

AD-A237 544



Development of Laser Radar ATR Algorithms:
Phase II—Military Objects

PHASE II TECHNICAL REPORT

Elie Bienenstock
Donald Geman
Stuart Geman
Donald E. McClure

31 October 1990

Contracting Unit:
Harry Diamond Laboratories, Adelphi, Maryland
With Support And Cooperation From
U.S. Army Center For Night Vision and Electro-Optics
Fort Belvoir, Virginia

Contract No. DAAL02-89-C-0081

Contractor:
Mathematical Technologies Inc.
P.O. Box 2476 (One Richmond Square)
Providence, Rhode Island 02906-0476

VIEWS, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS 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 OFFICIAL DOCUMENTATION

91

10

640

91-02451



1

Rec memo
A-1

Laser Radar ATR Algorithms
PHASE II FINAL REPORT: 9/1/89 THROUGH 9/30/90

Contract Reference Number: DAAL02-89-C-0081

Company Name: **Mathematical Technologies Inc.**

Period of Performance: 9/1/89 through 9/30/90

| | |
|--|----|
| 1. Introduction | 3 |
| 2. The MTI Recognition System | 5 |
| 2.1 Problem Statement and Heuristics | 5 |
| 2.2 Overview of the Algorithm | 9 |
| 2.2.1 Search Protocol | 9 |
| 2.2.2 Decision Tree | 11 |
| 2.2.3 Virtual Templates | 13 |
| 2.2.4 Occlusions | 14 |
| 3. Simulated LADAR Scenes | 15 |
| 3.1 Target Templates | 15 |
| 3.2 Scene Generation | 16 |
| 3.3 Sensor Model | 17 |
| 3.3.1 Unresolved Targets and Blur | 18 |
| 3.3.2 Noise Model | 24 |
| 3.3.3 Dropouts | 26 |
| 3.3.4 Illustrations | 26 |
| 4. Training: Probe Optimization | 27 |
| 4.1 Alignment | 27 |
| 4.2 Occlusion and Half-Targets | 28 |

| | |
|--|----|
| 4.3 Decision Tree | 30 |
| 4.3.1 Detection: Rounds 0 and 1 | 30 |
| 4.3.2 Classification: Rounds 2 and 3 | 32 |
| 4.3.3 Verification: Rounds 4 and 5 | 34 |
| 5. Experiments | 36 |
| Figures | 39 |
| References | 65 |

1 Introduction

Under this contract, MTI has developed a software system for the automatic recognition of military targets occurring in laser radar range data. The algorithm is designed to operate over a variety of scenarios: the targets may occur in multiple numbers and aspects, and may be partially occluded due to other vehicles, irregularities in the terrain, or background clutter. The particular algorithm delivered to CNVEO under this contract is configured to detect and classify three particular targets (M35, M60, M113), but the system is based on very general principles concerning invariant rigid body object recognition and easily extended to additional targets, clutter types, sensor effects, and degrees of freedom in the viewing angles.

In fact, this work represents an enhancement and special purpose implementation of a generic object recognition system developed at MTI. Initially, this system was developed for the classification of two-dimensional shapes in highly degraded, visible light intensity data; in particular, optical character recognition served as a convenient and challenging prototype problem, and MTI has now developed a commercially viable software product capable of accurate identification of printed characters in noisy and cluttered images, such as high magnification photographs of alphanumeric identifications on silicon wafers.

Work was divided into two phases. The initial phase, a five month effort, called for the delivery of an algorithm for the recognition of at least two "geometric shapes" in simulated laser radar range imagery, with a particular emphasis on realistic clutter models. Work in Phase I was devoted to

the construction of a surrogate database and corresponding recognition algorithm designed to accommodate the particular difficulties, especially clutter, occlusion, and invariance, inherent in rigid body recognition from range data. This database consisted of simulated range data for "scenes" composed of three-dimensional polygonal objects with features of military-like targets (e.g., planar facets). Each object was randomly composed from a number of randomly generated, rectangular parallelepipeds. A full description of these efforts, together with details of the search algorithm, were provided in the Phase I Technical Report [2].

It was also anticipated that during Phase I the Government Labs would provide MTI with "primitive models" for several military targets, although work on actual military targets did not commence until Phase II. The remainder of this document will focus on the work performed under Phase II. Due to the lack of *actual* LADAR imagery, it was decided to design and test the algorithm on imagery generated with the CNVEO LADAR Simulator, which uses CAD-based target models, and was developed in conjunction with Honeywell Systems and Research Center.

However, during Phase II of the contract, certain problems were revealed concerning the manner in which this Simulator incorporates sensor effects; see §3.3. Consequently, MTI has separately implemented a LADAR simulator and the experiments described in this contract were performed on imagery generated by the MTI LADAR Simulator. In order to simulate LADAR range imagery, we have constructed "scenes" and corresponding range images by randomly positioning targets and "semi-targets" over a simulated landscape and computing the appropriate depth values from a reference point; see Fig-

ure 1A for a blow-up of one such simulator example with range-to-target of approximately 3km and Figure 1B, for comparison, depicting real data of an M60 at range-to-target of 1.04km. The ray-traced simulated range images are designed to accommodate multiple targets at multiple ranges and aspects, systematic changes in background complexity and figure-to-ground separation, and the effects of obscuration, sensor blur, and sensor noise.

2 The MTI Recognition System

2.1 Problem Statement and Heuristics

Consider the general rigid body object recognition problem. We are given a list of 2D or 3D "objects"; for example, the letters of the alphabet, a collection of military vehicles, or an assortment of machine parts or manual tools. The objects are regarded as rigid and represent *particular* instances of the given shape class; thus, for example, the letters are represented by a particular font, the vehicles are specific tanks and trucks, and so forth. The objects are then arbitrarily positioned in 3-space (or in 2-space if they are two-dimensional), with respect to rotations and translations, and this "scene" is then imaged by an ordinary camera or perhaps by a range-finding device. The scene may contain multiple objects, each in multiple aspects; some objects may be partially occluded by others or by "clutter." In addition, there may be noise or other degrading effects caused by the way in which the scene was illuminated and sensed. The goal is then to construct a list of those objects present in the scene, together with the locations at which objects occur,

based on the image data and (exact) shape information about the objects, e.g., CAD-based specifications. This is the problem of *rigid body invariant object recognition*.

We should emphasize the distinction between *rigid* and *non-rigid* (or *deformable*) objects, for which, in addition to the multitude of representations induced by spatial positioning, there are the additional ambiguities associated with the varying intrinsic shapes of individual objects. In our problem the individual patterns exhibit no variability in shape and hence there is no need for a model for variation *within* shape classes. Still, the rigid body problem is quite challenging because we wish to solve it in *highly degraded environments*, including substantial degrees of sensor degradation, target-like clutter, and target obscuration.

As mentioned above, there are two particular cases MTI has examined in detail. They are representative of the problems encountered and of our practical experience in this area. The first is optical character recognition, in which the objects are the thirty-six alphanumeric characters and the images are ordinary visible light pictures obtained with a video camera. The second case is the one at hand: automatic target recognition in LADAR range data. However different these applications may be, the *general problems* are more or less the same—those indigenous to invariant rigid body recognition.

The most conceptually simple and straightforward approach would be to store a library of exact representations of all targets in all potential aspects and then search *individually* for these object-aspect combinations by some form of template matching or other global measure of fit. For example, given a reliable detection algorithm, the positions of potential targets could

be located and, in principle, template matching could be done with optical correlators. Whereas even for optical correlators, the enormous number of potential signatures might impose unacceptable limitations on speed, there is still another, more crucial, limitation: correlation severely degrades in the presence of noise and clutter. This is particularly true when distinct objects in varying aspects may have nearly identical signatures. The principal reason for this is that, by its nature, pure correlation *uniformly* emphasizes all regions of the objects (and similarly for other global measures of fit); in particular, there is no mechanism for *focusing* on ambiguous areas—those where confusions are likely to occur—and such ambiguities are in fact the essence of the problem. In the case of optical character recognition, we refer to this as the “E/F” dilemma: two presentations of an “E” may be farther apart in the metric induced by the measure of fit than an “E” and an “F”. Consequently, the very *representation* of an “E” must be influenced by the existence of an “F” among the list of hypotheses; see §2.2.2 and §4.3. The situation is identical for military vehicles in range data: for instance, a “noisy” tank may correlate better with a truck than with an ideal tank, and similarly an obscured tank may correlate better with a truck than with an unobscured tank. For these reasons, we believe that no single measure of fit is adequate, and, in particular, we have avoided correlation and other such measures from all steps of the algorithm—detection, classification, and verification.

Any search procedure should then proceed on a *coarse-to-fine* basis, in which many possibilities or “hypotheses” are considered at the early stages, giving way in a controlled progression to increasingly narrow and more spe-

cific investigations. Moreover, during this progression, the type of tests employed must have the "focusing property" lacking in global measures of fit, which is why we have employed *local, nonparametric measures of fit*; see §4.3. The separation of targets with nearly identical presentations should be delayed until the last stages of the search procedure. One natural protocol is then a coarse-to-fine *decision tree*. See Figure 2 for an example; the symbols "A,...,Z" may represent our three military vehicles in varying aspects.

Two other features which separate our algorithm from others actually in use are: (1) *Floating thresholds*, a particular form of "top-down" or "hypothesis-driven" processing; and (2) An off-line training procedure in which, at each node of the decision tree, the hypothesis representations themselves are "learned" by optimizing the selection of "tests" or "probes".

Whereas it has long been recognized in the computer vision community that object recognition cannot be purely data-driven, i.e., that decisions should somehow be interpretation guided, this has rarely been implemented in a practical or coherent fashion. In particular, we believe that any procedure based on blind segmentation, meaning fixed and universal thresholds, will fail in the presence of correlated noise, vagaries in illumination, and other factors common to real imagery. Still, local property values must be extracted from the data and represented in a form sufficiently simple for comparison to stored representations, and it is precisely at this stage of data reduction that we have found it effective to allow pending interpretations to determine the search parameters. In particular, we use "floating thresholds"; see §2.2.3 and §4.3.2.

Turning to the object representations, consider first conventional statis-

tical classifiers: a collection of "features," which are simply functions of the image data, are pre-specified and one attempts to estimate the conditional distribution of these features given the various hypotheses, the so-called class-conditional densities. The latter step is referred to as "training." Instead, in our method, the appropriate features are *learned* and our "training" consists of an intensive, but off-line, optimization procedure in which the object representations are constructed in terms of elementary features we call "probes." The cost criteria for these representations are economy, robustness, and discriminating power. The manner in which these representations or *virtual templates* are chosen is discussed in §4.3.

2.2 Overview of the Algorithm

2.2.1 Search Protocol

We will use the word *hypothesis* to indicate a particular object-aspect combination; thus, for example, there is one hypothesis for each target type for each triple of angles corresponding to an appropriate sampling of azimuth, tilt, and rotations in the ground plane. We may assume the scale is fixed since the camera-object distance is known. The number of degrees of freedom allowed is, of course, situation dependent. Formally, at least, the only difference is in the actual number of hypotheses. For the LADAR recognition problem, the most important degrees of freedom appear to be rotations in the ground plane, and we have focused to date on that case, assuming the tilt and azimuth are effectively zero; the methodology extends simply to situations in which irregularities in terrain or viewing angles are prominent.

The recognition strategy is based on sequentially visiting each (or most) image locations and implementing a decision tree for a field of view associated with that pixel. The output at each branch of the decision tree is a list indicating which hypotheses are "active" at the pixel, that is, have not been eliminated at any earlier junction of the decision tree. An hypothesis is "true" within a field of view if the object is positioned there in such a way that a distinguished point in a subimage containing the ideal object-aspect signature is aligned with the origin of the field of view.

Let us now imagine that a field of view is fixed; the precise registration mechanism will be explained in §4.1. The algorithm is based on a series of *probes* which are grouped into "Rounds" corresponding to the nodes on the decision tree. These probes refer to particular functions of the image data which are evaluated at predetermined locations, one type of function and one collection of locations for each node in the decision tree. These locations or "offsets" are determined by the aforementioned optimization procedure; see §4.3. Roughly speaking, the probes and offsets are optimized to minimize the error rates corresponding to false negatives (unidentified targets), false positives (non-targets mistaken for targets), and erroneous classifications (actual but mislabeled targets). These may be regarded as tests upon which detection and recognition are based: the observed data values determine the action taken at each branch of the decision tree. Hypotheses which are active at a given node and which "pass" a sufficient number of the tests for that node will remain active at the given location. The final output indicates which, if any, of the basic target types has been confirmed at the pending location; obviously most locations result in no confirmations.

The basic strategy is then a variant of "divide and conquer": many alternatives are pursued in parallel in the early stages, based on very general and mutually relevant criteria, whereas the intermediate stages focus on subclasses of hypotheses and finally, in the latter stages, the tests are designed to confirm or deny specific hypotheses against all the relevant alternatives, for example a particular orientation of a tank against all pending aspects of other target types.

2.2.2 Decision Tree

As mentioned earlier, the probes are grouped into five rounds corresponding to nodes on the decision tree. The purpose of Round 0 and Round 1 is the rapid detection of a possible target at the given location. Consequently, these rounds serve as filters to separate targets from background and to quickly eliminate most locations from further examination. Moreover, since in principle we allow no false negatives (unconfirmed targets), these filters must reliably identify all locations associated with actual targets. The probes in these rounds are elementary and generic, and no attempt is made to discriminate among targets. The result is that most locations which survive these rounds are in fact false positives and do not correspond to a distinguished location on an actual target but result instead from target-like clutter or other targets at nearby locations. See §4.3.1 for a detailed description of these early round probes.

In contrast, Rounds 2 and 3 (see §4.3.2) represent the core of the algorithm and are designed to separate targets from clutter and from each other.

Thus these rounds are more computationally intensive (although executed at only sparse locations, i.e., those which survive earlier rounds), involve more complex probes, and are geared towards resolving ambiguities. Differences among all presentations of distinct targets must be precisely identified and exploited in a manner which is robust to noise, clutter, and parameter selection. It is here that we utilize "top-down processing" by employing only hypothesis-driven thresholds. This is the true *recognition* aspect of the problem and is the area to which we have devoted most of our efforts and which we regard as the essence of the problem itself. The result of Round 3, as all earlier ones, is a list of hypotheses which remain active at the given location.

The purpose of Round 4 is to utilize the *internal structure* of the targets, i.e., the relative depth values of the pixels on target, to screen active hypotheses, i.e., pending object-aspect pairings. This is done by checking that the observed values are "consistent" with the stored ones. This stage involves a simple, one-parameter regression model; see §4.3.3. Finally, in Round 5, we disambiguate among confirmations which lie in close proximity. Final decisions are based on a "survival-of-the-fittest" protocol in which pending confirmations are tested against each other to determine which, if any, are declared as labeled target locations. These decisions are based on analyzing the *residuals* which arise in the statistical data fitting from Round 4; again, the exact mechanism is described below in §4.3.3. The procedure is only performed at very sparse locations and for candidate targets which have already "passed" all previous tests; consequently, the overall cost (say in computation time) is no greater than that of the previous rounds.

2.2.3 Virtual Templates

The stored probe values for each round are in effect “virtual templates” (or “sampled templates”). As the number of probes increases, these virtual templates converge to the literal templates (the perfect range information) with respect to the particular type of information upon which the probes are based. For example, suppose the literal templates are binary images indicating whether pixels are on or off the target, a variant of the “figure-ground” dichotomy in which the figure is determined by being closer to the viewer. The virtual template is a binary sequence corresponding to the template values at a distinguished subset of locations; it becomes the literal template as the number of locations approaches the number of pixels in the subimage defining the literal template. The idea is to use as few points as possible and still reliably accomplish the task associated with the given branch of the decision tree, for example to separate objects from background or to separate a particular target from all occurrences of other target types.

Still more powerful representations may be obtained with *relational primitives*. For example, we might associate with each *pair* of locations, usually in close proximity, a binary label corresponding to whether or not the pair of points straddles the object boundary, i.e., is a (figure, ground) *pair*. The figure/ground dichotomy is replaced by that of transition/no transition. In a real image containing that object, the transition pairs should typically correspond to significant differences in depth values whereas others should correspond to relatively small differences (depending on the nature of the background). Each hypothesis is again represented by a binary string and

the object silhouette is recovered in the limit as the number of pairs increases.

It is important to notice that relational template matching necessitates that the actual intensity values that are extracted at the predetermined locations associated with a probe must be converted to a label, usually just 0 or 1, for comparison with the stored models. This is the case in Rounds 2 and 3. We may think of a probe as the set of points together with a label that depends on the particular template in the field of view.

A critical factor in the success of this approach is that threshold values used in the conversion of intensity values to labels be driven by pending interpretations. The alternative, using global thresholds, renders the algorithm unduly sensitive to parameter selection, illumination changes, and other factors, and results in unacceptable error rates. One method of incorporating this top-down component uses a "floating threshold" in the formal statistical test of the particular hypothesis being entertained. Detailed descriptions of the proprietary approach are given in the technical description of the algorithm, delivered with the software. The rationale for our approach is to minimize the probability of detection error when, in fact, the entertained hypothesis is true. Such a form of hypothesis-driven segmentation is used in the generic MTI recognition algorithm and in the current ATR algorithm.

2.2.4 Occlusions

The obscuration problem is important because portions of actual targets may be hidden by various entities such as other targets and background objects. We have accounted for one particular type of obscuration in which

the lower portion of the target is hidden due to terrain anomalies. However the methodology easily extends to other forms of obscuration, such as lateral occlusion. Basically, this is accomplished by extending the methodology to new sets of hypotheses corresponding to “semi-targets”—full targets at the same aspects, but with some portion removed. In other words, we train the algorithm with semi-targets and search for them in the same way as for fully visible targets. The classification problem is of course more difficult because the partial targets are not as well separated; in addition, the “clutter model” must be appropriately modified.

3 Simulated LADAR Scenes

3.1 Target Templates

There are three targets, the M35 Truck, M60 Tank, and M113 APC, and there are 108 training images, corresponding to each of the three targets occurring in thirty-six aspects. These are ray-traced images, the largest approximately 60 x 130 pixels, corresponding to a range of about 3 km, an angular sampling interval of about 0.05 milliradians and sensor depth resolution of 4 cm, and are obtained from a CAD-CAM database. If we assume that the viewer is situated along the x-axis in a standard coordinate system, and that the x-y plane represents the ground plane, then the 36 aspects mentioned above correspond to each ten degree rotation of an object around the z axis. In effect, then, the aspect angle is zero and the tilt may be regarded as fixed by the initial positioning. Obviously additional degrees of freedom could be

introduced by considering additional rotations about appropriate axes.

One problem with this approach is that some rotations induce less variation than others; this is readily seen in Figures 3, 4, and 5 which show, respectively, the truck, tank, and APC in each of 18 aspects corresponding to rotations through 180 degrees. A better procedure would be to divide the range of angles according to some measure of similarity so that, roughly, each subinterval of angles would generate comparable variation between templates.

3.2 Scene Generation.

The first step was to simulate a high-resolution ray-traced image of a landscape, including a horizon line, by computing range data for a flat surface at a (small) aspect angle and locally and randomly perturbing this surface to account for terrain irregularities.

Next, several of the 108 high resolution, appropriately scaled target templates are randomly positioned in this landscape, some on the horizon and others in the background. Similarly, a model for clutter was generated by randomly selecting and randomly positioning target "remnants", i.e., pieces of targets, in the landscape. At this stage, we have an "ideal" simulated LADAR scene because we have not yet incorporated the effects of the sensor.

3.3 Sensor Model.

The CNVEO LADAR Simulator developed by Honeywell Systems and Research Center was designed to generate LADAR scenes including so-called resolved targets. As it is currently implemented, it cannot model the effects of the sensor for unresolved targets. The assumption that a target is resolved does not hold at all for objects of interest at ranges of three kilometers and it breaks down partially even for targets at much closer ranges.

When it came to our attention late into the contract period that the simulator had this limitation, we began a careful evaluation of the Honeywell Simulator Program, together with an analysis of the scientific and engineering literature concerned with range measurement by a heterodyne CW laser radar. The work on the simulator included discussions with scientific/engineering staff at Honeywell, and it was greatly facilitated by Richard Peters and Teresa Kipp at CNVEO. Many of the references on which this work was based were suggested or provided by Teresa Kipp.

The next step in the scene generation is to use the high-resolution ray-traced image described in §3.1 as input to a sensor model, which embodies the effects of the finite spot size of the LADAR, reflectance properties and range of different points of the scene, and attenuation of the signal energy by the atmosphere. The output of this step is a lower-resolution, blurred and sampled version of the input image. Finally, the output of the sensor model is given as input to a noise model, which incorporates the range dependence of the uncertainty of the range measurements. The noisy image can then be quantized to obtain both absolute and relative range data.

3.3.1 Unresolved Targets and Blur

First we shall address the question of an appropriate sensor model for unresolved targets.

In AM heterodyne range detection, the *phase difference* between an amplitude modulated transmitted signal and a reflected received signal is used to determine the distance between the transmitter/receiver and the point(s) in the field-of-view (FOV) on which the transmitted beam is incident. The beam actually has positive finite extent, so the reflected signal is a superposition of reflected coherent optical fields integrated over the area in the FOV on which the spot is incident.

We shall adopt certain simplifying and justifiable assumptions about surface reflectivity, spot size, transmittivity, quantum efficiency of photoreceptors, and so on, consistent with the assumptions made in the implementation of the Honeywell simulator. Specifically, we shall assume (i) that surfaces are "rough," i.e., that local surface irregularities are of comparable scale to the optical wavelength $c/(2\pi\nu_o)$ of the laser, (ii) that consequently reflectance is Lambertian, rather than specular, (iii) that the diameter of the transmitted beam is on the order of two milliradians (mrad), and (iv) that the angular resolution of the generated images will be 0.05mrad horizontally and vertically, or possibly coarser.

The physics of the description of the received energy from the reflected beam is quite well understood. In the physical modeling, it is important to distinguish between the so-called *resolved* and *unresolved* cases. The simpler case is when the target (object or background) is resolved. This means that

the entire spot is reflected. The simplest instance of a resolved target occurs when all points covered by the spot are at the same range, rather than being distributed over a wide set of range values. In contrast, a target is said to be *unresolved* if only part of the spot is incident on the target. This case occurs if the spot is larger than the target or, more frequently, if a beam intersects the edge of a target, with part of the spot on target and part of the spot off target (and on sky or infinite space). The unresolved target case is the right model to consider for points in the FOV that are on the horizon, including especially parts of objects that are above the horizon.

If the target is resolved and the spot is *entirely* incident on an area at distance R from the transmitter/receiver, then the received power P_r will be given by an expression of the form

$$P_r = P_t(\rho/(4\pi R^2))AT_a^2T_o^2 \quad (1)$$

where P_t is the transmitted power, ρ is the target reflectivity, R is the range to target, A is the effective area of the receiver, T_a is the one-way atmospheric attenuation, and T_o is the one-way optical system loss [1]. It is useful to highlight dependence of this expression on R . If atmospheric attenuation is uniform, described by the constant atmospheric attenuation coefficient α , then T_a will have the form $T_a = \exp(-\alpha R)$. The expression in eqn. (1) then has the general form

$$P_r = c_1(\exp(-2\alpha R)/R^2)P_t \quad (2)$$

The factor c_1 lumps together the other factors in (1).

The article by Goodman [5] contains a careful description of the basic physical principles on which one can build an understanding of unresolved

targets and of cases where the points covered by the spot are at widely varying ranges. Each polarization component of the received optical field at any point (u, v) in a plane just in front of the receiver aperture may be regarded of a sum of random-amplitude, random-phase, complex phasors contributed by the elementary scatterers. (As noted above, the reflected signal is a superposition of reflected coherent optical fields integrated over the area in the FOV on which the spot is incident.) From this fact, and using the Central Limit Theorem, one can theoretically justify the conclusion that the polarization components of the received optical field are complex Gaussian processes over space, and hence that the associated energy densities have the Rayleigh (viz. negative exponential) distribution. Several studies have supported the empirical validity of this conclusion as well [7,9].

Ideally, to develop a simulator that accurately models the physics, one would (i) compute how the *amplitude* of the received modulated signal depends on the transmitted waveform, beam shape, range, target reflectivity, and atmospheric and optical losses, and then (ii) model and compute how the range detector would determine a measured range from the combination of the received and transmitted AM optical fields. While the first step is reasonably straightforward, the second part is complex. It is agreed that this is a desirable, but not a feasible approach [4]. A reasonable alternative to the ideal simulator is to model the principal qualitative properties of the sensor—blur and other optical effects, and range dependence of detection sensitivity.

One important sensor property is a consequence of finite spot size: sensors blur. As noted above, the received optical field will be a superposition, a

convolution average, of fields reflected from individual points in the spot. The principal of superposition applies to optical field strength.

The principal of superposition does not apply to measured range values per se.

The basic flaw of the Honeywell simulator is that it builds a sensor model *completely* around a linear convolution of ray-traced range values. That is, the sensor-model output at pixel (r, c) is given by an expression of the form

$$R_s(r, c) = \sum_{u=-\Delta}^{\Delta} \sum_{v=-\Delta}^{\Delta} K(u, v) R_{rt}(r - u, c - v) \quad (3)$$

where R_s denotes the sensor-output range measurement, R_{rt} denotes the ray-traced range data and $K(,)$ is a convolution kernel. K is taken to be a Gaussian bump with its support restricted to a square region of angular extent ± 0.1 mrad. K is also normalized so that

$$\sum_{u=-\Delta}^{\Delta} \sum_{v=-\Delta}^{\Delta} K(u, v) = 1. \quad (4)$$

This model cannot be justified by the physics of the system and measurement process. Indeed, the physical principles embodied in eqn. (2) are incompatible with eqn. (3). Manifestly, the reflected energy from a point in the scene decreases in strength as range increases, approximately according to the law $\exp(-2\alpha R)/R^2$. However, eqn. (3) gives *greater* influence to points at greater distance from the transmitter/receiver. The dependence on range in (3) is the reverse of the dependence on range in (2). Still, one can argue in favor of the model (3) on the basis of the tenuous heuristic connection that convolutions *blur*, and we need a simple, flexible, and suitably general shortcut to modeling sensor *blur*.

The use of a linear convolution filter may be suitable for fully resolved targets, but it breaks down completely for unresolved targets. (Notably, the Honeywell sensor does not purport to model sensor effects for unresolved targets.) If a point of the spot is not incident on target, and therefore does not reflect any energy, it is *not* possible to identify any consistent "range" value to use for this point in the convolution (3). Indeed, some striking physical inconsistencies can occur if one still applies the convolution filter literally. If one were to associate the value ∞ with the point not on target, then the value of R_s , according to (3) at that point would be ∞ , regardless of how much of the spot is on target.

Note: The Honeywell program does this, in effect. The value 10^{30} is assigned to "sky" in the sensor model. Then if the filtered value (3) exceeds 10^{20} , it is reset to "sky"— 10^{30} . This has the effect of obliterating every unresolved point in the ray-traced scene. Figure 6 shows the effect of subjecting the M35 templates to the incorrect resolved-target sensor model.

To mitigate this inconsistency with the physics, we have adopted the following strategy for treating unresolved points in the scene. The strategy borrows on the heuristic identified above of using a convolution filter for the resolved portion of the spot. However, we explicitly assign no weight to points with no reflectance. As a surrogate for "reflected energy" at pixel (r, c) , we define the weight $W(r, c)$ by

$$W(r, c) = \sum_{u=-\Delta}^{\Delta} \sum_{v=-\Delta}^{\Delta} K(u, v) \cdot I_{\{R_r(r-u, c-v) < \infty\}}. \quad (5)$$

The factor defined by the indicator function $I_{\{ \}}$ will assume the value one at (u, v) if and only if the point $(r - u, c - v)$ in the ray-traced image is a

point of finite range, i.e., not sky and not infinite space; otherwise this factor is zero. Then set

$$\hat{R}_s(r, c) = \sum_{u=-\Delta}^{\Delta} \sum_{v=-\Delta}^{\Delta} K(u, v) R_{rt}(r - u, c - v) \cdot I_{\{R_{rt}(r-u, c-v) < \infty\}} \quad (6)$$

and compute the (average) sensed range at (r, c) as

$$R_s(r, c) = \hat{R}_s(r, c) / W(r, c). \quad (7)$$

Now for our heuristics: We can interpret eqn. (5) as describing the “strength” of the reflected waveform while eqn. (6) represents the *uncorrected* average range of the points in the spot that are on target. Eqn. (7) corrects for the fact that only a fraction of the spot is on target.

We recommend one additional feature for the unresolved target sensor model. The probability of detecting a target is directly dependent on the energy of the reflected optical field [1,5,8]. In particular, when the return has very weak amplitude, then the probability on no detection is greater. To incorporate this effect, our sensor model will deterministically or randomly, as the user chooses, allow a so-called “drop-out” if the amplitude of the reflected signal is weak. In the deterministic model, a drop-out occurs, and the corresponding pixel is identified as “not on target,” if $W(r, c) < \gamma$, where the threshold γ is a simulation algorithm parameter. In the randomized model, the probability of a drop-out is $1 - W(r, c)/\gamma$, provided $W(r, c) < \gamma$.

Implementation Notes. 1. It is not important that the input to the sensor model be a *high-resolution* ray-traced image. One can expect to achieve reliable simulations if the ray-traced input image is at the same resolution as the output image. Technically, the comparison can be thought

of as sampling after averaging vs. sampling before averaging, and how the choice affects the accuracy of the quadrature formulas (5) and (6). If the output of the sensor model will subsequently be the input to a realistic noise model, then the quadrature error incurred by using a low-resolution ray-traced input image will be overwhelmed by the random measurement errors incorporated in the noise model.

2. The nonlinear sensor model of equations (5)-(7) will not be nearly as amenable to efficient algorithmic design and implementation as the simple linear sensor model of equation (3). Presumably, execution speed is not an issue for the simulation algorithm. Speed is certainly less important than fidelity with the physics.

3.3.2 Noise Model.

We have employed a noise model that is dictated by empirical results from analysis of actual laser radar measurements. Relevant data are available from experiments done at Fort A.P. Hill in 1989 with the Raytheon Tri-Service Laser Radar. Range measurements were made of a flat wall at ranges of 1500m, 2040m and 3200m. From the actual range measurements, robust estimates of the standard deviation of the range values were formed. The results have been made available by CNVEO.

The Honeywell simulator also uses these data for calibration of a more intricate range-error model. However, we believe that it is probably a mistake to use the more intricate model (see [10]) in the simulator. The analysis in

[10] assumes a specific modulation waveform

$$F_{mod}^{AM}(t) = \sin[(\phi/2) \sin \omega_m t + (\pi/4)] \quad (8)$$

appropriate for a specific laser radar design. It does not claim to be generic. Further, the noise model in [10] is greatly oversimplified in the assumptions that it makes about the temporal variation of the noise. The model assumes that the noise is random, with an arguable marginal distribution related to the well-founded Rayleigh statistics, but it assumes that the spectrum of the noise process is degenerate. The extreme nature of this assumption renders the application of this model to be questionable at best. Finally, in the approach used in the Honeywell simulator, values for the range-measurement standard deviation are extrapolated beyond the maximum range for which empirical evidence is available. This is always a hazardous statistical practice; it cannot be justified or validated with available experiments.

Instead, we have adopted a more generic approach which uses the experimental results directly. It has been observed in discussions with personnel at CNVEO that the shape of the distribution of the flat-wall range measurements is approximately Gaussian. We then model the noise as an additive Gaussian process with the standard deviation of the measurements being range dependent. The dependence of the standard deviation on the range will be described by a curve, piecewise-linear in form, that interpolates the actual experimental results. For any range that exceeds the maximum range for which the standard deviation has been determined empirically, the associated standard deviation will be set to the maximum observed value.

3.3.3 Dropouts

The final step in the LADAR scene simulation is an effort to incorporate the effect of "dropouts", as explained above. It was decided to declare *every* pixel in the "sky" a dropout (since the amplitude of the reflected optical field is effectively zero) and a certain percentage (determined by an algorithm parameter) of the remaining (non-sky) pixels as dropouts. The pixels where dropouts will occur are randomly selected. The values assigned to these pixels were *uniformly chosen* over the dynamic range.

3.3.4 Illustrations

In Figure 7 we show a "scene" with several targets, some clutter, (dropout) noise in the sky, but no obscurations or other noise. The dynamic range of depth values necessitates 16 bits of brightness resolution but only the *upper* 8 bits are shown; consequently, it is impossible to discern any detail *within* the targets and other structures at any given range. Figure 8 is the same scene as Figure 7, except that only the *lower* 8 bits of brightness resolution are shown, accounting for the periodic appearance. The lack of detail in the upper 8 bits is particularly evident in Figure 9, a blow-up of a portion of Figure 7. Figure 10 shows the *lower* 8 bits of the same insert, revealing the internal structure of the objects, and Figures 11, 12, and 13 show, incrementally, the various effects of the sensor model, first adding blur, then range-dependent Gaussian noise, and finally the full model including the (non-sky) dropout noise. Finally, Figures 14 and 15 show, respectively, the upper and lower 8 bits of the full (original) scene with all aspects of the simulation program.

The pictures are as diabolical as they appear. There is virtually no resolution in the image of the upper 8 bits (due to the large range spanned by the full scene) and there is very high-frequency, nearly periodic variation in the image of the lower 8 bits (due to the low elevation of the sensor and the large ranges).

4 Training: Probe Optimization

4.1 Alignment

Recall that each location in the image is associated with a subimage field in which information is extracted at predetermined offsets relative to that location. In order to disambiguate hypotheses, this field must be larger than the minimum rectangle required to surround all hypotheses, i.e., target templates. Recall also that there is an individual template image corresponding to each of the three target types for each ten degree rotation in the ground plane. These training images are shown in Figures 3, 4 and 5 for the truck, tank and APC, each in 18 of the 36 aspects. (Actually, the inside template values must be modified to account for blurring, but this is done "on-line" based on the sensor model (more specifically, the blur weights) and the calculated range values.) Previously, in Phase I, we had registered the entire set of geometric shapes by aligning the centers of rectangles circumscribing their silhouettes; see [2]. Alignment is an important issue in that all tests through Round 3 are relative to this registration. Due to the highly non-isotropic nature of the shapes of the actual military targets, we decided to explore

several other methods, and found that the most effective procedure was to align the *centers of mass* rather than the centers of the circumscribing rectangles. The high resolution, ray-traced, targets templates are then mutually registered by aligning their centers of mass computed from the pixels on target. This provides an origin for a reference coordinate system, and this origin may then be regarded as the image location at which we are attempting to detect and classify a target. In the ensuing discussion, image coordinates in the field of view are then all relative to this reference point.

4.2 Occlusion and Semi-Targets

A crucial issue is target *occlusion*; clearly the algorithm must have the capability to detect and classify targets which are partially obscured by terrain anomalies, and perhaps also by other targets, although we have focused on the former.

First, it is evident and visually apparent that the range values are particularly ambiguous for pixels corresponding to the image areas near the bottom of the vehicles. Consequently, we decided to restrict the locations of "outside" points for probes (see §4.3 below) to an arc-like zone (or "halo") lying over the templates, whereas the "inside" points may still lie near the ground.

Moreover, after studying the imagery supplied by CNVEO, it was decided to concentrate on obscurations from below, which occur, for example, when targets are partially occluded by small hills, brush, and other obstacles in the field of view. Ideally, as described in [2] and other previous reports, obscurations would be accommodated by appending the list of hypotheses

to include the "semi-targets" composed of partial silhouettes. For obscurations from below, one would remove some portion of the target lying below a horizontal line (say one passing through the center of mass, although the algorithm would easily be adjusted for the degree of occlusion) and re-train with the expanded list of hypotheses. (In particular, the definition of "inside" and "outside" points must be modified accordingly.) However, due to time constraints, we decided to do something less ambitious, namely to train *separately* for the "half-targets" simply by restricting the locations of probes to an area above the center of mass. Figures 16, 17 and 18 show the collection of half-targets for the three vehicles. The actual search would then consist of two distinct steps: searching for the full targets with the original tests (with "outside points" restricted from certain zones as indicated above), and then searching for the "half-targets" with the customized tests. In actuality (and quite remarkably), the examples reported in the Figures included in this report were computed using *only* the training data for half-targets.

We emphasize again that this two-stage procedure would be less efficient than had we grouped the targets and half-targets into *one* collection of 216 hypotheses, for in that way the tests would be constructed to disambiguate *between* targets and half-targets, rather than simply *among* targets and *among* half-targets. Still, we have found that our tests are sufficiently powerful that nearly no erroneous classifications resulted from confusing a half-target of one type with any target of another type; see §5.

In what follows we shall describe the training mechanism for the full targets; the procedure for the half-targets is identical. Moreover, whereas not restated, it should be remembered that the outside tests are in fact

constrained to lie away from the area under the vehicles, as described above.

4.3 Decision Tree

4.3.1 Detection: Rounds 0 and 1

When Round 0 is implemented we are at the top of the decision tree and, basically, we are attempting to distinguish objects from background. Moreover, this filter must be applied at every image location. Consequently, we seek speed and generality, the former by restricting the number and complexity of the probes, and the latter by requiring that every probe should provide information about *every* hypothesis, i.e., about the presence or absence of each object in each orientation. Moreover, the only information about the actual objects (i.e., the target templates) that is utilized in all the probes until Rounds 4 and 5 is the silhouette; the internal structure of the objects (i.e., the actual range data) is only exploited further down the decision tree for final disambiguation and hypothesis verification.

Recall that we have registered the silhouettes to provide an origin for a reference coordinate system, and this origin may be regarded as an image location at which we are attempting to detect and classify a target within a field of view centered there. Relative to this coordinate system, a probe in Round 0 is a point with a label, in this case indicating whether the point should be “inside” or “outside” the *collection* of (registered) shapes. For simplicity, we chose the same number J of inside and outside points, yielding $2J$ points in all and denoted by (I_j, O_j) , $I_j = (I_j^1, I_j^2)$, $O_j = (O_j^1, O_j^2)$, $j = 1, 2, \dots, J$. In the current implementation, for example, $J = 20$.

Ideally, the points must be chosen such that each I_j lies *inside* each shape and each O_j lies *outside* each shape. In addition, we found it useful to restrict the points from strongly clustering. The inside and outside points were chosen by designing two cost or "energy" functionals over sets of points. One cost functional governed selection of the inside points and the other governed selection of the outside points. Details of the proprietary procedure for optimization of tests are described in the technical documentation, delivered with the software.

Round 1 is similar to Round 0 except that the original 108 shapes are divided into 18 *groups*, each group consisting of six shapes; the first group is the tank, call it Object A, at the six rotations 0, 10, 20, 30, 40, and 50 degrees, the second group is the tank at rotations 60, ... ,110 degrees, etc; this accounts for the first 6 groups. The next 6 groups are defined in the same manner relative to Object B, the truck, and the last 6 groups relative to Object C, the APC. This grouping was done to facilitate discrimination among hypotheses. *The probes in Round 1 are thus group-dependent; they needn't accommodate all 108 objects and can therefore be sufficiently discriminating to eliminate most of the false positives resulting from Round 0.*

The probes are constructed in a similar fashion to those in Round 0, except that there are now separate energy functions for each group $g = 1, \dots, 18$. These are defined in a manner similar to the cost functionals used for optimization of the Round 0 tests. Details are provided in the separate technical description of the algorithm.

4.3.2 Classification: Rounds 2 and 3

At this stage of the decision tree we now wish to compare many hypotheses simultaneously and retain those with some reasonable probability of occurrence at the current location. Given that we have reached this stage, there is the strong likelihood that Object A, B, or C (or clutter) is within the field of view, although not necessarily at offset zero relative to the center of the field of view, i.e., the current image location. Remember that, ideally, a hypothesis is confirmed at this location exactly when the offset is zero. Since we are no longer primarily interested in separating objects from background, and since there are as yet no specific hypotheses to entertain (only active groups), we desire probes which effectively disambiguate among all relevant pairs of hypotheses.

The probes in Rounds 2 and 3 involve relational template matching. Each probe is a labeled *pair* (u, v) of locations. Again only the object silhouettes are utilized during these rounds. The label depends on which template or offset (translated) template is present at the reference point (i.e., within the field of view) and indicates the positioning of probe coordinates relative to the silhouette. Specifically, the label of (u, v) for hypothesis l is denoted (I, O) , (O, I) or (I, I) according to whether u is inside and v is outside shape l , vice-versa, or both u and v are inside; we do not consider pairs for which *both* points lie outside any one of the templates.

Now given two shapes, say l and k , with l at offset 0 and k at offset b (a vector) relative to the origin of the field of view and given a set x of N pairs of points, $x = (u_n, v_n), n = 1, 2, \dots, N$, we define the discrepancy $D(x; l, k, b)$

between l and k in terms of differences of label types between l and k for the probes at locations x . Details of the definition of D are provided in the technical description of the algorithm, delivered with the software.

As with Round 1, we define separate cost functionals H_g for each group, $g = 1, 2, \dots, 18$, in order to find optimal probes (relative to D) to distinguish the shapes in C_g from those corresponding to the other objects. Let C_1, \dots, C_6 denote the groups for Object A, C_7, \dots, C_{12} denote the groups for Object B, and C_{13}, \dots, C_{18} denote those for Object C. Fix g , say $g \leq 6$ (the other cases are similar) and $x = (u_n, v_n), n = 1, \dots, N$. The cost functional for group g (Object A) is defined as a function of x which measures how difficult it is for the probes associated with locations x to separate group g from all competing presentations of Object B and Object C. Details of the definition are provided in the technical description of the algorithm, delivered with the software.

We used coordinate-wise descent to find a value of x^* for which the cost functional of group g is small, thereby providing a set of probes which separates as well as possible the particular presentations of Object A represented by group g from *all* competing presentations of Object B and Object C.

When a field of view is fixed and the search is performed, the image intensity values are observed at the coordinates in x^* and for each hypothesis $l \in C_g$ the observed depth differences for the pairs (u_n, v_n) are assigned one of the labels (I, O) , (O, I) or (I, I) using "floating thresholds" designed to minimize the probability of a detection error if, in fact, hypothesis l (at offset 0) is true. This is the hypothesis-driven segmentation we have mentioned in previous descriptions of the algorithm. The result of this stage of the

search is a collection of specific hypotheses for which the distance between the observed and template values falls below a specified level.

Before Round 3 is implemented at a particular location, there is necessarily a list of active hypotheses at this location; otherwise the search would have terminated with the outcome "no confirmations." We then designed hypothesis-specific tests in order to separately confirm or deny each of the active hypotheses. Thus, for each hypothesis l , we define a cost functional depending on probe locations x which measures how difficult it is for the probes associated with locations x to separate hypothesis l from all competing presentations of the other object types. Minimization of the cost functional for hypothesis l results in a set of probes for testing each shape against all relevant alternatives, and a shape will remain active in the search procedure only if the associated distance is suitably small. At this stage we find that the probes characterize subtle differences among the shapes; this is now possible for higher thresholds than in Round 2 since there are many fewer patterns to disambiguate.

4.3.3 Verification: Rounds 4 and 5

The result of Round 3 is a list of hypotheses which are active at the given image location; of course nearly all image locations have no pending confirmations by this point. The purpose of Round 4 is to exploit the *internal structure* of the objects, that is the depth differences among locations within the silhouette, to filter or screen the list of active hypotheses. Round 5 then disambiguates among confirmations which lie in close proximity.

Let k denote the label of an active hypothesis at a given image location after Round 3, and let $X_i^k, i = 1, \dots, n_k$, denote the (relative) depth values for the pixels on the target template associated with k , i.e., X_i^k is the (relative) range to pixel i for the particular target-aspect pairing indicated by k . (Actually, as mentioned above, these ideal depth values must be adjusted for the blurring effects of the sensor; we do this by simply (i) applying the known sensor blur model to the idealized ray-traced template and (ii) excluding pixels that are within the radius of the blur support from the boundary of the template.) Let Y_i denote the actual measured intensity value at pixel i . We wish to check whether or not the observed values are consistent with the presence of hypothesis k at the reference point.

The consistency check is done by defining a statistic T^k , chosen by design to be a suitably invariant dissimilarity measure between the observed data Y_i and the hypothetical values X_i^k currently being considered. T^k is designed so that it has a distribution function of known form when the active hypothesis k is true. T^k is also designed so that the way in which its values differ from the so-called null distribution when k is false is well understood. Consequently, we were able to screen effectively simply by following the standard paradigm of testing a statistical hypothesis. We are able to implement this test so that no false negatives are introduced at this stage of the decision tree and it is still a powerful discriminant between hypotheses. Details of the Round 4 consistency check are provided in the technical description of the algorithm, delivered with the software.

Finally, in Round 5, we compare pending confirmations which lie in close proximity to decide which target to confirm at the given location. This is

easily and effectively accomplished by comparing values $T^k, k = 1, \dots, K$, where K denotes the number of pending hypotheses in the comparison view field. Details of the Round 5 selection among competing hypotheses are provided in the technical description of the algorithm, delivered with the software.

5 Experiments.

The first experiment shows the results of the algorithm on the data given by Figures 14 and 15. In Figure 19, five vehicles have been correct identified; we have omitted the sky, displayed only the lower 8 bits, and outlined the targets for easier viewing. There are no substitution errors and no false positives. This scene contains two M35s, two M60s, and one M113 at a range of 3 kilometers. None of the objects are occluded. In addition, there are five pieces of clutter and dropout noise which has probability 0.1 of replacing each pixel in the scene with a random uniformly-distributed two byte integer.

Figures 20 and 21 show another simulated scene and Figure 22 another example of the output of the ATR algorithm. In Figure 22, five vehicles have been correct identified; again, we have omitted the sky, displayed only the lower 8 bits, and outlined the targets for easier viewing. There are no substitution errors and no false positives. This scene also contains two M35s, two M60s, and one M113 at a range of 3 kilometers, however it is considerably more challenging than the first example because all five targets have been partially obscured by burying them. In addition, there are five pieces of clutter and dropout noise which has probability 0.1 of replacing each pixel

in the scene with a random uniformly-distributed two byte integer.

Finally, Figures 23 and 24 depict a still more challenging example, with the results shown in Figure 25. In Figure 25, eight vehicles have been correctly identified; again, we have omitted the sky, displayed only the lower 8 bits, and outlined the targets for easier viewing. There are no substitution errors and no false positives. This scene also contains three M35s, two M60s, and three M113s at a range of 3 kilometers. This scene is still more challenging than the first and second examples because it contains some of the smallest and most difficult to recognize targets (a front view of an APC) and four of the objects are partially obscured. In addition, there are twenty pieces of clutter and dropout noise which has a probability of 0.1 of replacing each pixel in the scene with a random uniformly-distributed two byte integer.

We note again that the poor appearance of these pictures is, in fact, a fair representation of the actual data being processed. There is virtually no resolution in the image of the upper 8 bits (due to the large range spanned by the full scene) and there is very high-frequency, nearly periodic variation in the image of the lower 8 bits (due to the low elevation of the sensor and the large ranges).

Figures

Figure 1A. Blow-up of simulated scene at range 3km

Figure 1B. Blow-up of real data at range 1km

Figure 2. Decision tree for object recognition

Figure 3. Templates for M35 truck at 18 aspects

Figure 4. Templates for M60 tank at 18 aspects

Figure 5. Templates for M113 APC at 18 aspects

Figure 6. Templates for M35 truck subjected to "resolved-target" sensor model

Figure 7. Most significant 8 bits of ray-traced image (no blur, no noise)

Figure 8. Least significant 8 bits of ray-traced image (no blur, no noise)

Figure 9. Blow-up of most significant 8 bits of ray-traced image (no blur, no noise)

Figure 10. Blow-up of least significant 8 bits of ray-traced image (no blur, no noise)

Figure 11. Blow-up of least significant 8 bits of ray-traced image after application of sensor model

Figure 12. Blow-up of least significant 8 bits of ray-traced image after application of sensor model and noise model

Figure 13. Blow-up of least significant 8 bits of ray-traced image after application of sensor model, noise model and injection of dropouts

Figure 14. Most significant 8 bits of a simulated scene containing 5 objects and 5 pieces of clutter

Figure 15. Least significant 8 bits of a simulated scene containing 5 objects and 5 pieces of clutter

Figure 16. Training templates of half-targets for M35 truck at 18 aspects

Figure 17. Training templates of half-targets for M60 tank at 18 aspects

Figure 18. Training templates of half-targets for M113 APC at 18 aspects

Figure 19. Detected objects (boxed) in scene depicted in Figure 15

Figure 20. Most significant 8 bits of a simulated scene containing 5 objects and 5 pieces of clutter

Figure 21. Least significant 8 bits of a simulated scene containing 5 objects and 5 pieces of clutter

Figure 22. Detected objects (boxed) in scene depicted in Figure 21

Figure 23. Most significant 8 bits of a simulated scene containing 8 objects and 20 pieces of clutter

Figure 24. Least significant 8 bits of a simulated scene containing 8 objects and 20 pieces of clutter

Figure 25. Detected objects (boxed) in scene depicted in Figure 24

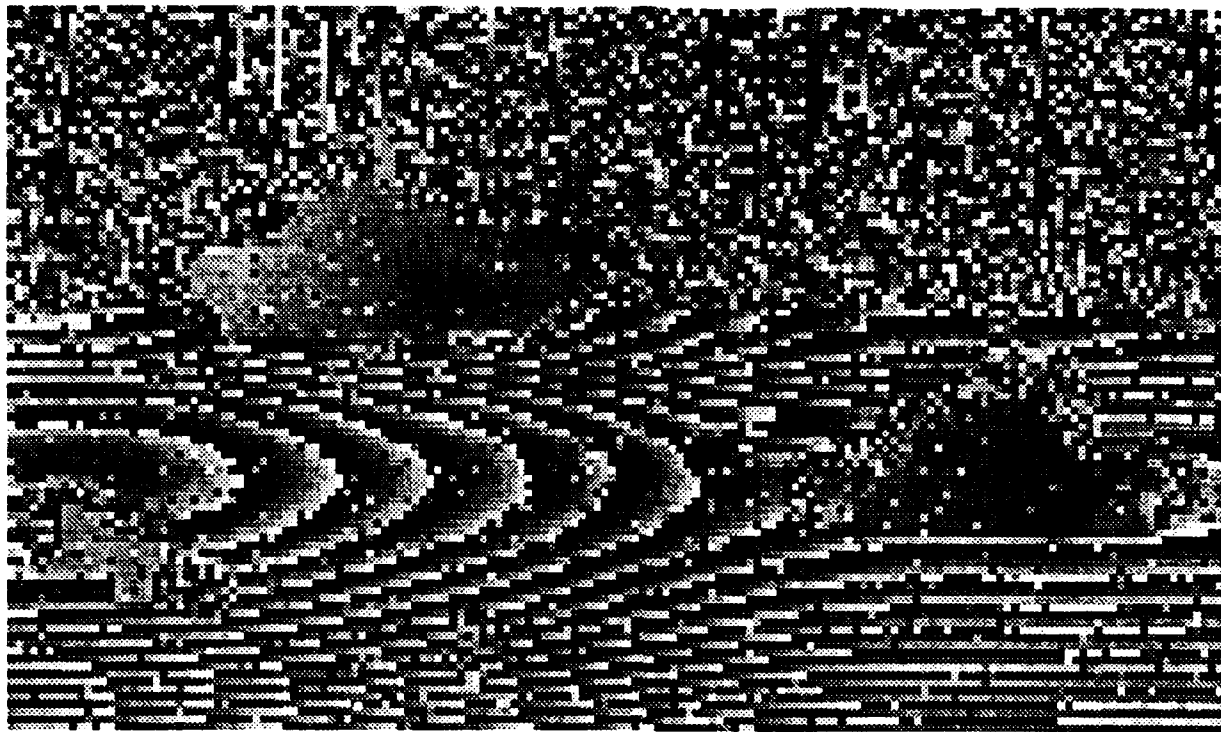


Figure 1A

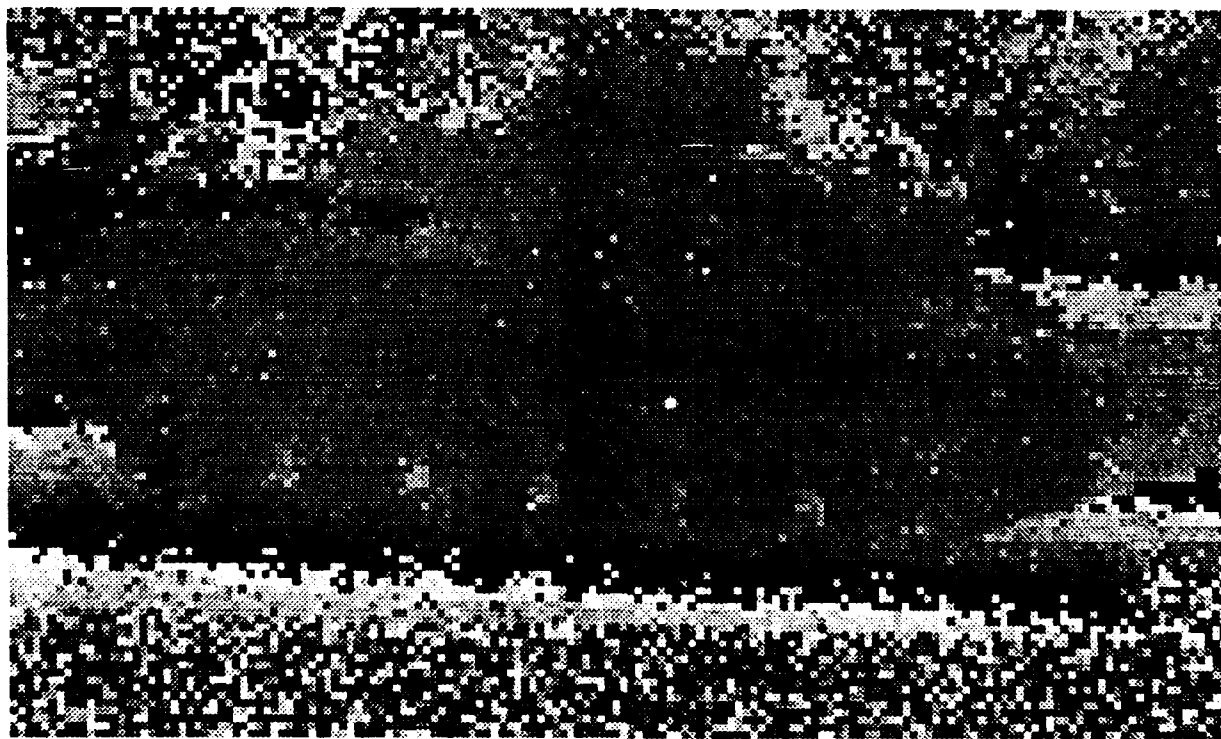


Figure 1B

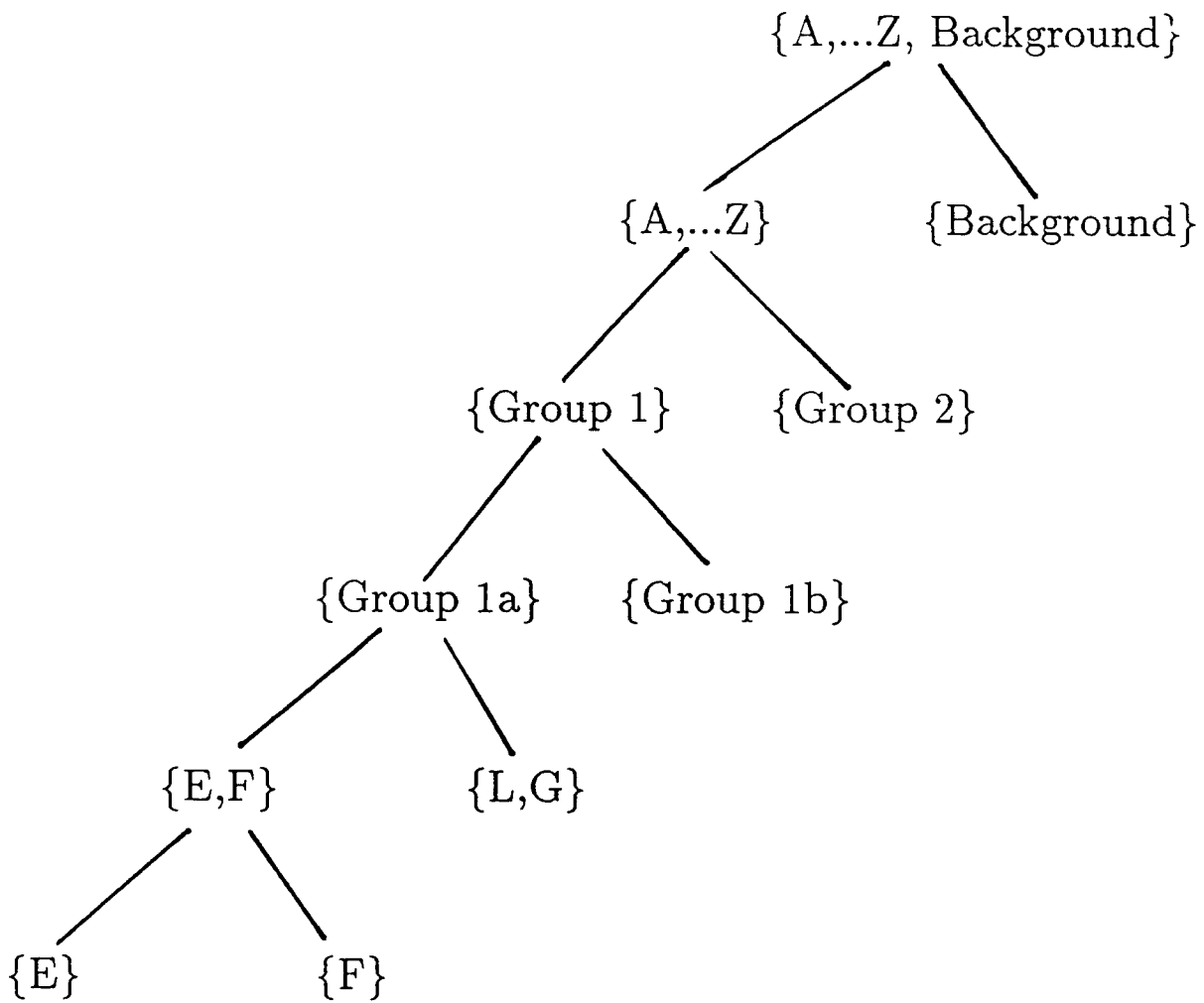


Figure 2. Decision tree for object recognition

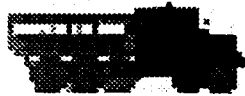


Figure 3

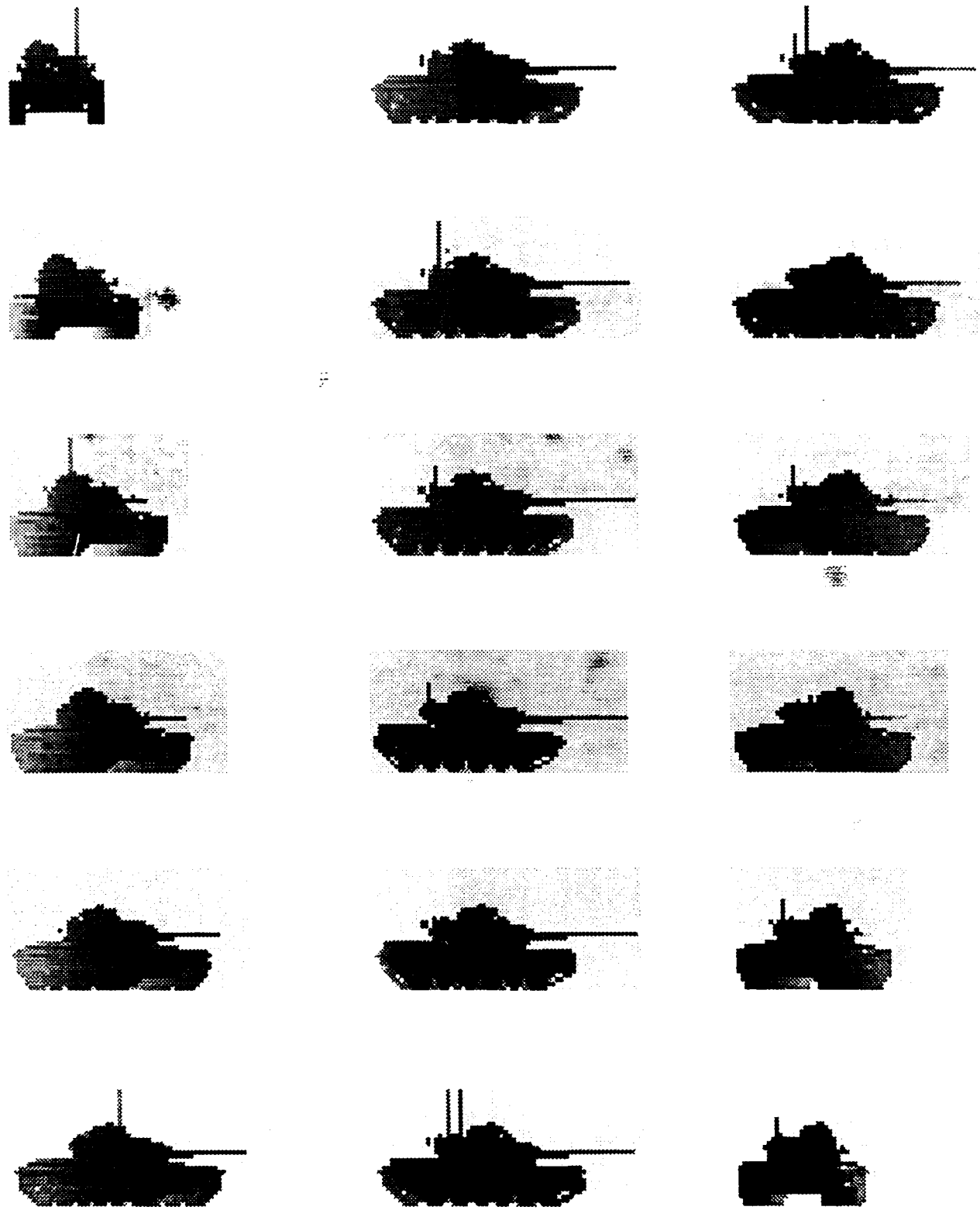


Figure 4



Figure 5

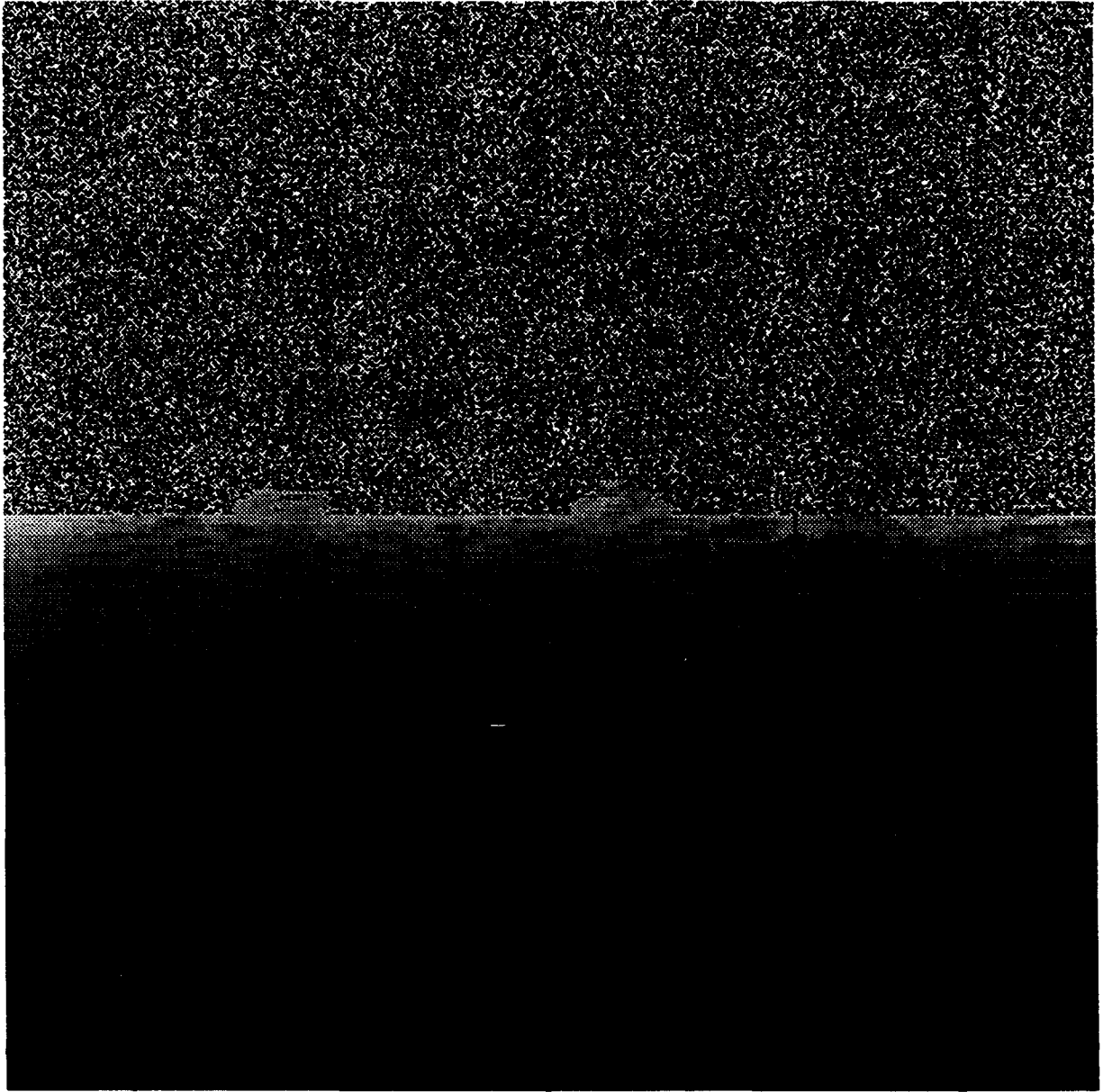


Figure 7

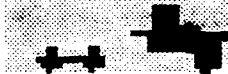


Figure 6

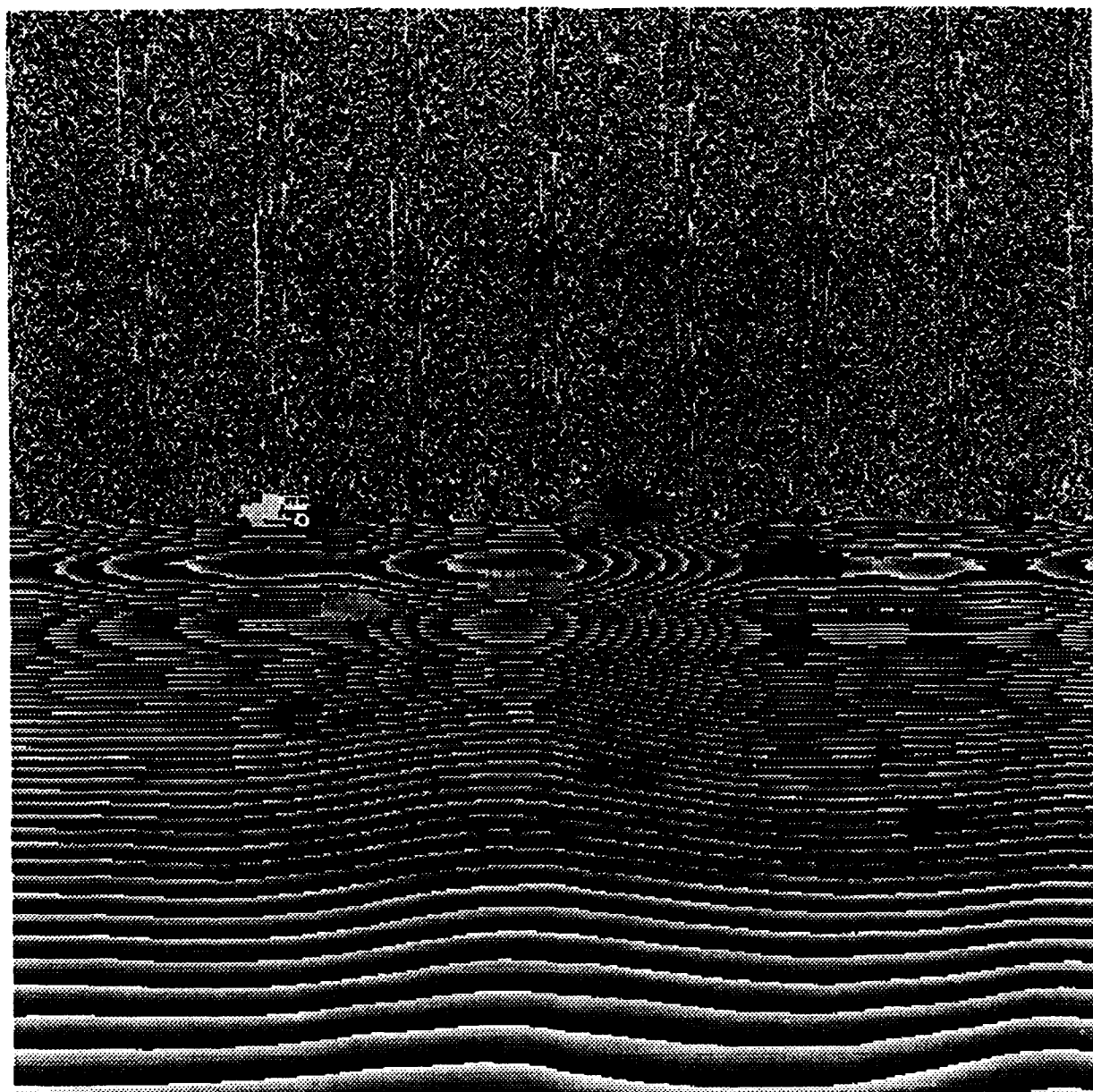


Figure 8

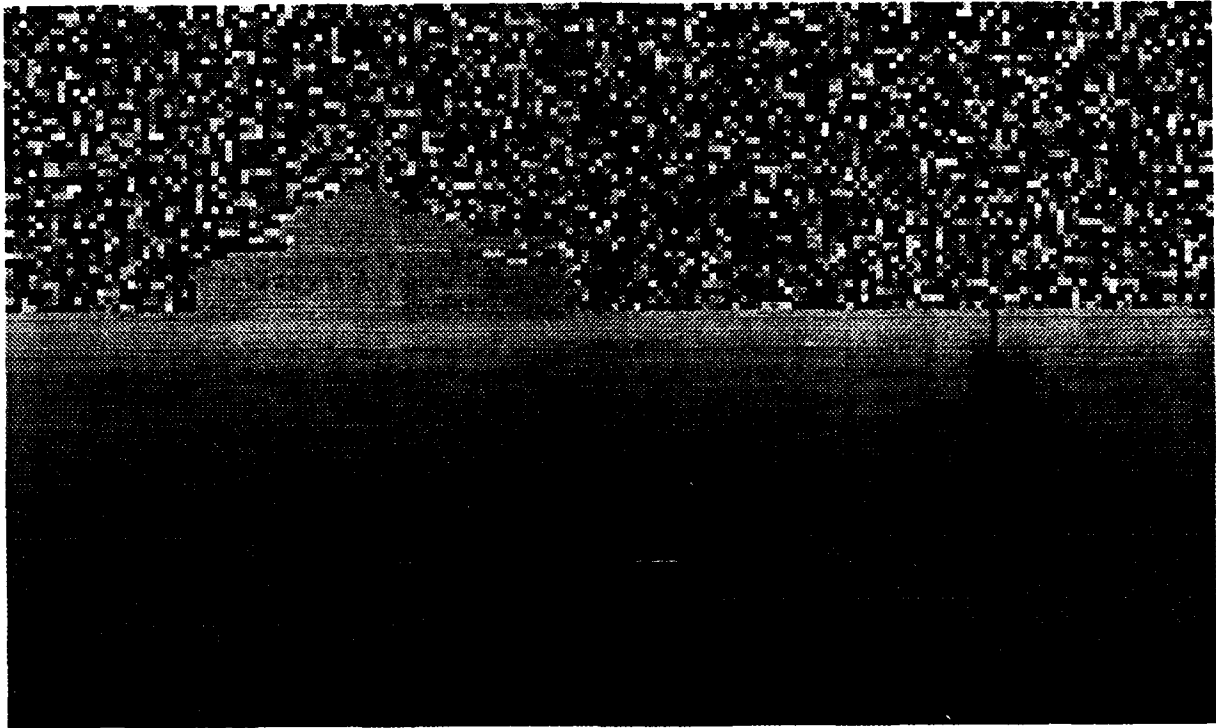


Figure 9

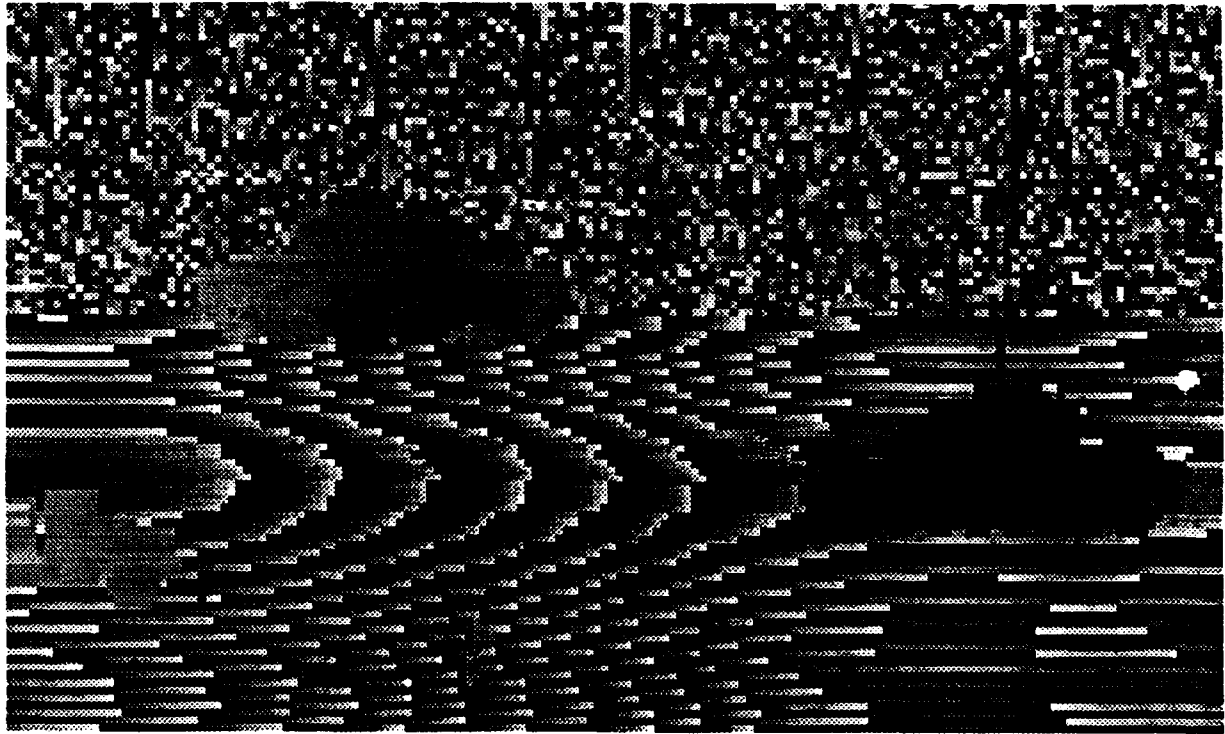


Figure 10

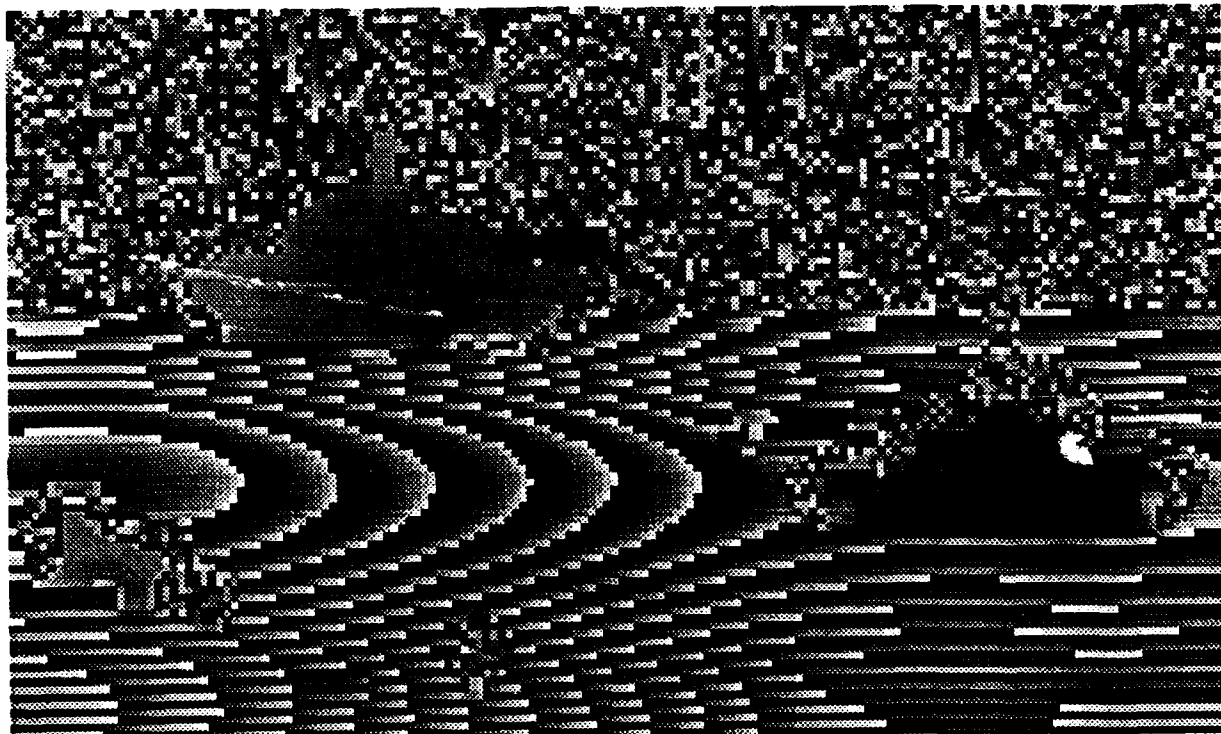


Figure 11

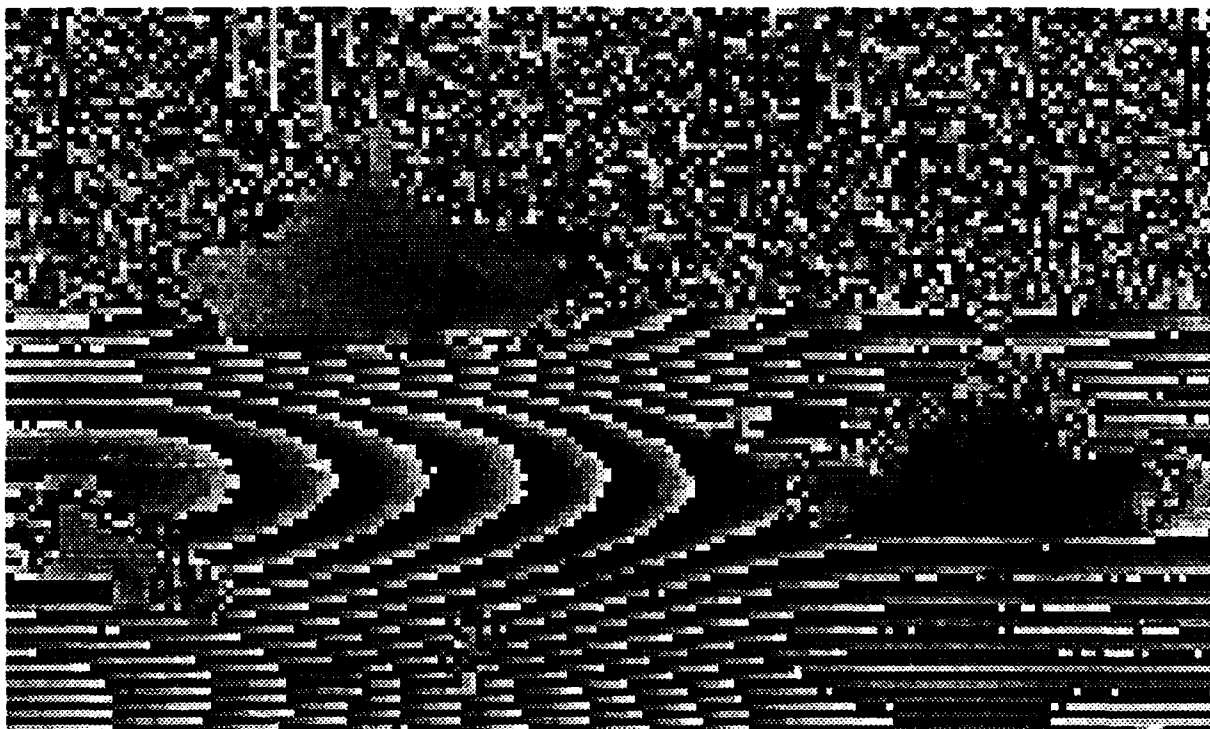


Figure 12

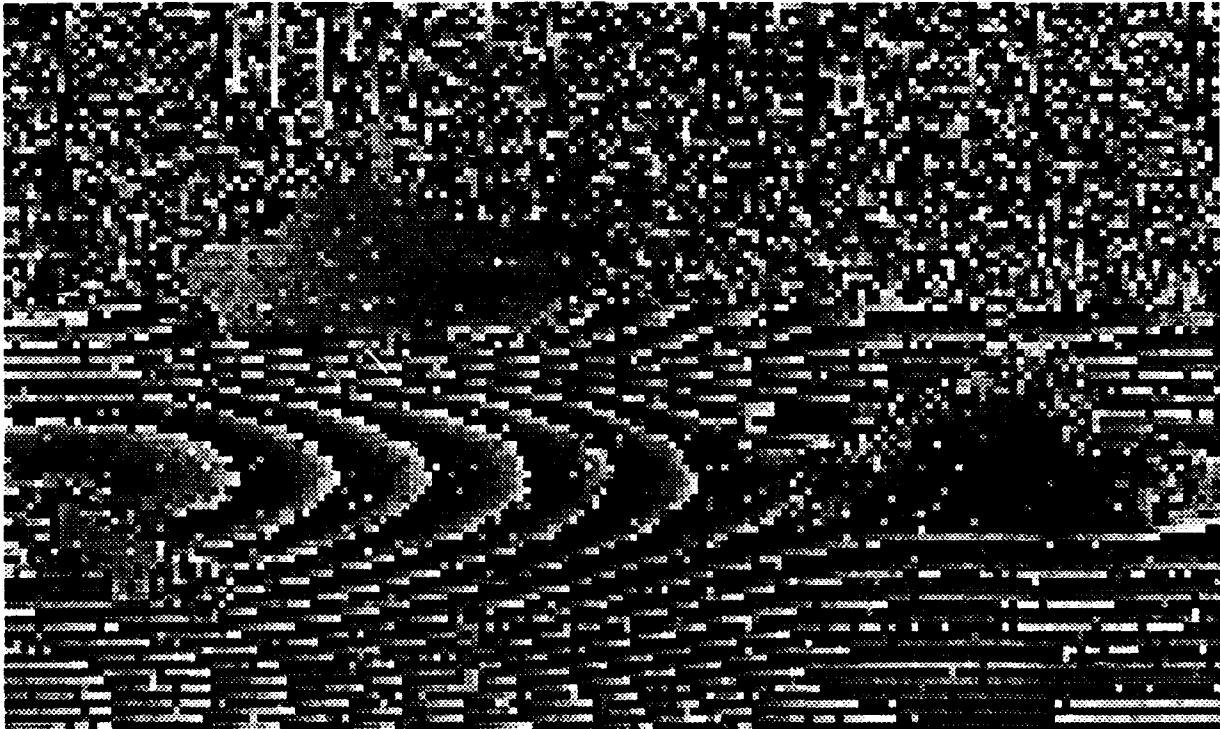


Figure 13

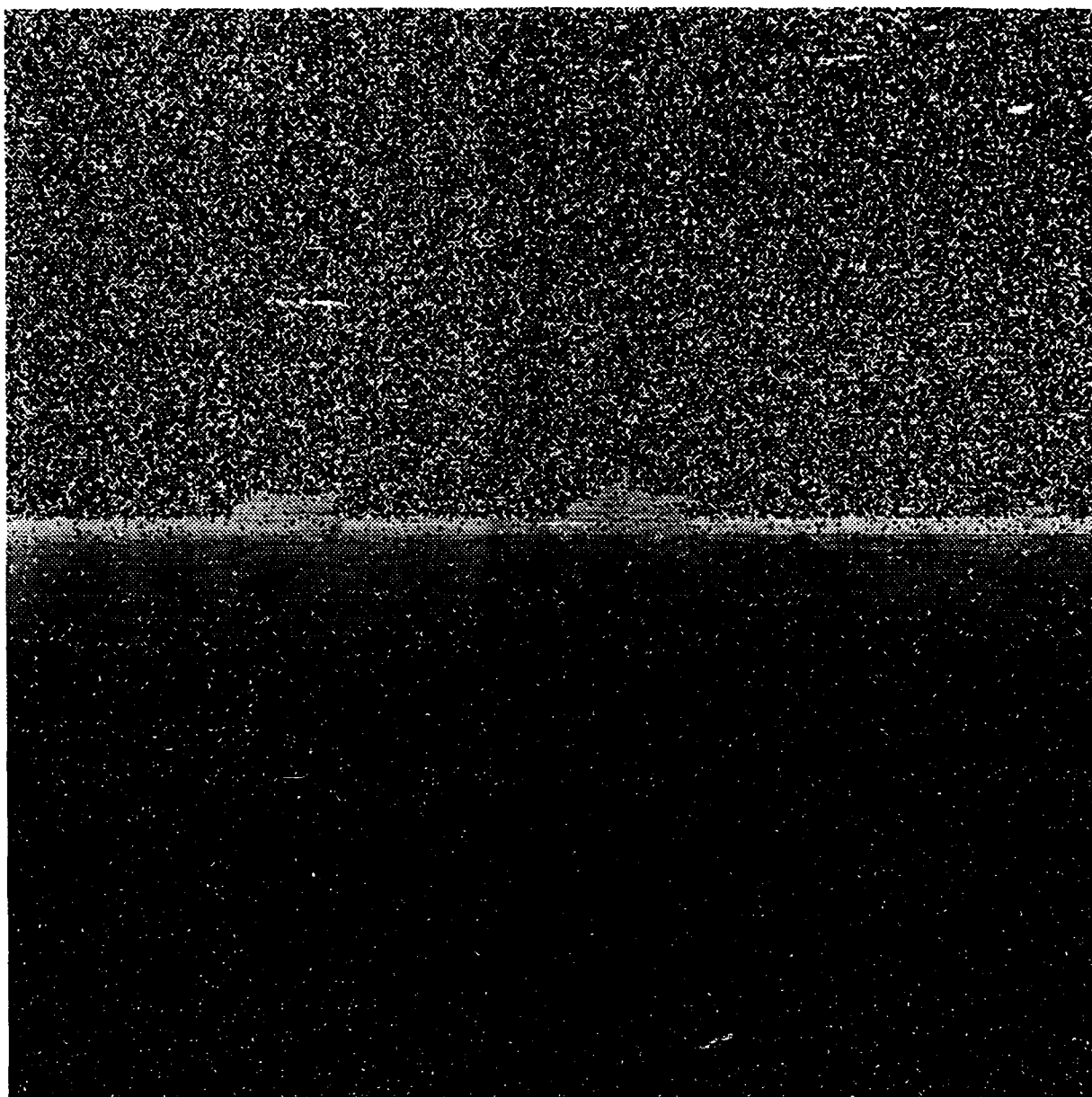


Figure 14

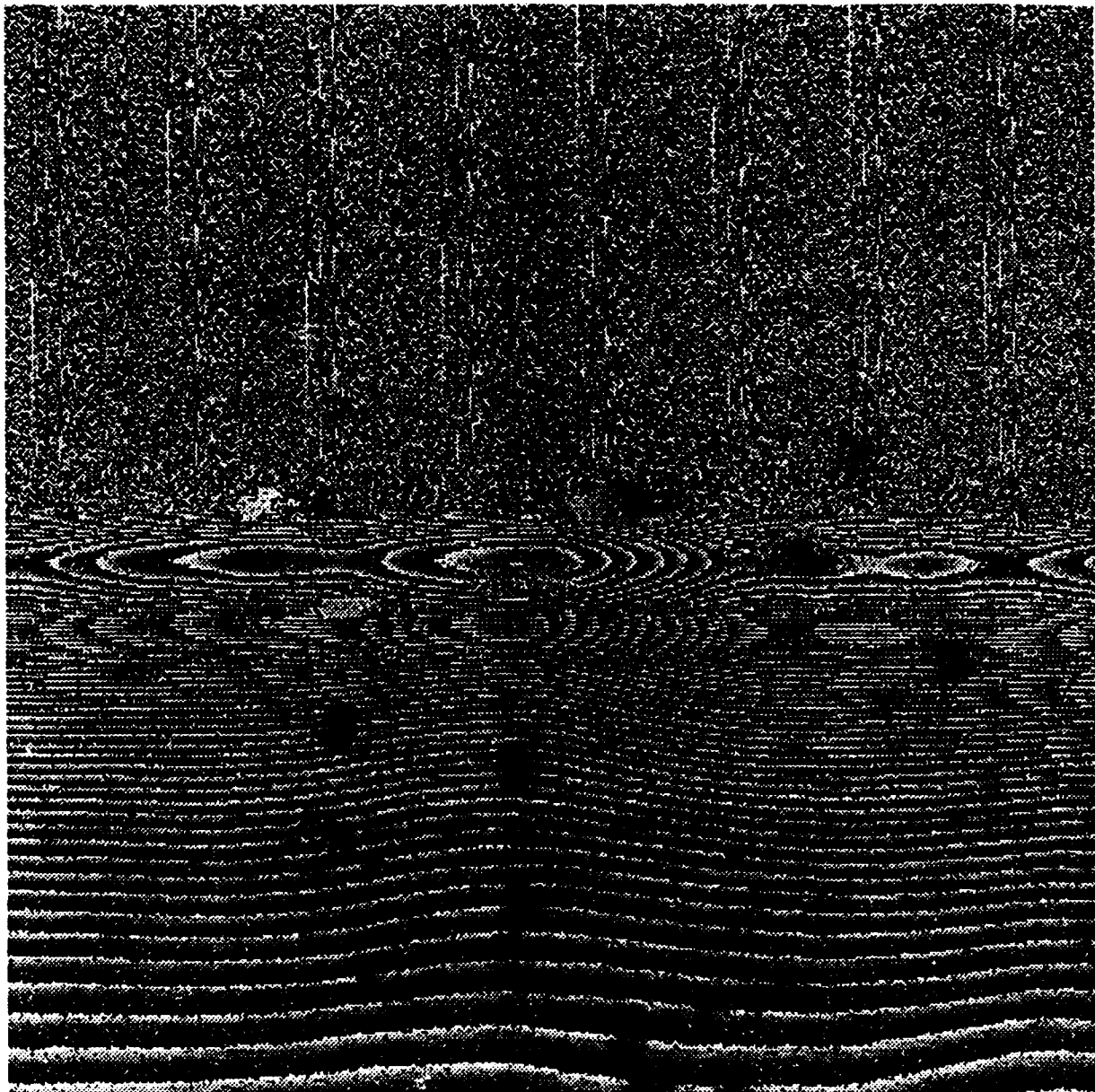


Figure 15



Figure 16

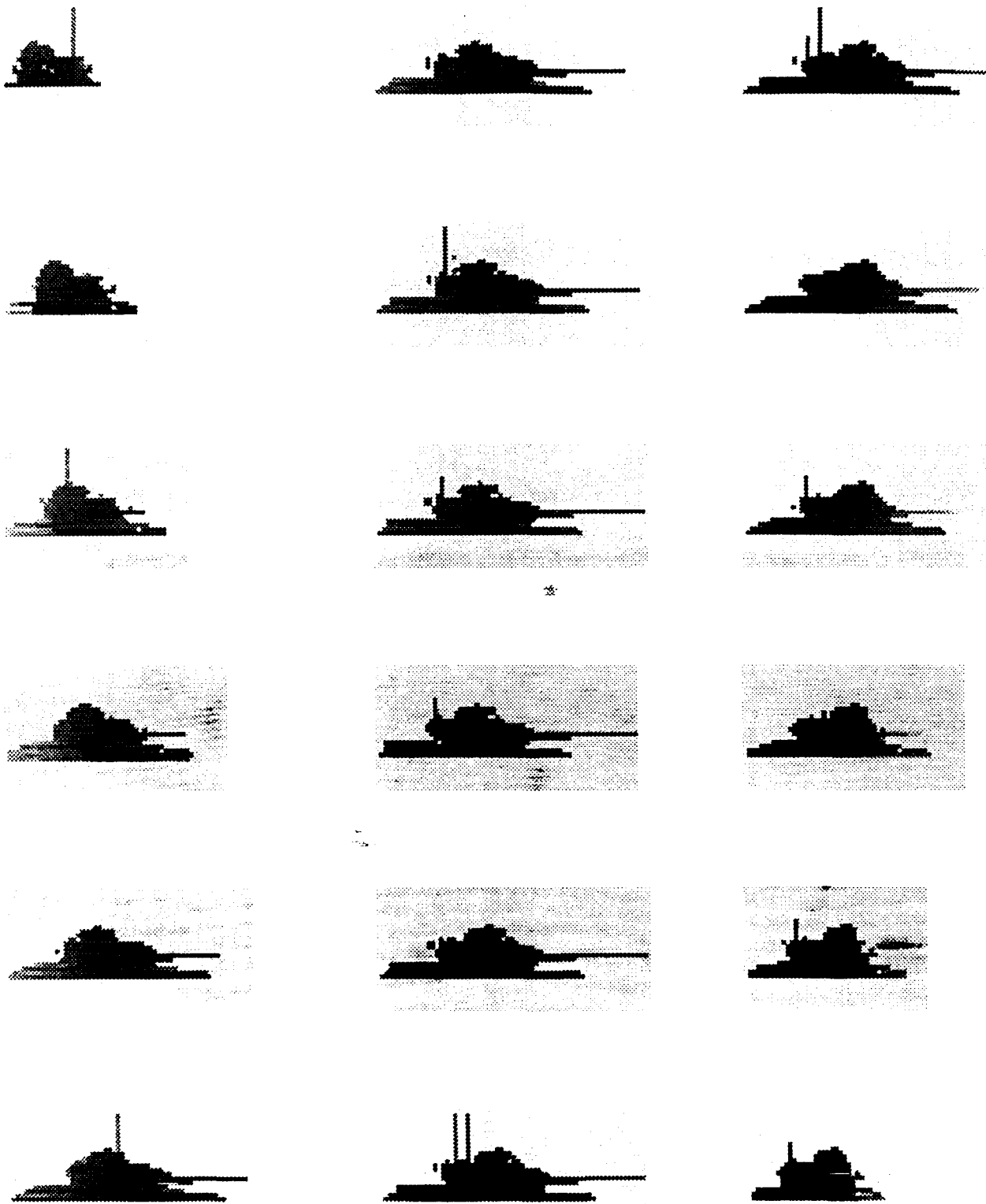


Figure 17



Figure 18

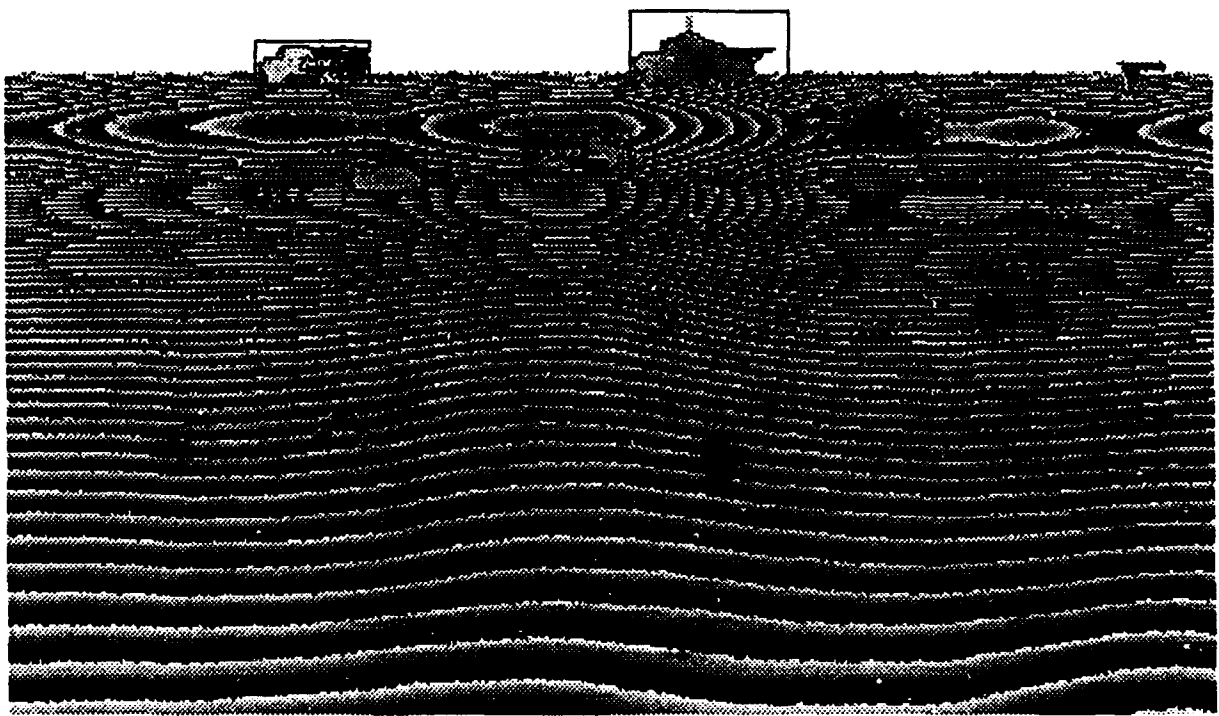


Figure 19

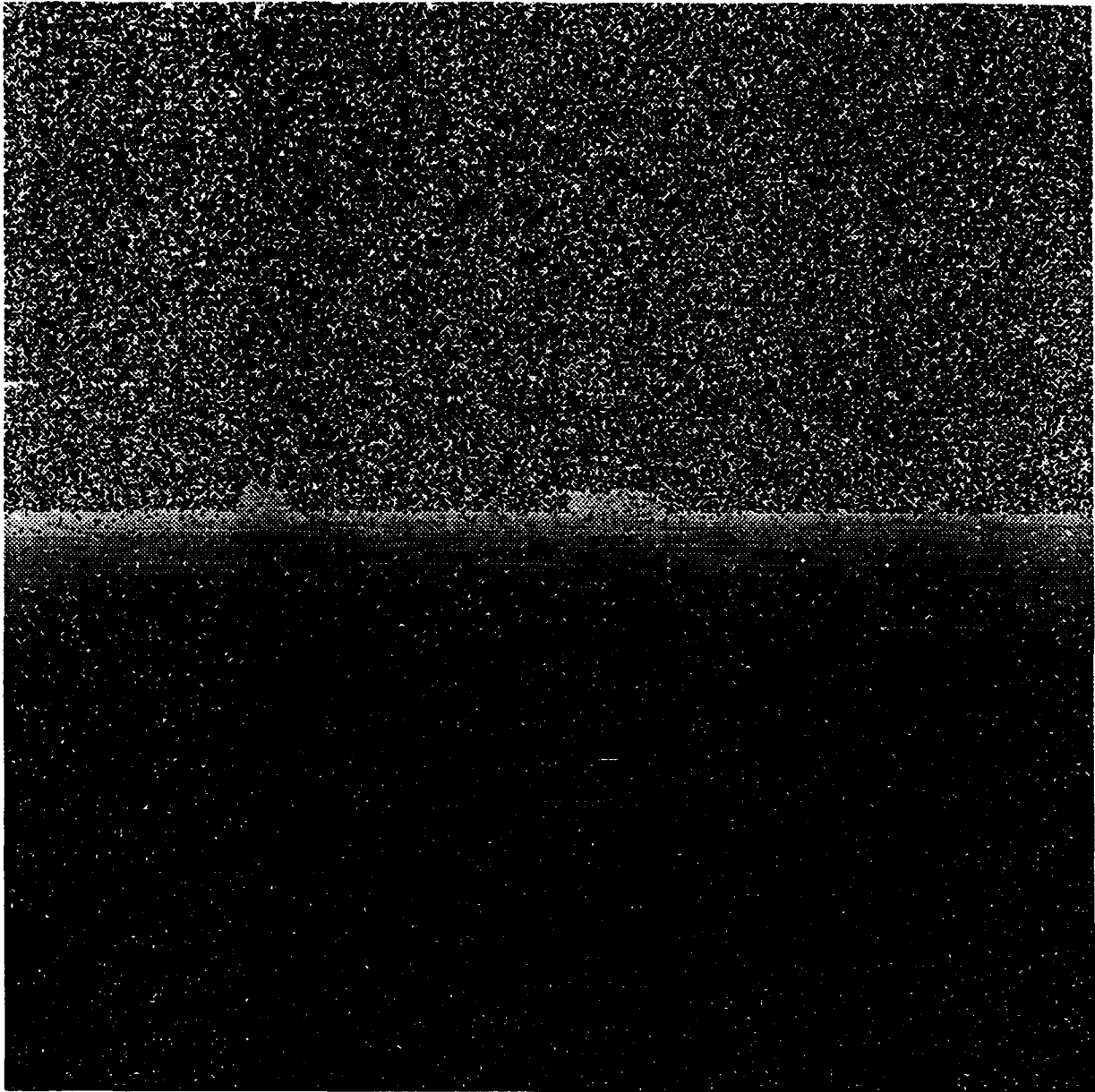


Figure 20

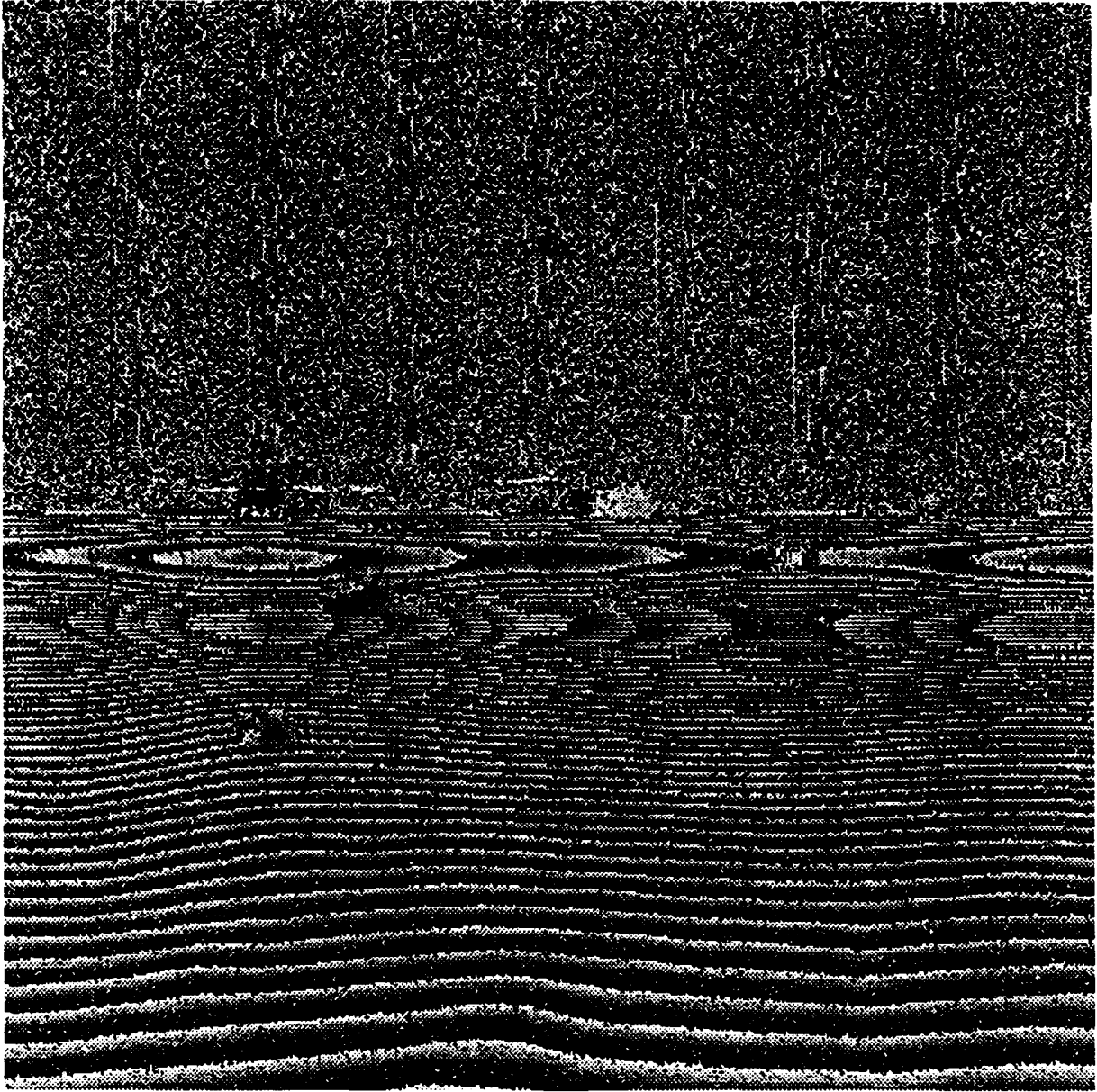


Figure 21



Figure 22

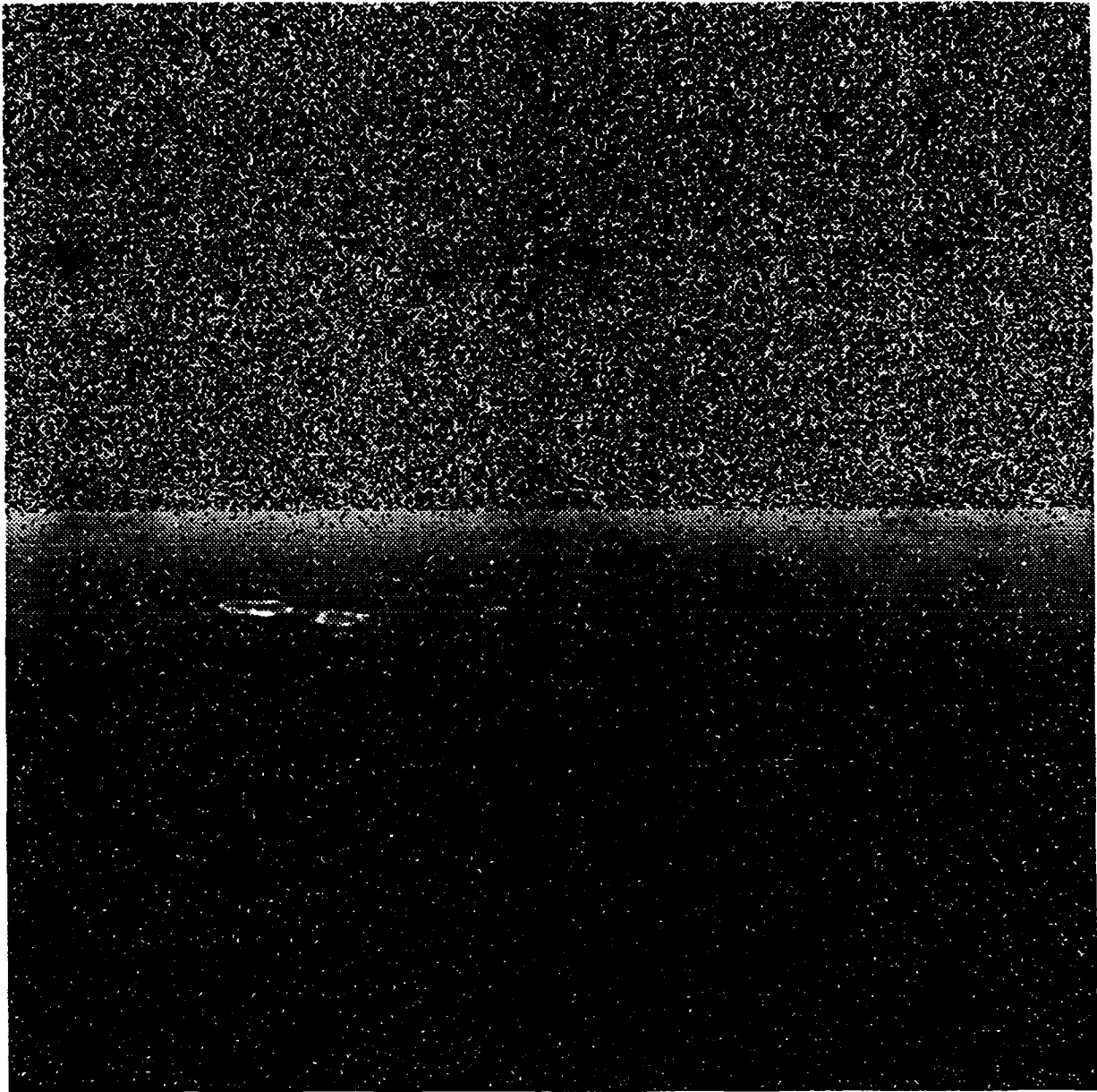


Figure 23

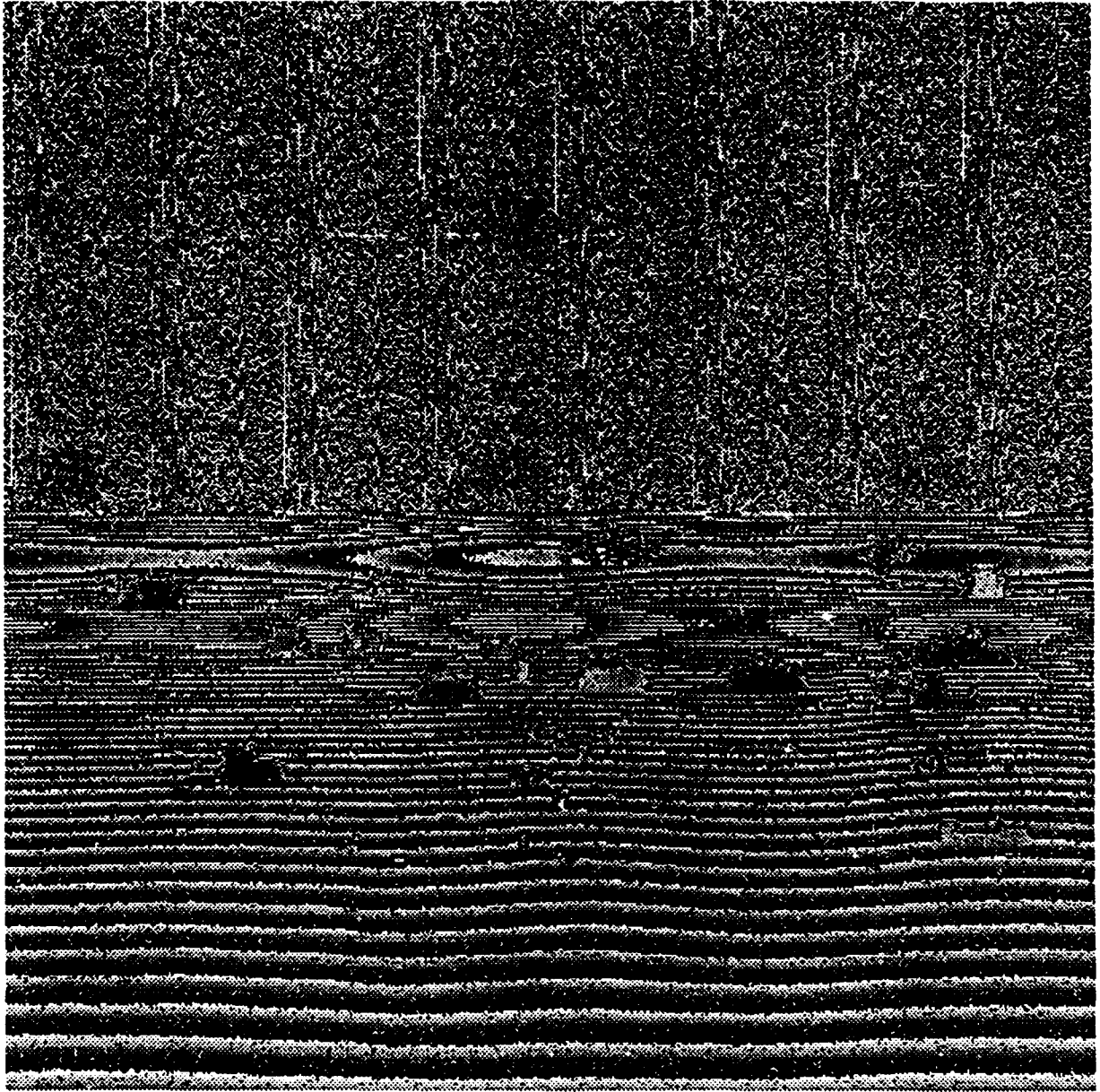


Figure 24

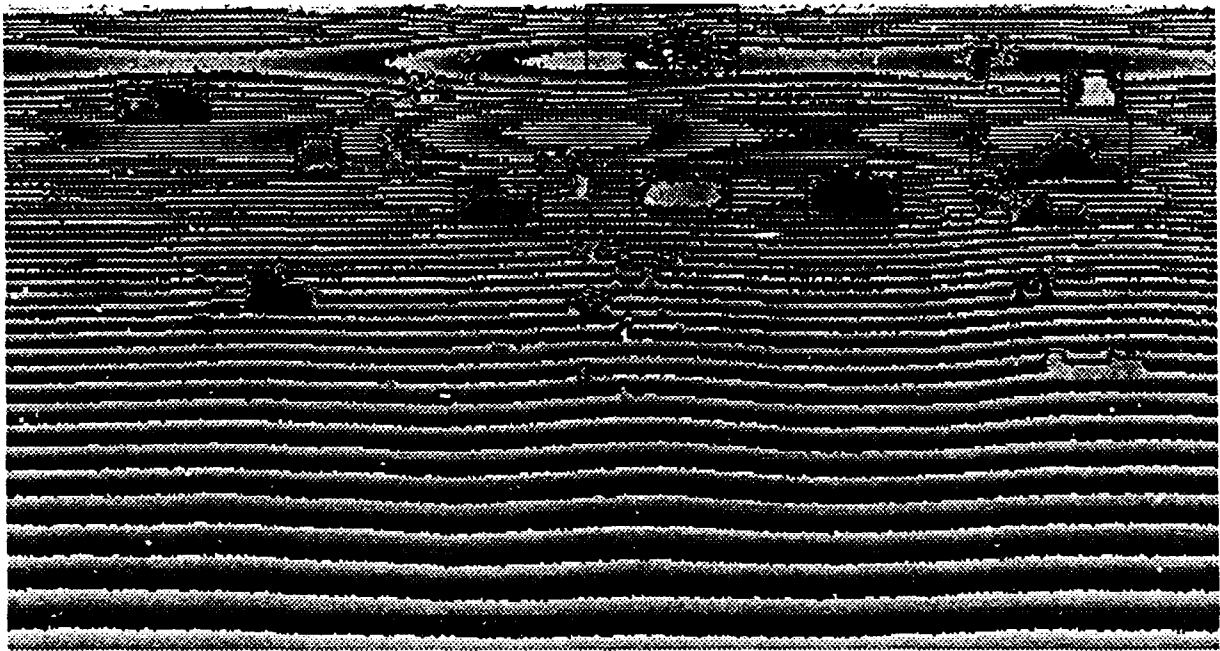


Figure 25

References

1. C.G. Bachman, *Laser Radar Systems and Techniques*, Artech House Inc., (1979).
2. E. Bienenstock, D. Geman, S. Geman, D.E. McClure, *Development of Laser Radar ATR Algorithms: Phase I - Geometric Shapes*, Phase I Technical Report, Mathematical Technologies Inc., (1990).
3. S.C. Cohen, "Heterodyne detection: phase front alignment, beam spot size, and detector uniformity," *Applied Optics*, Vol. 14, 1953-1959, (1975).
4. J. Gilberg, Honeywell Systems and Research Center, Personal communication, July 1990.
5. J.W. Goodman, "Some effects of target-induced scintillation on optical radar performance," *Proc. IEEE*, Vol. 53, 1688-1700, (1965).
6. C.M. Johnson, "Laser Radars," Chapter 37 in *Radar Handbook*, M. Skolnick ed., McGraw Hill, (1970).
7. D. Letalick and A. Oestberg, "Computer model for an imaging laser radar," Contract Report ISSN 0347-3708, September 1983.
8. M.B. Mark and J.H. Shapiro, "Multipixel, multidimensional laser radar system performance," M.I.T. Technical Report (1986).

9. I. Renhorn, O. Steinvall, D. Letalick, K. Gullberg, T. Claesson and A. Widen, "Performance study of a coherent laser radar," *SPIE Proceedings No. 415*, 39-50, (1983).
10. A.H. Reynolds, "Comparison of laser radar modulation formats," *Proc. IRIS Active Systems*, 193-209, (1982).
11. Summary results of scale parameter estimation for errors in TSLR range measurements from 1987 and 1989 Fort A.P. Hill field tests, Data provided by CNVEO, September 1990.