data set are listed in Tables 2 and 3 respectively. Similarly the two results on LMC Sept. 94 test data are listed in Tables 4 and 5. The ACC ATR results on the NVL-amber data is shown in Table 6.

A comparison between Tables 2 and 3 indicated that the adaptive approach achieved similar probability of target detection. However, the number of false alarm is significantly less in the adaptive approach than that of the manually trained approach. Similar conclusion is observed in the LMC data set by comparing Tables 4 and 5.

Table3: Test Results on Carson Data by Manually-trained method

		Groundtruth	
		Target	Non-target
Test	Target	60	103
	Non-target	41	----

Table 4: Test Results on Carson Data by Adaptive Approach

		Groundtruth	
		Target	Non-target
Test	Target	54	39
	Non-target	47	----

Table 5: Test Results on LMC Sept. 94 Data by Manually-trained Approach

		Groundtruth	
		Target	Non-target
Test	Target	442	690
	Non-target	157	----

Table 6: Test Results on LMC Sept. 94 Data by Adaptive Approach

		Groundtruth	
		Target	Non-target
Test	Target	399	218
	Non-target	200	----

Table 7: Test Results on NVL-amber Data by Adaptive Approach

		Groundtruth	
		Target	Non-target
Test	Target	192	159
	Non-target	154	----

Figure 12 shows three images of diverse scenarios and their detection results generated by the manually trained approach and the adaptive approach. The rectangular box is where the true

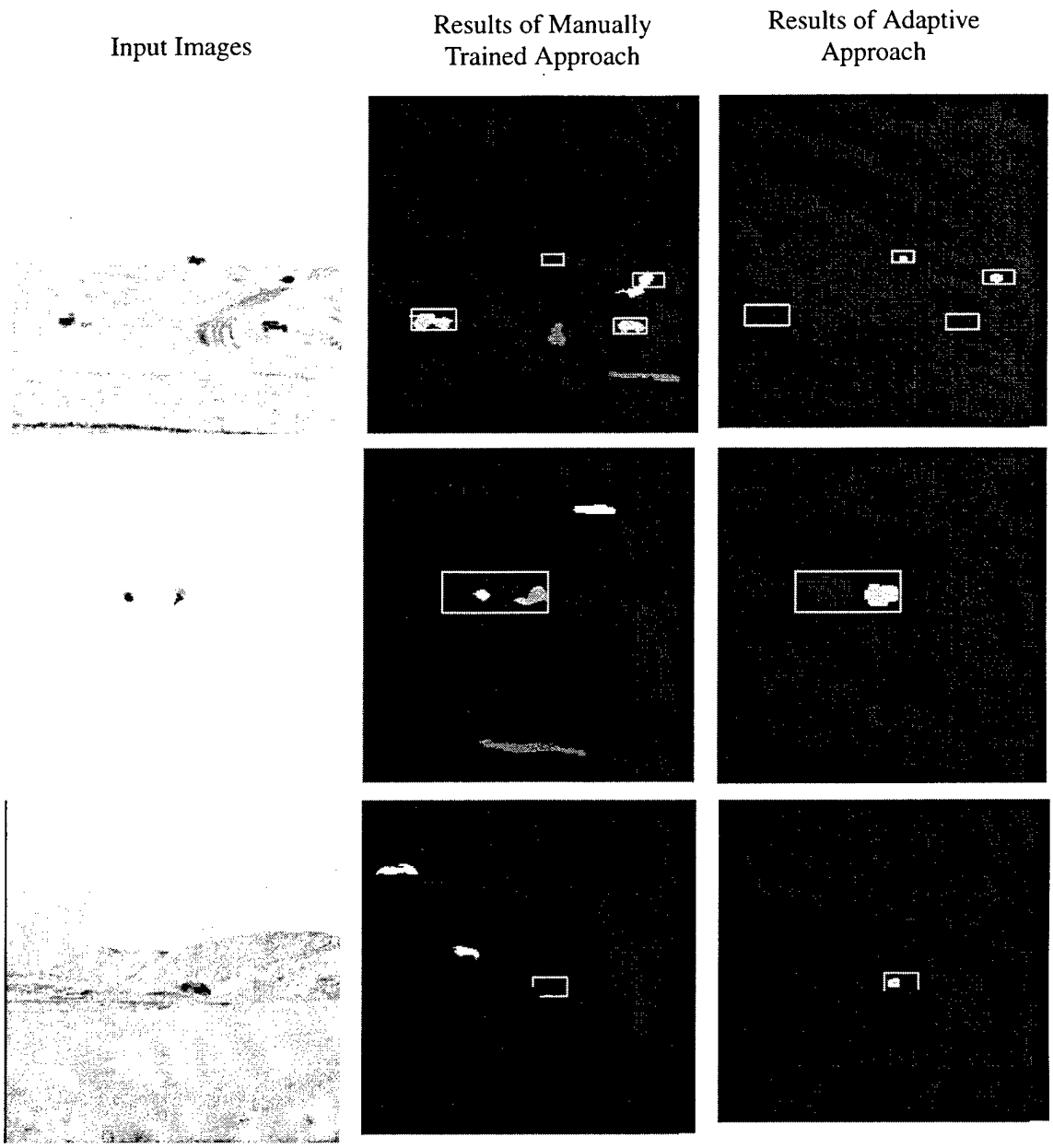


Figure 10: Results of the Manually-trained approach and the Adaptive Approach on Three Diverse Scenarios (clutter level, range, and time of day).

target location are. The other blobs are the detected objects. It is evident that the adaptive approach produced less false alarms.

7 Future Directions

The GA system allows us to look across the trained scenarios to identify the weak links in processing chains (e.g., modules which require different algorithms in each scenario) and gain insights into performance of algorithm suite components. Hence, the GA system has merit as a stand-alone performance evaluation tool, independent of the whole self-adaptive ATR system. Honeywell is developing a Universal Performance Evaluation Tool (UPET) based on the GA approach under a Wright Lab funded program.

Note that an alternative to SP would involve the use of Case-Based Reasoning as described in [Roberts, Au, 95]. It should be noted that the authors have yet to develop any software to implement this alternative.

Ultimately it is the choice of "context" and the means by which the context -to- configuration pairs are recorded and indexed that are cornerstones of the system. The authors are currently studying these issues in detail and making system modifications to improve performance.

8 Summary

A self-adaptive ATR system using context-based configuration and control has been developed. During the training phase, the system employs a Genetic Algorithm to autonomously configure algorithm suites and set the parameter values. Then, a Context Capture tool determines the key context for the mission scenarios, and stores the configuration-context information into a Bayes Net. During field operation, the adaptive system employs a Selective Perception tool to dynamically configure and control the ATR processor based on the inference of the changing mission context in the Bayes Net.

Our approach works for any ATR system regardless of its sensor modality; thus our approach is applicable to SAR-, FLIR-, and LADAR-based ATRs. The context of these different ATR types would be different due to the different sensing phenomenology. However, the Context Capture tool is capable of acquiring these different context.

The Genetic Algorithm, the Context Capture and the Selective Perception modules have been completed. These three components are integrated into an adaptive ATR system, which was demonstrated during the Demo C of the UGV program. The system successfully detected the two targets presented without any false alarms. In-depth self-evaluation on large data sets indicated that the adaptive ATR system produced better training results and performed better than manually-trained baseline algorithms during testing.

The triad of tools being assembled under the Honeywell effort, form an adaptive context/ mission-dependent ATR *system* that holds the promise of robust ATR performance that will be acceptable to the user community. The paradigm used in our effort will change the way image understanding is performed, raising IU robustness and user acceptability to a higher plateau.

The system being developed is novel and innovative in addressing what are major problems/ weaknesses of ATR systems. What is needed to solve the ATR problem is not more algorithm development per se, rather the intelligent selection and application of algorithms to best meet the scenario at hand. Hence, it is our feeling that the ACC system will be a key component in the UGV RSTA system; enabling robust RSTA operation under the variety of scenarios to be encountered during an extended UGV mission. The key measure of success of our ATR *system* is the quantifiable improvement in ATR performance that it produces within the variety of RSTA scenarios.

By-products of the combined effort will include disseminable code for Bayes Net and Decision Theory algorithms, and the GA and CC software tools to facilitate the development, characterization, and training of a broad category of algorithms (i.e., not just ATR algorithms). Both of the code deliverables will be of general use to the RSTA community (as will the whole self-adaptive ATR system).

Note that the transfer of technology is not a one-way street. The development of other teams could be very useful to the effort described herein. Certainly, it is desirable to have access to the algorithms being used by other RSTA contractors; the algorithms can be included in the algorithm library being used by the GA system.

Acknowledgments

The work that is described herein has benefited greatly from the efforts of Prof. Chris Brown and Mike VanWie of the University of Rochester. Their contributions have made the Selective Perception and Context Capture components of the adaptive ATR system a reality.

The authors also would like to thank Erik Mettala, Oscar Firschein and Tom Strat for their support and encouragement throughout the RSTA effort.

References

[Au, Roberts, 95] Wing Au, Barry Roberts, "Adaptive Configuration and Control in an ATR System", Proc. IU Workshop, Feb.,

1995 Palm Spring, CA. Morgan Kaufman Publisher.

[Booker et al, 89] L.Booker, D.Goldberg, and J. Holland, "Classifier systems and genetic algorithms," Artificial intelligence, vol. 40, pp235-282, 1989.

[Holland, 75] J. Holland, "Adaptation in Natural and Artifical Systems," Univ. of Michigan Press, Ann Arbor, MI, 1975.

[Rimey, Brown, 91] Rimey, R.D., Brown, C.M., "Task-Oriented Vision with Multiple Bayes Nets," Technical Report 398, Department of Computer Science, University of Rochester, November 1991.

[Roberts, Brown, 94] Barry Roberts, Chris Brown "Adaptive Configuration and Control in an ATR System", Proc. IU Workshop, Nov., 1994 Monterey CA. Morgan Kaufman Publisher.

[Roberts, Au, 95] Barry Roberts, Wing Au, "Adaptive Configuration and Control in an ATR system", Proc. SPIE, Vol 2496, Detection Technologies for Mines and Minelike Targets, April, 95. Orlando, FL.