J.G. Verly, R.L. Delanoy, and D.E. Dudgeon

Machine Intelligence Technology for Automatic Target Recognition

Automatic target recognition (ATR)—the use of computer processing to detect and identify targets automatically—is becoming critically important in several military applications. ATR systems can reduce the workload of tactical aircraft crews and decrease the communication bandwidth that remotely piloted vehicles need. ATR will also be a crucial component of future "smart" weapons, missiles programmed to seek out and destroy specified targets. Furthermore, ATR technology has potential commercial application in the field of robotic vision.

The development of a comprehensive ATR system is difficult because the system must handle a variety of targets under a variety of conditions. By using techniques from the field of artificial intelligence, often called machine intelligence, we have developed an experimental ATR system that processes image data from a laser radar and then automatically detects and recognizes specified targets in the data. The system has performed well with real-life and synthetic data.

The goal of an automatic target-recognition (ATR) system is to detect and identify enemy targets—such as tanks, howitzers, and armored personnel carriers (APC)—by using data, typically *images*, gathered from one or more sensors. (See the box "Laser Radar Sensors" for an explanation of laser radar imagery.) In addition, an ATR system might have, and will probably need, access to other information such as geographical maps, navigational data, suspected target locations, suspected target types, and meteorological data. The ideal output of an ATR system would include a list of targets (prioritized by tactical importance) and such attributes as the targets' locations, orientations, and configurations.

(In this article, we will use the words "identify," "recognize," and "classify" interchangeably. The different terms refer to the labeling of a target as belonging to a narrowly defined class such as "tanks.")

Building a comprehensive ATR system is difficult for many reasons. First, a target's appearance in a radar image changes when the target's orientation with respect to the sensor changes or when the target's state of articulation is altered. Second, a target's appearance is also affected by camouflaging, obscuring objects, the time of day or night, or weather conditions. Finally, a target's appearance differs from one sensor to another—e.g., from a millimeter-wave radar to an infrared laser radar—and across different imaging modalities of the same sensor—e.g., range and intensity images as defined in the box "Laser Radar Sensors."

ATR systems have potential use in a variety of military applications. In some applications, such as those on the tactical battlefield, an ATR system must locate a large number of targets within limited geographical areas. In other applications, such as locating truck-borne missile launchers, an ATR system must locate a small number of targets within a large geographical area. ATR systems can aid an aircraft's crew by locating targets automatically. For a remotely piloted vehicle, an ATR system can decrease the bandwidth of data links needed to transfer vital information back to the vehicle's command post. ATR is also a key component of "smart" weapons, missiles programmed to seek out and destroy specified targets. In particular, we are interested in the tactical air-to-ground situation in which an aircraft must automatically detect, identify, and prioritize targets on

Laser Radar Sensors

A typical laser-radar sensor system is shown in the simplified block diagram in Fig. A. In the figure, laser energy is transmitted into the environment with a pulse waveform. When reflected energy returns to the sensor and is detected and recorded, a waveform like the one shown in Fig. B results. E_p, the peak energy, is a measure of the reflectivity of an object encountered by the laser radiation; t_p, the peak location in time, is a measure of the range to the reflecting object. By scanning the laser beam across a field of view, each of these measurements can be made to form a 2-D *image* of the field. In an intensity image, each picture element, or pixel, corresponds to the square root of an E_p value. In a range image, each pixel corresponds to a t_p value. Figure C shows the ground-based system, developed by Lincoln Laboratory's Opto-Radar Group, that was used to take the real-life images shown in this article.

Other measurements can also be made with a laser radar. For example, by using a continuous waveform, Doppler shift can be measured to indicate certain translations or rotations of an object. And, by using the same optical system, passive infrared detectors can be installed to provide an object's thermal image. The thermal image would be pixel-registered with the range and intensity images.

For additional information, the reader should consult recent *Proceedings of SPIE* such as Ref. 1 or 2 and the special issue of *Optical Engineering* on laser radars [3].

References

1. R.J. Becherer, ed., *Laser Radar III, Proc. of SPIE* **999**, *Boston, 6–7 Sept. 1988.*
2. V.G. Roper, ed., *Active Infrared Systems and Technology, Proc. of SPIE* **806,** *The Hague, 2–3 April 1987.*
3. R.J. Becherer, ed., "Special Issue on Laser Radars," *Opt. Eng.*, Jan. 1990.

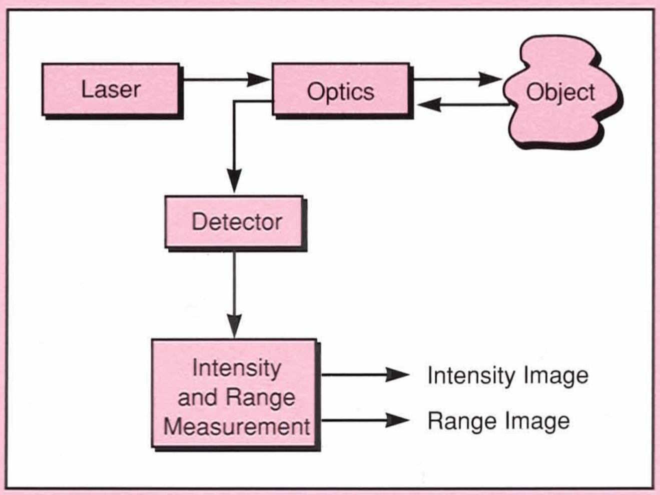


Fig. A—A simplified block diagram of a laser radar.

the ground according to the targets' tactical importance. In spite of this interest, we did the bulk of our initial work with imagery from a ground-based laser radar sensor, because of the availability of such data.

In developing our ATR system, we relied partly on some of the techniques developed over the past 20 years in the field of artificial intelligence, often called machine intelligence. In the following sections, we describe the system we devel-

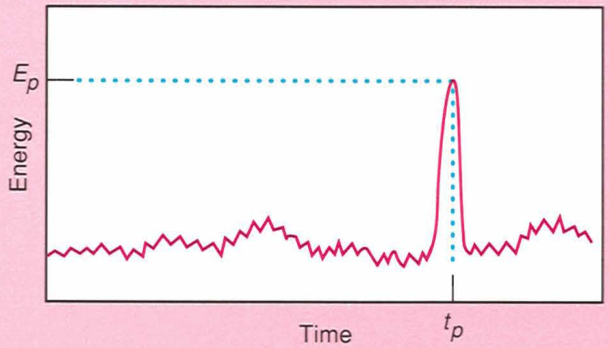


Fig. B—Typical laser radar return signal.

Fig. C—Ground-based laser radar sensor developed by the Opto-Radar Group at Lincoln Laboratory.

oped and discuss its performance on both real-life and synthetic data.

Model-Based Recognition

In developing an experimental ATR system, we chose a *model-based approach* [1] to target recognition. A model-based recognition system generally consists of four major elements, as depicted in Fig. 1.

In the *event-characterization subsystem*, many image-processing and image-analysis

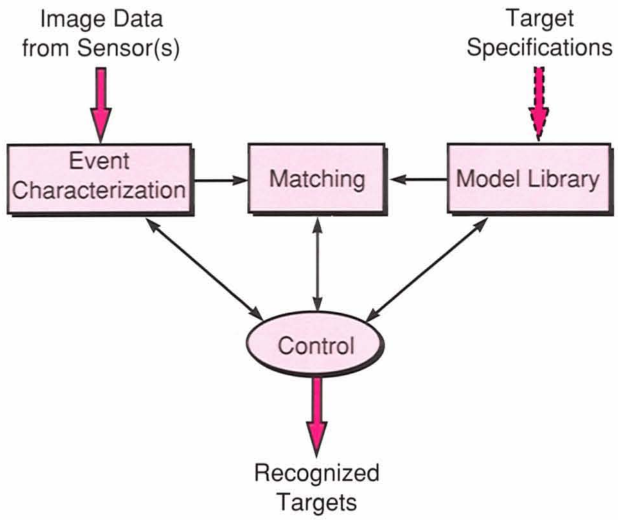


Fig. 1—Block diagram of a general model-based recognition system.

algorithms analyze imagery from sensors to detect and extract information about *events* in the field of view of the sensors. (An event is some irregularity in the sensor data that might or might not indicate the presence of an *object*. An object is some physical entity in a sensor's field of view that might or might not be a target such as a tank. Note that objects are not necessarily targets. For example, an object could turn out to be a boulder.) The subsystem then describes, or characterizes, each event by using a data representation that combines both numeric and symbolic information.

The *model library* contains models that are computer representations of targets and that convey information about the environments in which they exist. Herein lies a great challenge: how to encode this diversity of *a priori* knowledge to match the descriptions generated by the event-characterization subsystem. Arranging this knowledge in a structured and easily accessible form has many advantages. It simplifies the initial design and implementation of the system; it facilitates later alterations of the system, including extensions to handle other target types; and, finally, it makes it easier for the user to understand how the system actually recognizes targets.

Between the event-characterization subsystem and the model library lies the *matching subsystem*, which attempts to match an event detected in the input imagery to one of the target models contained in the model library. The development of a practical matching subsystem is a challenge. A simple approach would be to compare each detected event with each target model. Computationally, however, this approach is generally not practical because the system usually needs to match a large number of events with a large number of models. Consequently, it is desirable to structure the matching operation so that, by first performing a relatively small number of computations, many models can be eliminated at an early stage. We will examine one such approach later.

The final major subsystem is the *control subsystem*. The control subsystem's primary job is to apply computational resources to the recognition problem as efficiently as possible to reduce the overall number of calculations needed to identify targets.

A general mechanism by which computational resources can be conserved is *attention focusing*. For example, in the event-characterization subsystem in which most of the numerical calculations take place, we can allocate our computational resources more efficiently by avoiding the systematic usage of computation-intensive image-processing and image-analysis algorithms across whole images. Instead, we could first apply inexpensive algorithms to the images in order to pick out any areas of potential interest. The areas could then be passed to the control subsystem, which would decide whether to commit more specialized and/or powerful algorithms to the selected areas. To illustrate, very little is gained by applying a wheel-detector algorithm to a part of an image that is found to be a body of water.

Experimental Target-Recognition System

Over the past several years we have developed a model-based ATR system for recognizing tanks, howitzers, and armored personnel carriers (APC) from laser radar images. Our goal was to create an end-to-end system—one that accepts laser radar images as its input and produces a recognition decision as its output. We did not require that each piece of the system be optimal or state of the art. But we did want to have enough pieces in place to comprise a complete system so that we could conduct experiments with different subsystems, algorithms, and parameters to evaluate their impact on the overall system's recognition performance.

We have been working primarily with imagery taken from a ground-based infrared laser radar system developed by the Opto-Radar Group at Lincoln Laboratory [2]. (See the box "Laser Radar Sensors.") A laser radar is an active sensor: it illuminates a scene spot by spot and measures the amount of light reflected by each spot. It also measures how long the light takes to return to the sensor. These two measurements are used to produce intensity and range images, which are shown respectively in Figs. 2 and 3. In Fig. 2, the top row of intensity images shows three views of a tank. The middle row shows three views of a howitzer, and the bottom row three views of an APC. A bright picture element, or pixel, in an intensity image (Fig. 2) indicates a strong reflector of laser radiation; a dark pixel indicates that very little energy was returned to the sensor from the illuminated region of space. Figure 3 shows the corresponding range images, which are pixel-registered with the intensity images. In range images, pixel brightness corresponds to the distance between an object and the sensor. Bright pixels indicate a distant object, dark pixels a near one.

The range images of Fig. 3 contain numerous *dropout* and *outlier* pixels. Dropouts, shown as black pixels in Fig. 3, correspond to those areas in the sensor's field of view which, for whatever reason, reflect very little radiation back to the sensor. The return signals thus remain below the detection threshold. Consequently, the sensor cannot make reliable range estimates for these pixels, and labels them as dropouts. An outlier pixel results when noise causes the sensor to pick the wrong peak in a return signal. Some outliers are visible as white dots within the black area at the bottom of the images in Fig. 3. Dropouts and outliers are collectively called *missing values*.

Our primary data set consists of 40 such pairs of real-life intensity and range images. Each image is 60 pixels high by 128 pixels wide. (The images are zoomed vertically by a factor of two so that the targets appear with the correct aspect ratio.) Each image pair exhibits a tank, a howitzer, or an APC at a range of approximately 700 m. The orientation of the vehicle and, in the case of tanks and howitzers, the turret-to-body rotational angle may vary from one image pair to the next.

The sensor used to acquire our primary data set has a range precision of approximately 6 m. Because of this precision, most pixels corresponding to an object fell into a single range grouping, called a bin, or the pixels straddled the boundary between two adjacent bins.

In addition to our primary data set, we have other sets of images that contain real-life targets at varying ranges against different types of backgrounds. And we are currently examining airborne imagery acquired from the improved-range-precision laser radar sensor of the Opto-Radar Group. We have also augmented our image data set by developing a synthetic-image generator, which allows us to construct artificial scenes containing objects in situations for which we do not have real-life imagery. Our synthetic-image generator also allows us to know exactly the contents of the images we process, and it enables us to vary sensor parameters, such as range precision, angular resolution, and the percentage of outlier pixels in the images.

Real-life imagery, as shown in Figs. 2 and 3, is characterized by coarse range precision and a significant amount of dropout and outlier noise. Furthermore, the objects that appear in real-life imagery are not precisely known; they may be partially occluded, have articulated parts rotated to various orientations, or have missing or extraneous parts. These and other factors all contribute to the complexity of the ATR problem.

Our discussion will focus on range imagery, as opposed to intensity imagery, because our system relies mainly on range data. (The intensity imagery is used only as a cue in the early stages of processing.) The subject of recognizing 3-D

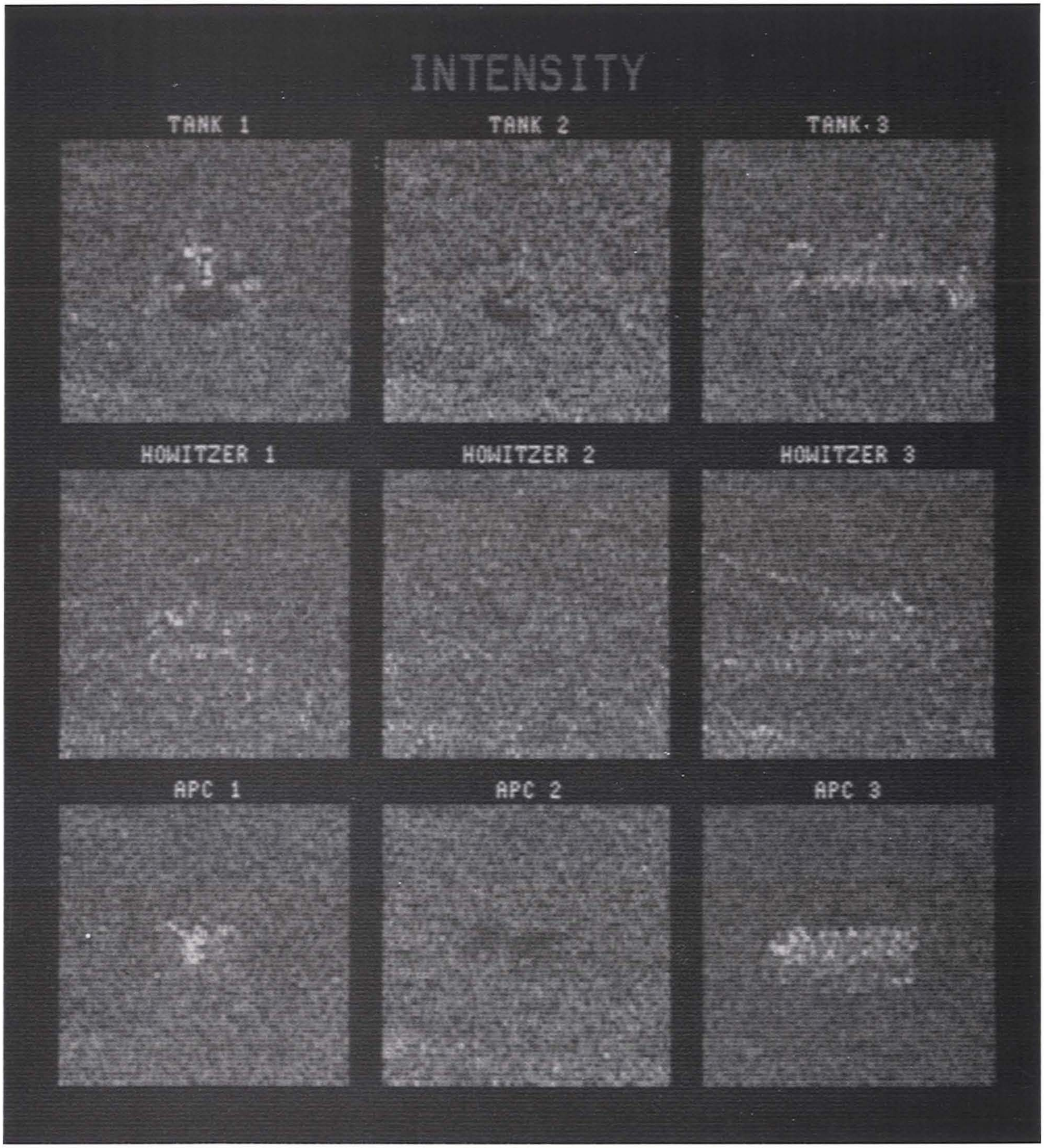


Fig. 2—Intensity images from a laser radar sensor.

targets from range imagery has received much attention in the last few years [3]. However, most published approaches assume virtually ideal conditions: high-precision range images with excellent angular resolution in well-controlled, nearly noise-free environments. Also, the targets sought are generally known precisely and are usually rigid, as opposed to articulated. In

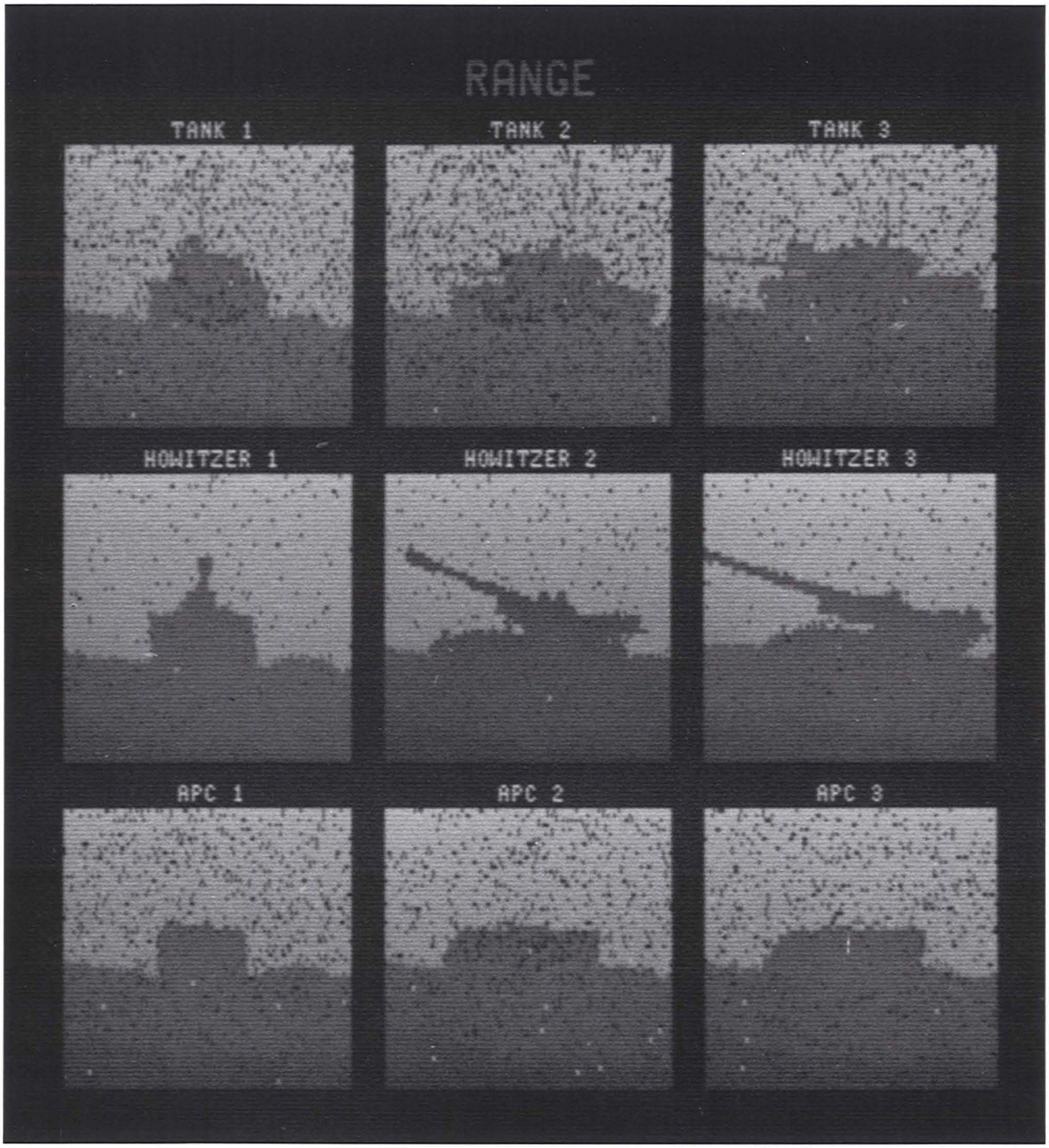


Fig. 3—Range images from a laser radar sensor.

addition, the targets are usually located at short distances, from a fraction of a meter to a few meters away.

Because of the coarse range precision in our ground-based imagery, we need to use 2-D silhouettes to recognize 3-D targets. Thus, most of the published approaches are not well suited to our problem; in fact, the problem of recogniz-

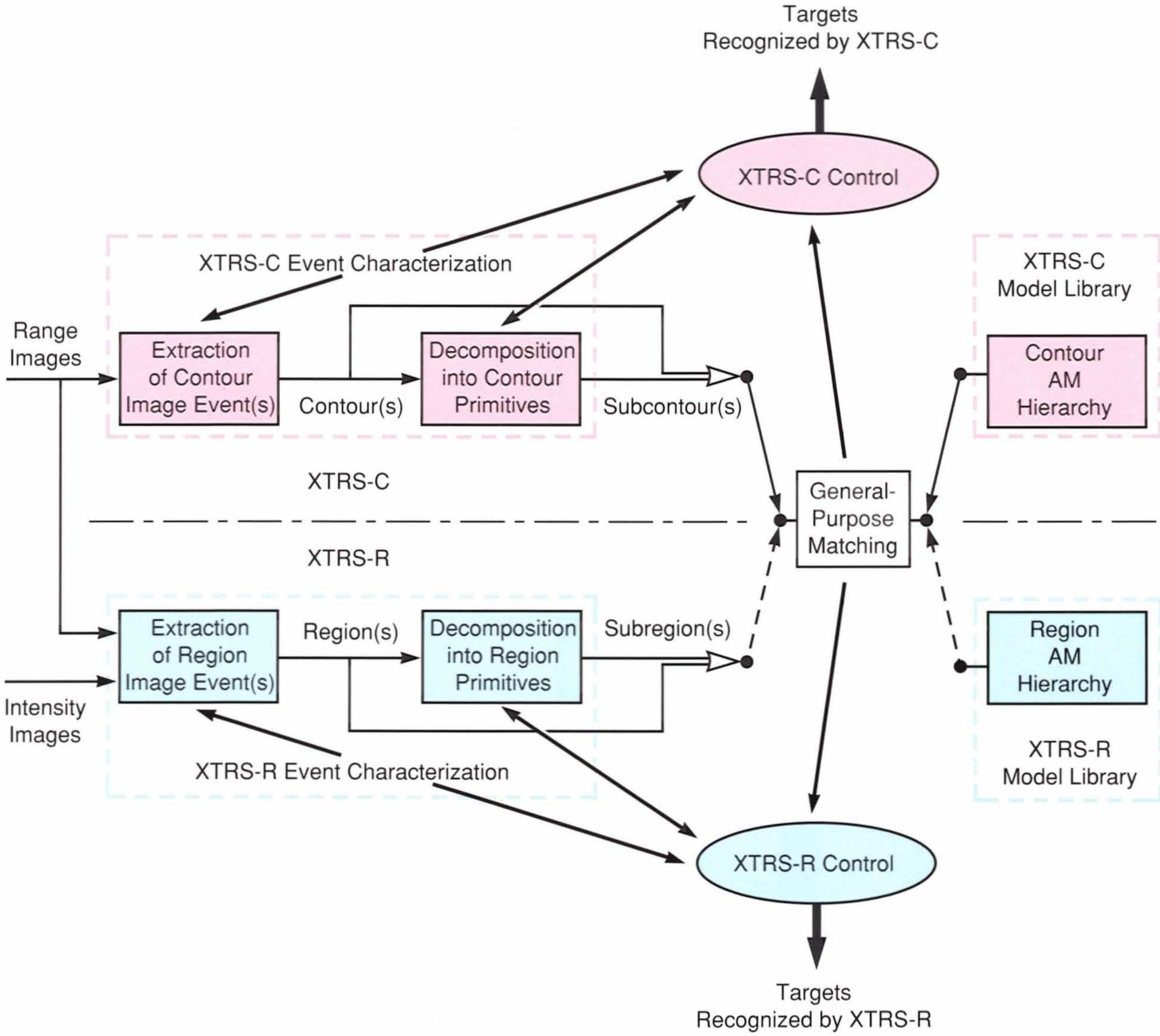


Fig. 4—Block diagram of the experimental target-recognition system (XTRS).

ing 3-D targets from their 2-D silhouettes has not been widely studied. One recent and promising approach is that of P.L. Van Hove [4, 5]. However, his approach requires detailed target models and relatively clean silhouettes, and his system cannot currently deal with articulated targets. The distinguishing feature of our experimental target-recognition system (XTRS) is its ability to recognize possibly articulated, imprecisely known targets even when they are viewed from a variety of vantage points.

The block diagram of XTRS (Fig. 4) reflects the general diagram (Fig. 1) discussed earlier, with one major variation. There are actually two distinct recognition systems within XTRS. Although both systems extract range silhouettes from laser radar imagery and both make recognition decisions automatically, the two systems differ in their processing approaches. The contour-based system (XTRS-C) attempts to extract range discontinuities that correspond to object boundaries, while the region-based system (XTRS-R) attempts to extract constant-range regions. XTRS-C and XTRS-R differ only in their event-characterization subsystems, their model libraries, and, to a lesser extent, their control subsystems; both XTRS-C and XTRS-R use the same general-purpose matching subsystem.

In the following subsections, we will discuss in greater detail each of XTRS's different subsystems.

Event Characterization

Image events [6] are irregularities in the sensor imagery that might indicate the presence of a target. (Of course, in practice not all image events correspond to targets—or even objects, for that matter—and the recognition system must be able to deal with any spurious event to reduce false alarms.) The specific image events that XTRS is interested in are silhouettes occurring in range imagery. XTRS-C looks for silhouette contours and XTRS-R looks for silhouette regions.

In XTRS-C and XTRS-R, the event-characterization subsystem consists of two separate steps (Fig. 4): extraction of events from sensor imagery, and decomposition of those events into basic features called *primitives*. In XTRS-C, the primitives are arc and corner subcontours; in XTRS-R, the primitives are subregions.

The processing steps of XTRS-C's event-characterization subsystem are illustrated in Figs. 5-1 through Figs. 5-8. Figure 5-1 shows an unprocessed range image, the only input currently used by XTRS-C.

Event extraction begins with the identification of all dropouts and outliers (shown in red in Fig. 5-2) and their subsequent replacement with locally reasonable values [7]. A simple gray-scale mathematical-morphology operation [8] removes all remaining isolated black pixels [9, 10]. Figure 5-3 shows the cleaned range image, which is then processed by the use of a difference-of-Gaussian (DOG) edge-detection operator [11]. The resulting *zero crossings* (ZC) of Fig. 5-4 correspond to edges, or borders, separating areas of different range values in the cleaned range image of Fig. 5-3. ZCs corresponding to the largest range discontinuities are selected for further processing (Fig. 5-5). Finally, in highly simplified terms, the desired contour event is obtained by taking the longest of the retained ZCs (shown in blue in Fig. 5-5), splicing together other open ZCs in the vicinity, and attaching closed contours that are nearby, elongated, and properly oriented (shown in yellow in Fig. 5-5). (Details of the heuristic procedure used to splice together ZCs are beyond the scope of this article.) In the particular case considered, our procedure has the effect of lengthening one of the antennas, reattaching the other, and recruiting a large ZC that provides the bottom part of the gun and the front of the tank. Figure 5-6 shows the extracted contour event and the event's range in meters.

Next, to decompose the image event of Fig. 5-6, XTRS-C first computes a polygonal approximation by using U. Ramer's algorithm [12] (Fig. 5-7). The polygonal approximation is edited and subsequently decomposed into arc and corner subcontours by using an approach similar to that of Ref. 13. In Fig. 5-8, convex corners are shown in green and concave corners in red.

The processing steps of XTRS-R's event-characterization subsystem are illustrated in Figs. 6-1 to 6-8. Figure 6-1 shows the unprocessed intensity image used by XTRS-R. The unprocessed range image that constitutes the second input to XTRS-R is identical to the image shown in Fig. 5-1.

Event extraction in XTRS-R begins with cleaning an unprocessed range image in a manner similar to that of XTRS-C (Fig. 6-2). Next, a combined *interest image* (Fig. 6-3) is created by merging three individual interest images: the intensity image of Fig. 6-1, an image that highlights vertical surfaces of limited height, and an image that highlights rod-shaped features. (Interest images are images in which large pixel values will most likely correspond to the presence of a target. For a more precise definition of "interest image" and a detailed explanation of the interest images used in XTRS-R, see Ref. 14.) Range values corresponding to the largest peaks in the combined interest image are then selected. For each peak, the pixels in the cleaned range image that are at the same or similar range as the peak are picked out and grouped into separate regions.

Six such regions are shown in different colors

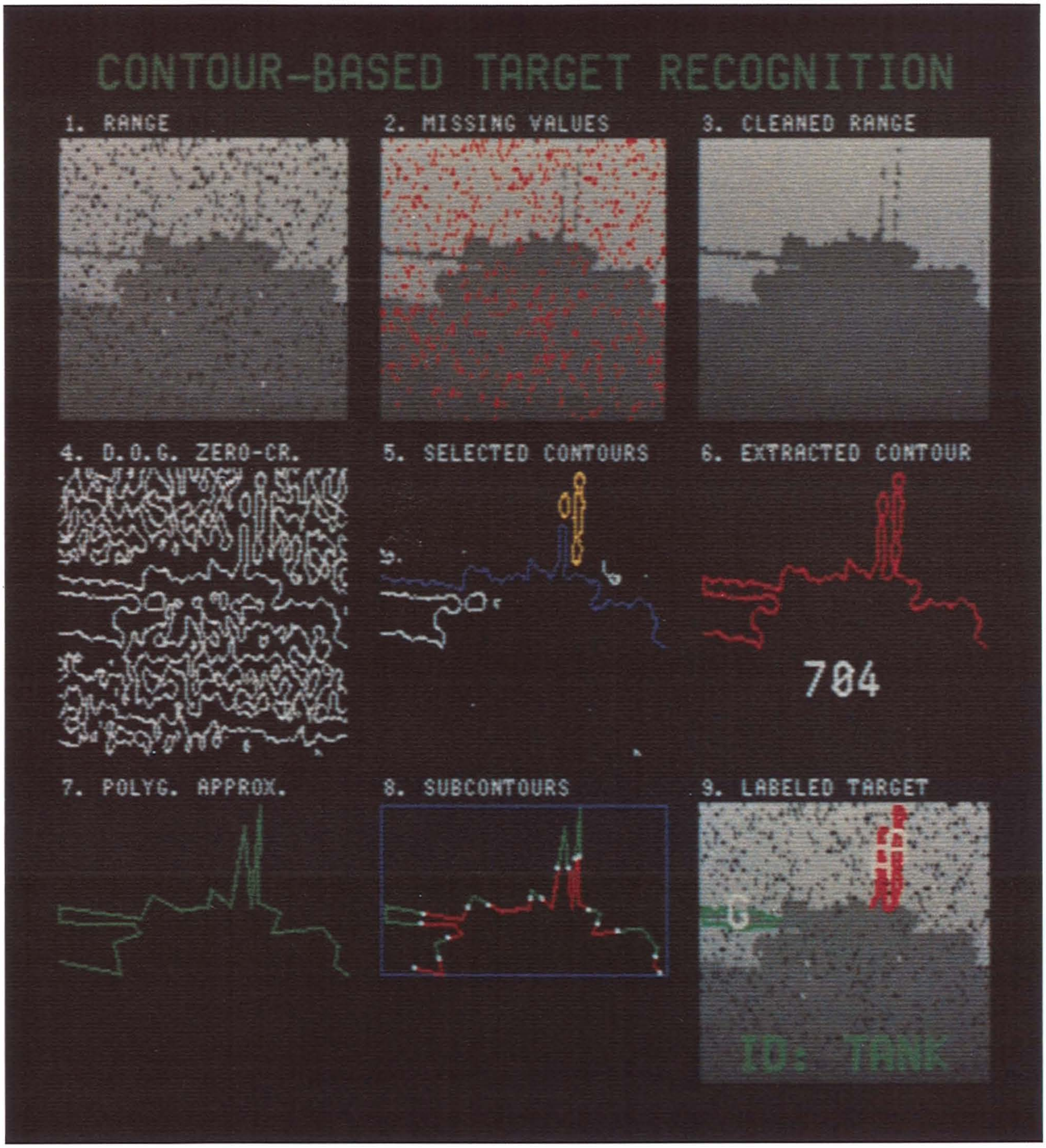


Fig. 5—Processing steps in contour-based target extraction, decomposition, and matching for XTRS-C.

in Fig. 6-4. The region with the highest concentration of interest-image values (shown in magenta in Fig. 6-4) is selected for further analysis. (If that region does not turn out to be a target, the region with the next highest concentration of interest is examined.) The magenta region of Fig. 6-4 is shown in gray in Fig. 6-5. The nongray pixels in Fig. 6-5 are classified as being either

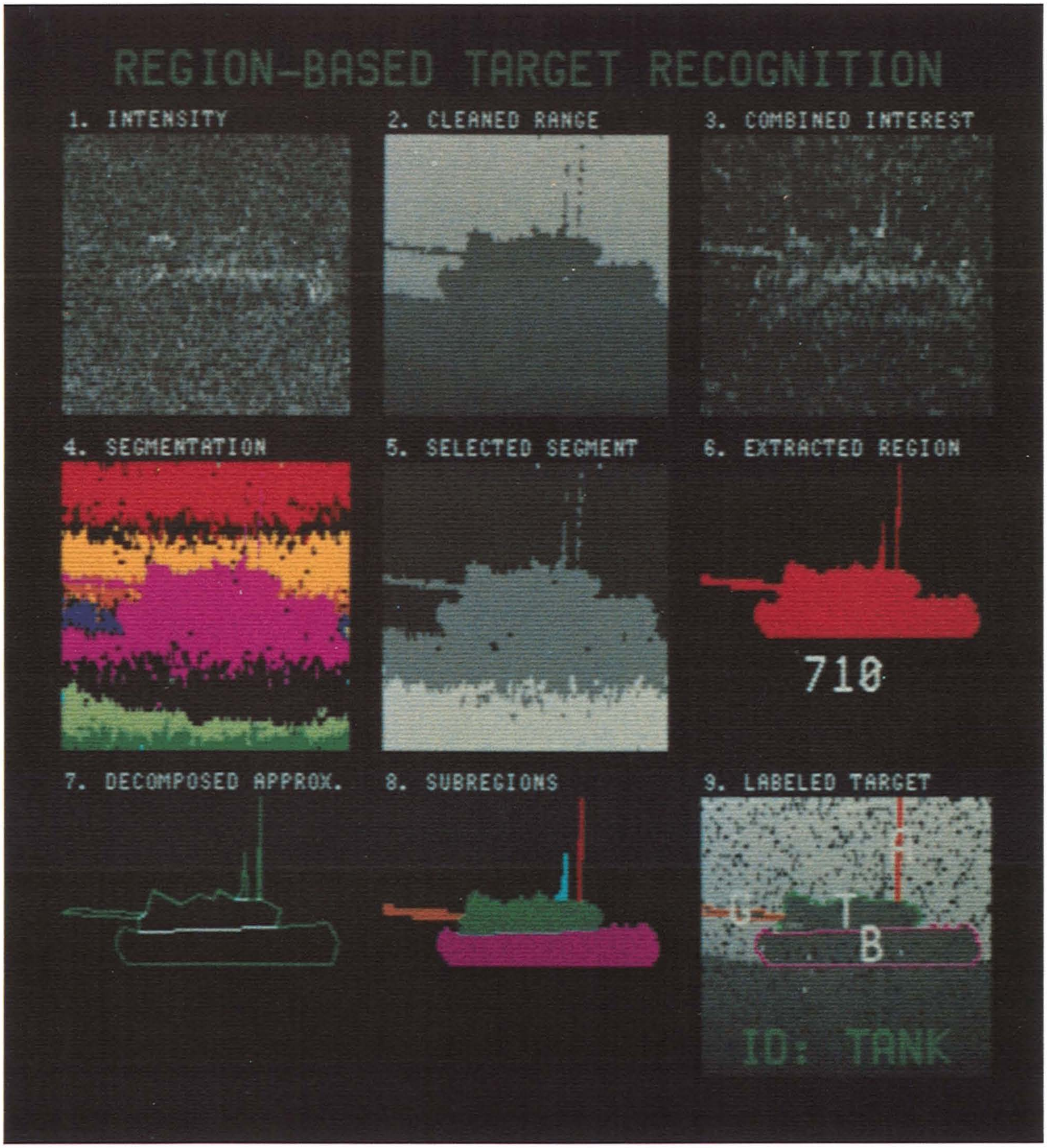


Fig. 6—Processing steps in region-based target extraction, decomposition, and matching for XTRS-R.

closer (white pixels) or farther (black pixels) than the region of interest. The closer (white) pixels are then changed into either gray or black pixels by a nearest-neighbor type of algorithm. Finally, the resulting binary image of gray and black pixels is processed by algorithms that (*a*) remove any holes (those areas appearing as black dots within the gray tank region of Fig. 6-5) in the region of interest; (*b*) estimate the ground line at the range of the region; and (*c*) reconnect any appendages (guns and antennas in Fig. 6-5) to the region's main body by using mathematical-morphology techniques similar to those of Refs. 9 and 10. The result, shown in Fig. 6-6, is the extracted region event and its measured range in meters.

XTRS-R decomposes the image event (Fig. 6-6) by first computing and editing the Ramer polygonal approximation [12] of the event's boundary (Fig. 6-7). The polygonal approximation is then decomposed by heuristically selected baselines (the gray lines in Fig. 6-7). The resulting figure is then mapped back onto the original event in order to decompose the event into subregions (Fig. 6-8). (For a detailed explanation of the XTRS-R event-extraction and event-decomposition procedure, see Ref. 14.)

For each event extracted by XTRS-C or XTRS-R, the output of the event-characterization subsystem consists of symbolic descriptions of the event itself and the event's primitives: arc and corner subcontours for XTRS-C, and subregions for XTRS-R. To facilitate the use of this information by the matching subsystem, the primitives and their descriptions are organized into *attributed relational graphs* (ARG) [15], which are data structures similar to semantic networks [16]. An ARG is a graph whose nodes contain symbolic and numeric attributes describing the primitives and whose links between nodes indicate relationships between the primitives.

Model Library

The model libraries for XTRS-C and XTRS-R contain *appearance models* (AM) that store information about each type of target that the systems need to be able to identify. An AM is not a full 3-D model of a target; rather, it is a computer representation of how the target would appear in a 2-D image produced by a given sensor's imaging modality. A single AM describes the appearance of a target in a variety of aspects and states of articulation. AMs are also designed to allow for missing and extraneous parts as well as for some degree of target occlusion. (The concept of an AM was first introduced by P.G. Selfridge [17], but our AM definition, implementation, and usage differ from his. A detailed explanation of the concept and construction of AMs as defined in this article is given in Ref. 18.)

In our present system, each AM describes a target in terms of its parts and the relations between them. (The box "A Simple AM Example" gives a detailed explanation of the AM for a simple target.) In creating AMs for our library, we describe targets in terms of whatever parts we believe would be easily identifiable in the imagery. For example, we describe a tank in terms of its body, turret, gun, and antenna.

To organize information about each AM, we use data structures, similar to semantic networks [16], that describe targets in terms of the sizes and shapes of their parts and how the parts relate. We define each AM as a separate *part hierarchy* in which the hierarchy's root node represents a target (see Figs. 7 and 8). The direct descendant nodes of each root node in a part hierarchy represent the parts that comprise the target represented by that root node. In addition, parts may be defined in terms of their own parts so that a part hierarchy may contain several levels of nodes. A part hierarchy's lowest-level nodes, called *terminal nodes*, correspond to a target's *atomic parts*. Terminal nodes contain descriptions, called *property sets*, that can be compared with the descriptions of primitives that the event-characterization subsystem produces.

The part hierarchies also contain information regarding the physical relationships between the different parts of a target. For example, in Fig. 8 the tank AM specifies that a tank's antenna must always be above its turret. These constraints and the parts of a target's AM define the expected possible appearances of that target in a 2-D laser-radar range image.

Our system's complement of AM part hierarchies is organized into a larger hierarchy that acts as a model library for XTRS. The overall hierarchy, called an AM hierarchy, comprises XTRS's knowledge about the kinds of targets that XTRS is able to identify. It consists of a *specialization hierarchy* and the part hierarchies of the different AMs (Figs. 7 and 8). The specialization hierarchy is a hierarchy of categories; i.e., the descendants of a given node represent subcategories of the category represented by that node. For example, in Figs. 7 and 8, the category "vehicle" includes the subcategories "tank," "howitzer," and "APC."

As discussed in the previous subsection, the kinds of descriptions of events and primitives generated in XTRS-C's event-characterization subsystem differ from those derived in XTRS-R's subsystem. AM hierarchies need to be defined in terms of those descriptions in order to permit the matching process. Consequently, we had to develop two separate AM hierarchies: a contour-based hierarchy for XTRS-C, and a region-based one for XTRS-R.

Figure 7 depicts the AM hierarchy for XTRS-C. Three types of vehicles (*V*) are shown: a tank (*T*), a howitzer (*H*), and an APC (*A*). XTRS-C recognizes the type of vehicle present in a range image solely from the presence, or absence, of characteristic appendages such as gun barrels and antennas. This approach was used because the contour-based event-characterization subsystem can easily extract such appendagelike parts from range imagery. Also, in the case of our primary data set, appendagelike parts are very effective in distinguishing different vehicles from one another. For example, if a vehicle has a gunlike appendage, we know from our AM hierarchy in Fig. 7 that the vehicle cannot be type *A*.

In Fig. 7, the terminal nodes describe different subcontours in terms of attributes such as concavity/convexity, length, width, and orientation. One constraint between the antenna nodes indicates that the antennas' axes of least inertia must be relatively near to one another. The other constraint, "left of," is used to distinguish the antennas.

The AM hierarchy for XTRS-R is shown in Fig. 8. XTRS's hierarchy is defined in terms of the region-based symbolic descriptions generated by XTRS's event-characterization subsystem. Thus, the part nodes describe different subregions in terms of attributes such as area, length, width, and orientation. The constraints between nodes (e.g., "above," "beside," or "proportional to") are defined in terms of the possible relations among the subregions. In this particular hierarchy, the descriptions contained within each of the *V*, *T*, *H*, and *A* category nodes include the expected size of the silhouette areas, in square meters, for targets belonging to each of those particular vehicle categories. (In XTRS-C, category nodes do not currently contain such property sets.)

Note that most of the parts in both types of AM hierarchies we expect to be present in the targets. However, through the use of the "shouldn't be present" condition (Figs. 7 and 8), it is possible to include, within the description, parts that should not be present in a given target. For example, in our primary data set, APCs do not have antennas.

Matching

Conceptually, the act of recognition in our narrow context consists of determining which, if any, of the AMs match the event descriptions extracted from the laser radar imagery. (A detailed explanation of the matching algorithm and the procedures that implement the algorithm is contained in Ref. 18.) The recognition procedure implemented in the matching subsystem is intended to be general purpose. Thus, the same matching subsystem should be able to handle different event-characterization subsystems and different AM hierarchies.

The input to the matching subsystem consists of a symbolic description, either contour- or region-based, representing the image event of interest; the corresponding symbolic descriptions of the event's primitives; and an appropriate AM hierarchy, i.e., either contour- or region-based. With this data, the matching subsystem deduces the identity of the target in the AM hierarchy that best matches the image event of interest.

The matching subsystem consists of three steps (Fig. 9). In the first step, *symbolic matching*, we determine the best possible pairings of primitive features to atomic parts for each AM. In the *belief-computation* step, we compute the

A Simple AM Example

Consider the simple target shown in Fig. A-1. Its silhouette will take the form of Fig. A-2 when viewed from any one of a number of directions. Figure A-3 depicts an appearance model (AM) that describes the expected appearances of the target. The AM is appropriate only for those types of images which lend themselves to the extraction of 2-D target silhouettes. The model assumes that the target will be viewed only from points at roughly the same elevation as the center of the target and that the target is roughly horizontal when viewed. For these aspects, the silhouette region of Fig. A-2 should always consist of two subregions. The corresponding AM is a part hierarchy in which the *object* is defined to be comprised of the two parts *body* and *appendage*. Each of these parts is described by a set of *property functions*, collectively termed a *property set*. A property function is a fuzzy predicate [1] defined over the values of some part's attribute(s). A property function, after receiving a given attribute, will return a value in the closed range [0, 1] indicating the degree to which the attribute satisfies the particular property function.

For example, consider the property function *f*, which concerns the expected length of *body*. As shown in Fig. A-3, this function returns a value of 1.0 for regions whose length is between B and $\sqrt{C^2 + B^2}$, a value between 0.0 and 1.0 for regions whose length lies just outside this interval, and 0.0 for regions of any other length. The definition of this function reflects that the projected length of the body is expected to vary between B and $\sqrt{C^2 + B^2}$ for the assumed range of viewpoints. The rising and falling ramps on either side of this expected length interval allow for variations in length measurements due to sources of error such as image noise, inaccurate silhouette extraction, and inaccurate computation of region length. In a similar fashion, property functions are also defined for body height and for appendage diameter, height, and major-axis orientation (Fig. A-3).

In addition to descriptions of the target's body and appendage, the model includes the constraint *above* on the spatial relationship between the two parts. This constraint is defined by a fuzzy predicate [1] defined over values of *g*, which is the signed distance between the appendage region's apparent center of gravity and the top of the body region.

References

1. Y. Ohta, *Knowledge-Based Interpretation of Outdoor Natural Color Scenes* (Pitman Publishing, Boston, 1985).

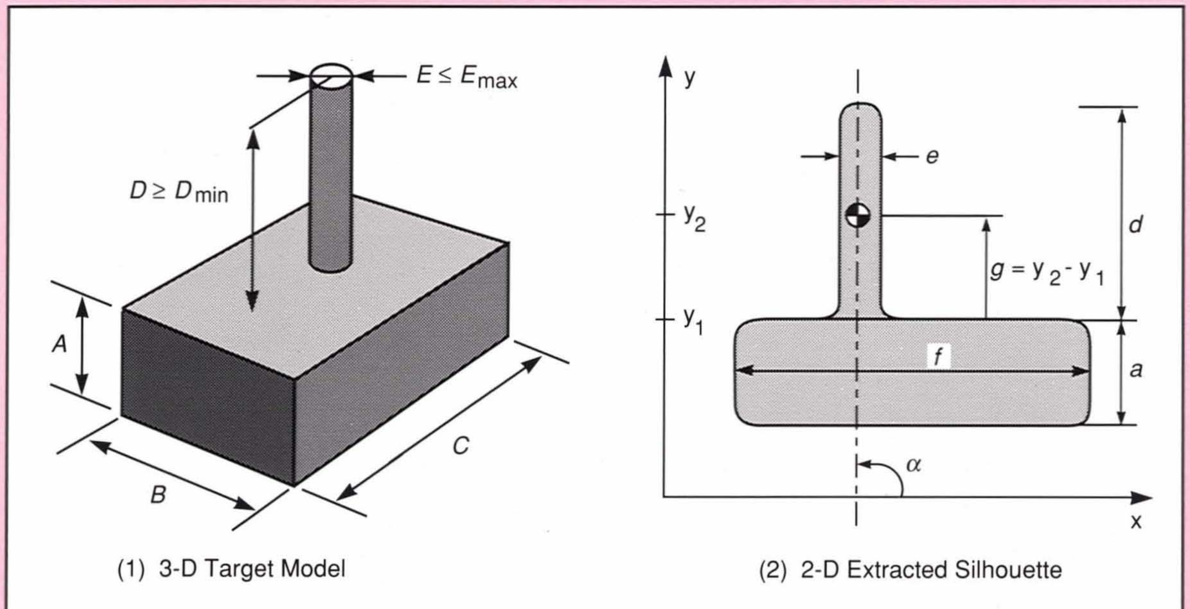
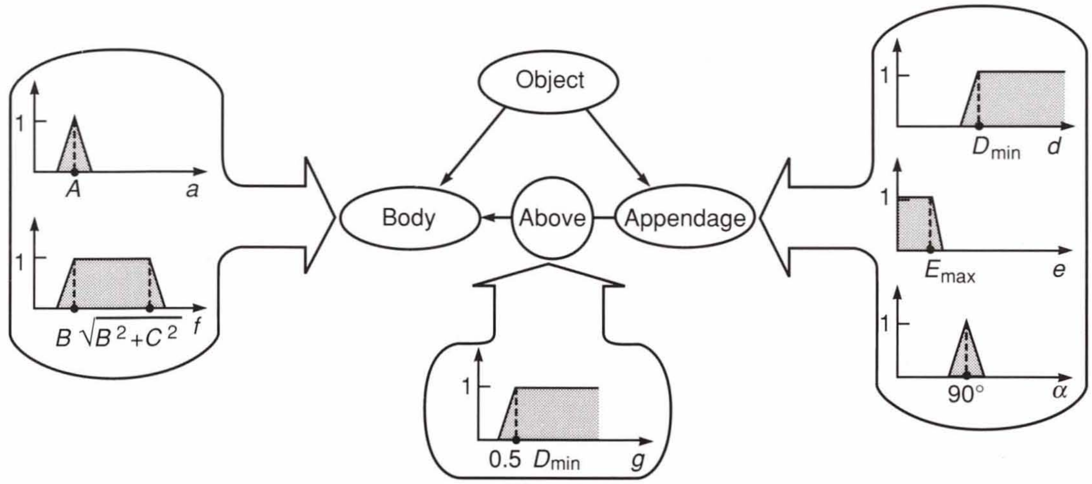


Fig. A—A simple target, its silhouette, and its appearance model.



(3) Appearance Model

belief we have in each of the different hypotheses H_t and H_0, where H_t is the hypothesis "the image event under consideration corresponds to the target t of the AM hierarchy" and H_0 is the hypothesis "the image event being considered does not correspond to any of the targets described in the AM hierarchy." (In the current version of XTRS, t is either a tank, a howitzer, or an APC.) In the matching subsystem's final step, *decision*, we decide the most likely target identity of the image event of interest.

Symbolic matching consists of two phases: *pruning* and *AM matching*. The pruning phase eliminates whole categories of AMs from further consideration on the basis of an image event's description. Those AMs which survive the pruning are called *active AMs*.

After the pruning is completed, the AM-matching phase is then performed independently on each of the active AMs. For a given active AM, distinct primitives are arbitrarily paired with each of the AM's terminal nodes. Note that some primitives might remain unused and some nodes might remain unpaired. A corresponding *degree of matching*—a number between zero and one that indicates how closely a primitive description matches the property set of a particular terminal node—is computed for each of the pairings. After the terminal nodes are paired with the primitives, degrees of matching are then computed for all of the higher-level nodes and for all of the constraints in the given AM. These degrees of matching are computed recursively, from the terminal nodes up to and including the AM's root node. The root node's degree of matching indicates how well, for a particular combination of primitive-to-atomic-part pairings, an image event matches a given AM.

All other combinations of primitive-to-atomic-part pairings are examined in turn, and corresponding degrees of matching for the same root node are computed. The combination of pairings with the highest degree of matching is then selected, as it represents the best possible assignment of primitives to atomic parts for that particular AM.

This sequence of operations is repeated for every active AM. Thus the end result, for each AM, is the best possible combination of pairings and the associated degrees of matching for each

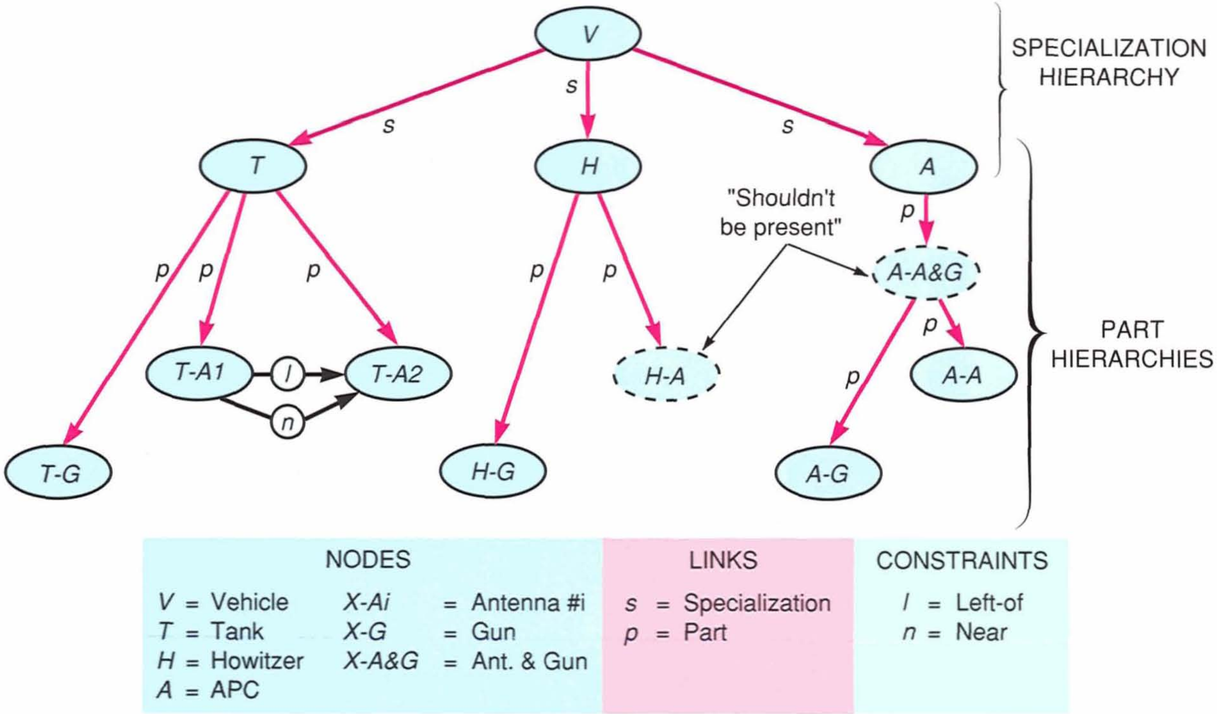


Fig. 7—The appearance model (AM) hierarchy for XTRS-C.

of the AM's parts and constraints. The pruning and AM-matching phases of the symbolic-matching step are illustrated in the box "Symbolic Matching: An Example."

In the *belief-computation* step, we use only the degrees of matching associated with the direct-descendant nodes of each active root node and the constraints that might exist between those nodes. In other words, the degrees of matching associated with all other nodes and constraints are no longer used. A high degree of matching for either a direct-descendant part of target t or a constraint between direct-descendant parts of t is used to increase the belief in hypothesis H_t. Conversely, a low degree of matching is used to decrease the belief in H_t or, in our implementation, to increase the belief in the null hypothesis H_0, and all active hypotheses other than H_t.

To combine the applicable degrees of matching for the different hypotheses, we use the method provided by the Dempster-Shafer theory of evidence [19–21]. (See "Appendix: Elements of the Dempster-Shafer Theory of Evidence" for a brief tutorial on the method.) When we apply the Dempster-Shafer method, each active target hypothesis receives a corresponding *degree of belief*, i.e., a number, ranging from zero to one, that indicates the likelihood that an image event is a particular target. The degree of belief for those hypotheses rejected in the pruning stage is necessarily zero.

A *decision* rule is now needed to translate this set of degrees of belief into a single recognition decision. Currently, XTRS uses a very simple rule: The hypothesis with the largest degree of belief is chosen. However, in a more sophisticated system a more complex decision rule might be used. For example, it might make sense to choose the hypothesis with the largest degree of belief *only* if that hypothesis' degree of belief exceeds its nearest competitors by a wide margin. Thus, if two hypotheses have nearly equal degrees of belief, neither hypothesis will be eliminated. Additional processing can then be used to distinguish between the two hypotheses.

Once the control subsystem has accepted the matching subsystem's decision, the matching results are passed outside the system (Fig. 4). The results consist of the most likely target identity, along with the associated degree of belief, and the corresponding assignments of primitives to atomic parts, along with the related degrees of matching. Examples of target identifications and labeled parts are shown in Figs. 5-9 and 6-9. In the figures, *B* stands for body, *T* for turret, *G* for gun, and *A* for antenna.

Control

The control subsystem's primary responsibility is to provide the overall system with flexibility. Ultimately, the control subsystem should embody a variety of recognition strategies along with the means of selecting the appropriate strategy for a given situation. This topic requires much more work and is currently one of our research interests. We have begun by investigating the use of high-level feedback to improve XTRS's recognition performance.

A common means of achieving flexibility is through the use of thresholds and adjustable parameters that adapt specific functions to either specific image conditions or target classes. For example, the tolerance that the subsystem uses to create a polygonal approximation of an object's silhouette contour needs to vary with respect to the distance between the object and sensor.

Figure 10 illustrates XTRS's control structure, which contains three main processing modules and four distinct feedback modules. Note that each main processing module has an associated *local-feedback* module that can initiate short-loop feedback. These feedback modules can correct algorithm failures. For ex-

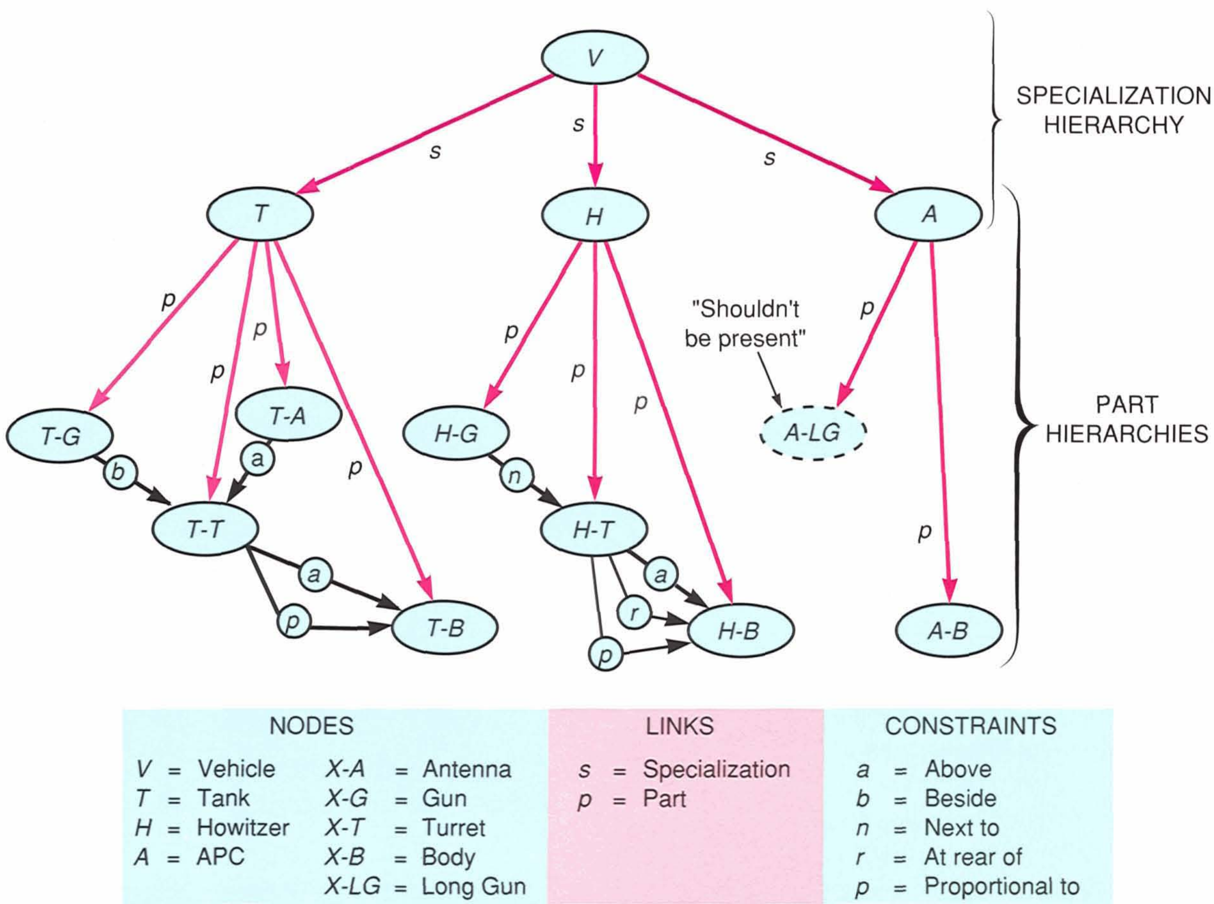


Fig. 8—The appearance model (AM) hierarchy for XTRS-R.

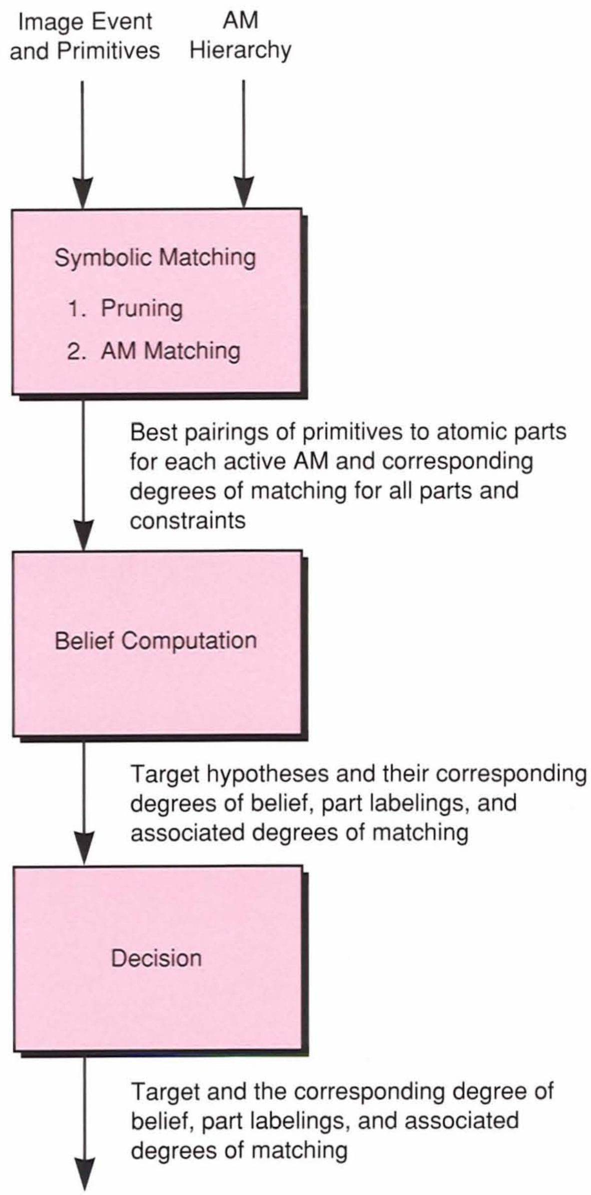


Fig. 9—Processing steps in the matching subsystem of XTRS.

ample, during the decomposition of an image event the failure to find an acceptable polygonal approximation of a silhouette usually results in an increase of the tolerance parameter followed by the subsystem's repeated attempt to decompose the silhouette. Alternatively, intermediate results might indicate that the decomposition module is working on a dead end, which would then result in XTRS rejecting the particular image event under investigation.

If no problems are detected, control is passed on to the next processing module. Local feedback modules also have the option of deciding that the most appropriate response is outside the main processing module associated with that particular local-feedback module. In such a case, control is passed to the *global-feedback* module. Here, the decision is made whether to quit altogether, to return to the extraction module to pick another image event, or, if the current image event is judged salvageable, to change one of the parameters and return to any of the three main processing modules. To avoid the possibility of infinite loops developing in the system, we use counters in each feedback module.

Although we have already implemented the complete feedback system, we are still gaining experience in using it effectively.

Results

Results are summarized for three examples in Figs. 11 through 13. In each case, the top left image is an ideogram depicting an overhead view of an object. The ideogram shows the object's orientation and simplified shape, corresponding to the intensity and range images depicted in the top middle and top right images. Note that the ideogram is oriented so that the laser radar looks toward the object along a viewing direction running from the bottom to the top of the ideogram. Subfigures C1 through C3, respectively, illustrate silhouette extraction, decomposition, and matching for XTRS-C; subfigures R1 through R3 illustrate the same steps for XTRS-R. In both cases, the matching subfigure shows the target type and the labeled target parts. Note that the entire processing sequence, from the input images to the recognized and labeled targets, is executed automatically by the system, which uses a single set of processing parameters. Also note that contour- and region-based processing are completely independent of each other. (However, as mentioned earlier, similar image-cleaning algorithms and the same matching subsystem are used in both cases.)

Figure 11 shows a tank at an oblique angle to the sensor. Both XTRS-C and XTRS-R correctly recognized this target despite failing to extract a perfect silhouette. Because of noise in the range image, XTRS-C did not extract all of the gun barrel and XTRS-R failed to extract one of the antennas. (The XTRS-R did not recognize the other antenna because the corresponding region was longer than allowed.) Despite these small problems, both systems acquired enough information to arrive at a correct decision. This example suggests that a combination of results from both XTRS-C and XTRS-R might lead to correct labeling of more parts.

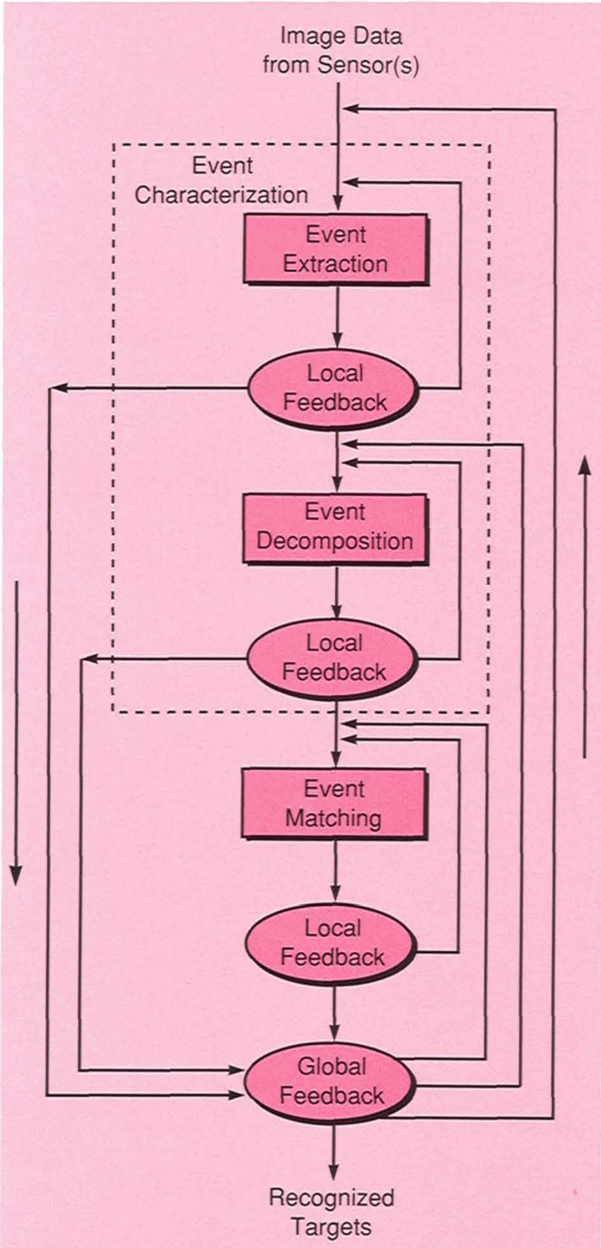


Fig. 10—A flow chart of the control subsystem of XTRS.

A howitzer facing the sensor is shown in Fig. 12. Because the gun is elevated enough, it is visible above the howitzer's body. Both XTRS-C and XTRS-R correctly recognized the target and correctly labeled the gun despite the unusual geometry. The AMs for both systems contained enough information to indicate that the gun could appear in this way. In both XTRS-C and XTRS-R, the gun was distinguished from an antenna on the basis of the appendage's width.

Figure 13 shows an APC at a 90° angle to the sensor's line of sight. Both systems correctly identified the vehicle primarily on the basis of its size and lack of other distinguishing features.

We used XTRS on 40 scenes similar to those of Figs. 11 to 13. The scenes were views from various aspects of vehicles of types *T, H,* and *A*. For vehicles that were articulated, i.e., types *T* and *H*, the scenes also included the vehicles with a variety of turret-to-body rotational angles. XTRS-C and XTRS-R each attained an overall recognition rate of 100% on this limited data set.

The reader should interpret the 100% recognition in light of the fact that the 40 test images were relatively easy to handle. In the 40 images the targets were isolated, they filled a large portion of the frame, noise was manageable, and target-to-background contrast was good. In practice, however, images containing considerable noise, distant objects, object occlusions, and poor contrast between object and background will often be encountered. As a result, one of the consistent difficulties in constructing an ATR system is to ensure robust system performance over a wide spectrum of viewing conditions.

Figures 14 and 15 show examples of more complex cases. Figure 14 shows a howitzer perpendicular to the sensor's line of sight at a range of about 1 km. Because of the depression angle at which this target is viewed, the background is relatively close to the target. The resulting low contrast makes event extraction and decompo-

Symbolic Matching: An Example

As discussed in the section "Matching," symbolic matching consists of two steps: pruning and AM matching. AM matching itself embodies two functions: part matching and constraint evaluation. Fig. A-1 shows a region event (i.e., a target's silhouette) extracted from the laser radar imagery and decomposed into subregion primitives. Both the event and the primitives are numerically and symbolically described in terms of attributes such as size, shape, orientation, and location. The description of the primitives also includes spatial relations between the primitives.

The matching of the event to the AM hierarchy of Fig. A-2 begins with the *pruning* phase. Pruning attempts to remove from further consideration those target hypotheses which do not match well with the gross characteristics of the extracted event or, in this case, the silhouette. We begin with the root node of the AM hierarchy, labeled *V* for vehicle (Fig. A-2). Since the size of the extracted silhouette is consistent with the property set stored in the *V* node, we drop to the next level of detail, which in this case contains the target hypothesis nodes

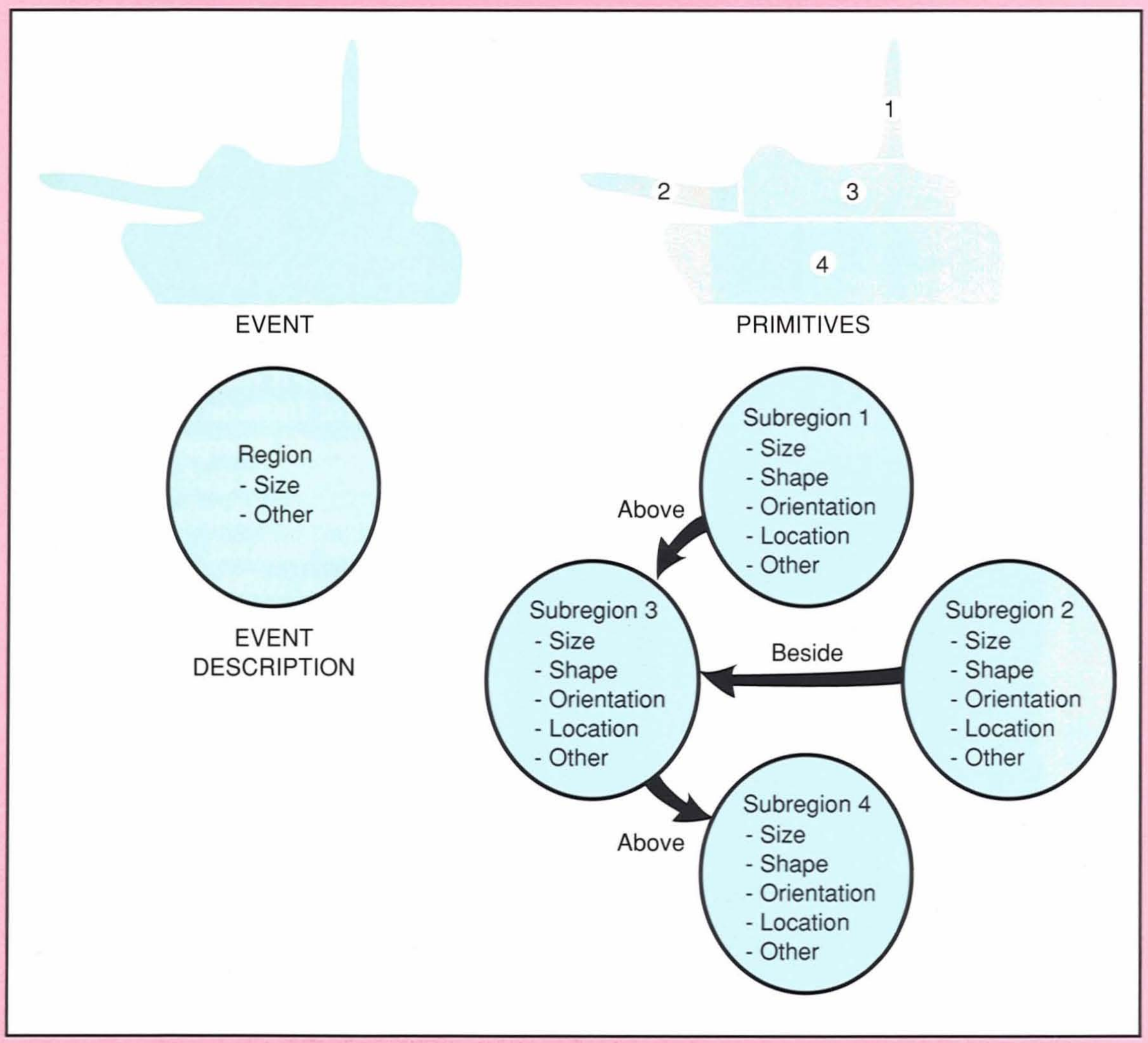


Fig. A—Event characterization and symbolic matching. (1) Region event and primitives.

themselves: *T* for tank, *H* for howitzer, and *A* for armored personnel carrier.

The silhouette characterization is compared to the model information stored in the *T*, *H*, and *A* nodes. In this case, the gross silhouette satisfies the property sets in the nodes *T* and *H* but not in node *A*. Thus, *T* and *H* become the active hypotheses; *A* is pruned from the hypothesis tree and is no longer considered.

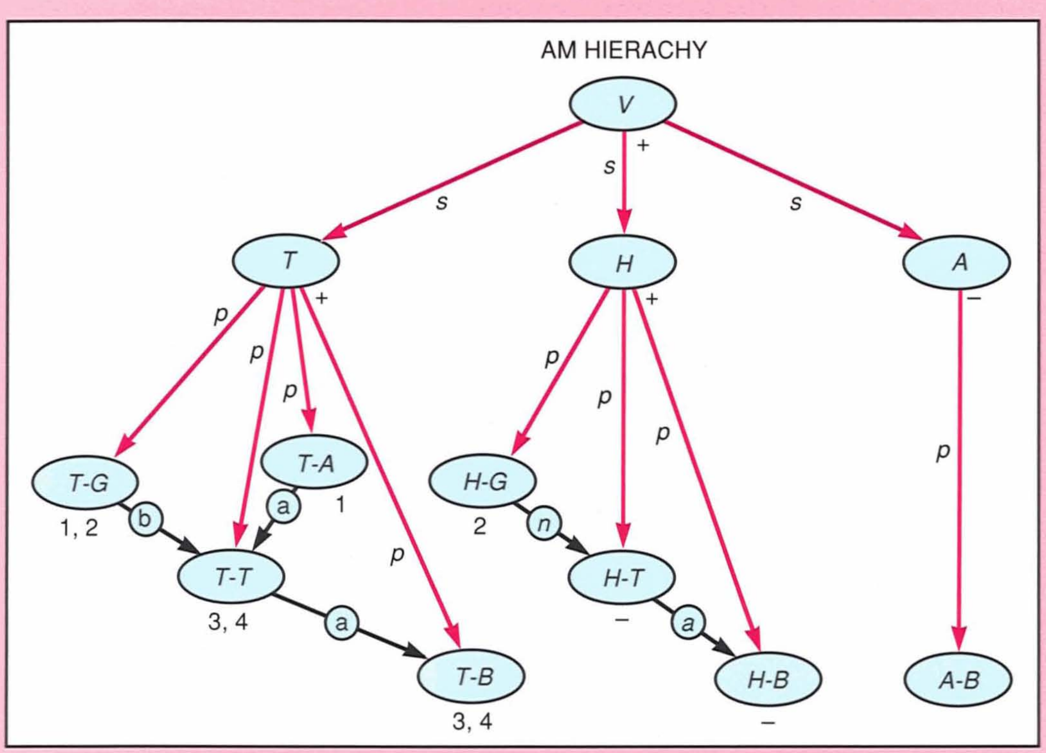
In the second phase, *AM matching*, the four subregions resulting from the silhouette decomposition are matched to the various parts contained in the AMs of the active hypotheses. The tank AM, for example, contains the following parts: gun (*T-G*), antenna (*T-A*), turret (*T-T*), and body (*T-B*). The tank AM also contains different spatial relationships between the different parts: *a* for "above" and *b* for "beside." (The AM hierarchy of Fig. A-2 also contains the spatial relationship *n* for "next to." "Beside" means either "to the left of" or "to the right of." "Next to" includes the categories "above" and "beside." In the figure, *s* stands for "specialization link" and *p* stands for "part link.")

Using the property sets of the various parts, we can match subregion 1 to the *T-G* or the *T-A*, subregion 2 to the *T-G* or the *H-G*, subregion 3 to the *T-T* or the *T-B*, and subregion 4 to the *T-T* or the *T-B*. (It is possible for subregion 1 to correspond to a target's gun if the gun has been greatly elevated and oriented directly toward or away from the sensor. Figure 12 shows how a target's gun could be mistaken for an antenna.) The property sets for subregions 3 and 4 do not satisfy the properties for the howitzer turret or body.

Since the *T-T* must be above the *T-B*, the assignment of subregion 3 to the *T-T* and subregion 4 to the *T-B* is the only choice consistent with this constraint. Finally, matching subregion 1 to part *T-A* and subregion 2 to part *T-G* is assumed to give a higher *degree of matching* for the tank hypothesis than the alternative of matching subregion 1 to *T-G* and leaving subregion 2 unmatched. For the tank hypothesis, then, the assignment of subregions to model parts that gives the best possible match is

$1 \rightarrow T\text{-}A,\ 2 \rightarrow T\text{-}G,\ 3 \rightarrow T\text{-}T,$ and $4 \rightarrow T\text{-}B.$

For the howitzer hypothesis, the best possible match is the assignment of subregion 2 to the *H-G*; no other subregions satisfy the properties of the other howitzer parts.



(2) AM hierarchy (simplified version of Fig. 8).

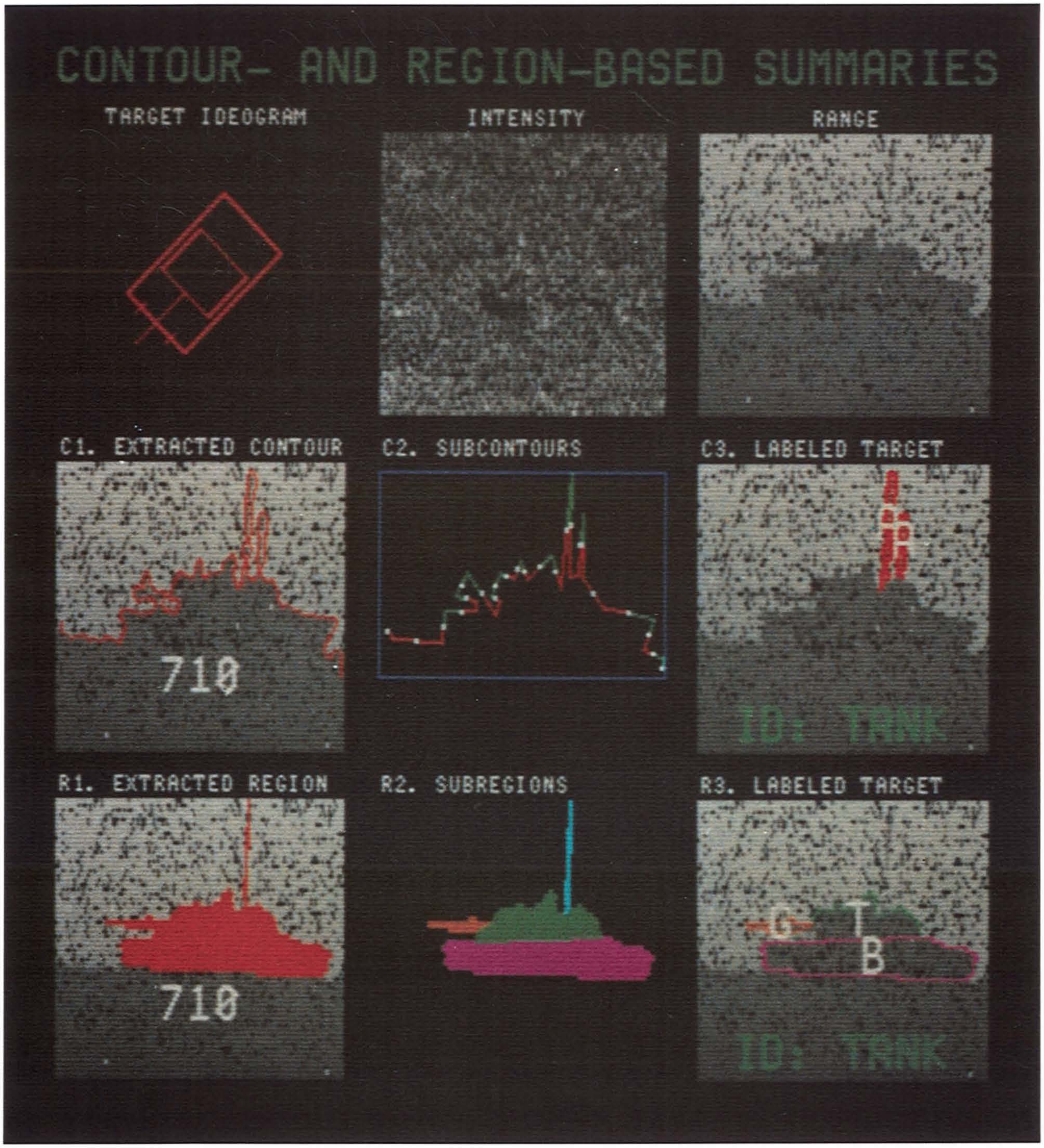


Fig. 11—Recognition example of a tank.

sition difficult. However, both systems were able to extract appropriate silhouettes, and both systems correctly recognized the howitzer. Figure 15 shows a target at a range of about 1.3 km. In addition to the target, the image contains utility poles and general foliage clutter. Nonetheless, XTRS-R was able to extract the correct region and identify it as a tank on the basis of the size of the vehicle's turret and body subregions. XTRS-R handled this particular example suc-

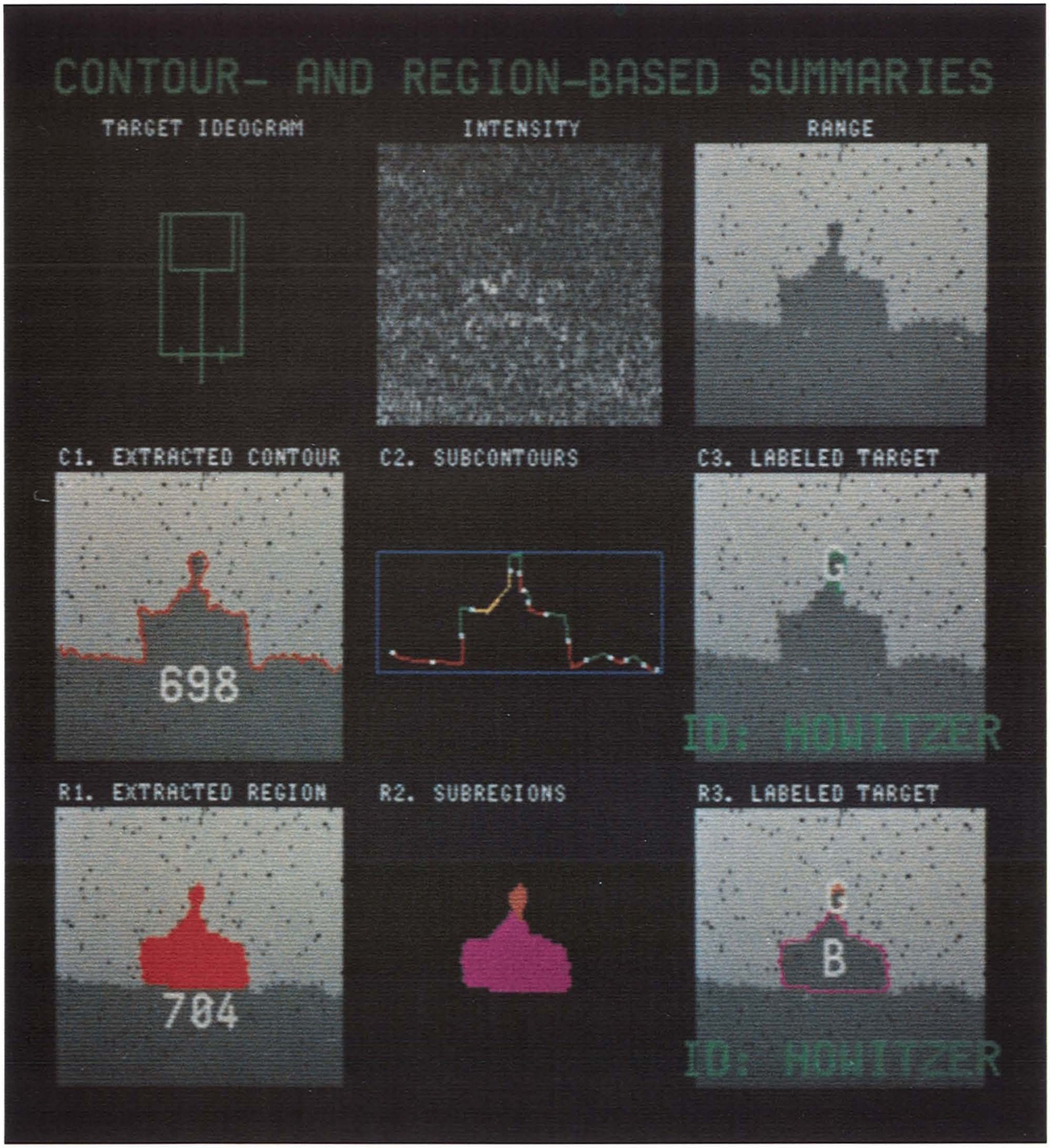


Fig. 12—Recognition example of a howitzer.

cessfully, but, in such complicated imagery, the system often has had difficulty extracting the correct target region. This area clearly calls for further work.

One way of acquiring a data set with additional targets under a greater variety of conditions is to generate images synthetically. We have developed a synthetic-image-generation system that uses 3-D–target geometry to create images in which each pixel contains range and angle-of-

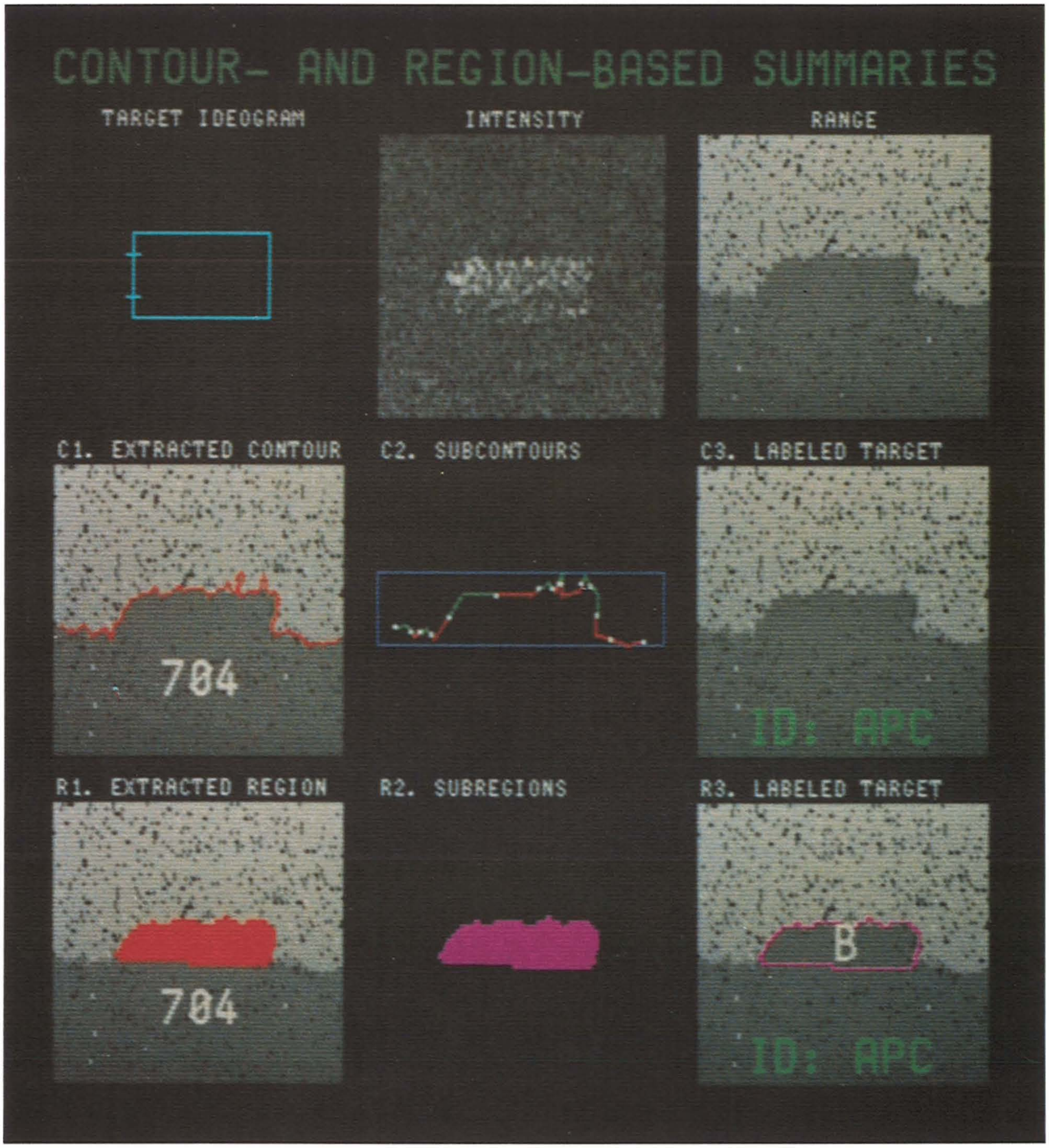


Fig. 13—Recognition example of an armored personnel carrier (APC).

incidence information. The images are then degraded by appropriate types of noise to obtain realistic range and intensity images. This capability allows us to generate data sets to study the performance of XTRS as a function of variables such as range to target. Figure 16 shows synthetic intensity and range images for a tank in a configuration meant to mimic Fig. 11. Figure 17 shows synthetic images of a tank viewed from 1 km away at a 5° depression angle. Note that

Fig. 14—Recognition example for an image with low range contrast.

other objects (in Fig. 17, a telephone pole) can be merged with targets in an image. Figures 16 and 17 give a qualitative sense of how realistic the synthetic imagery can be. The target model used to generate Figs. 16 and 17 is one of more than a dozen currently available to the synthetic-image generator.

By using the synthetic data sets, we can perform experiments to evaluate quantitatively the performance of XTRS. Figure 18 shows a graph of the recognition performance of XTRS-C and XTRS-R as the range to the target increases from 750 m to 3 km. (The angular resolution is the same as that found in our primary data set.) Each data point represents 20 trials using a mix of targets—eight instances of a tank, eight of a howitzer, and four of an APC—at various orientations and states of articulation. (The same mix of synthetic targets is used in all of the experiments described below.) It is interesting to note

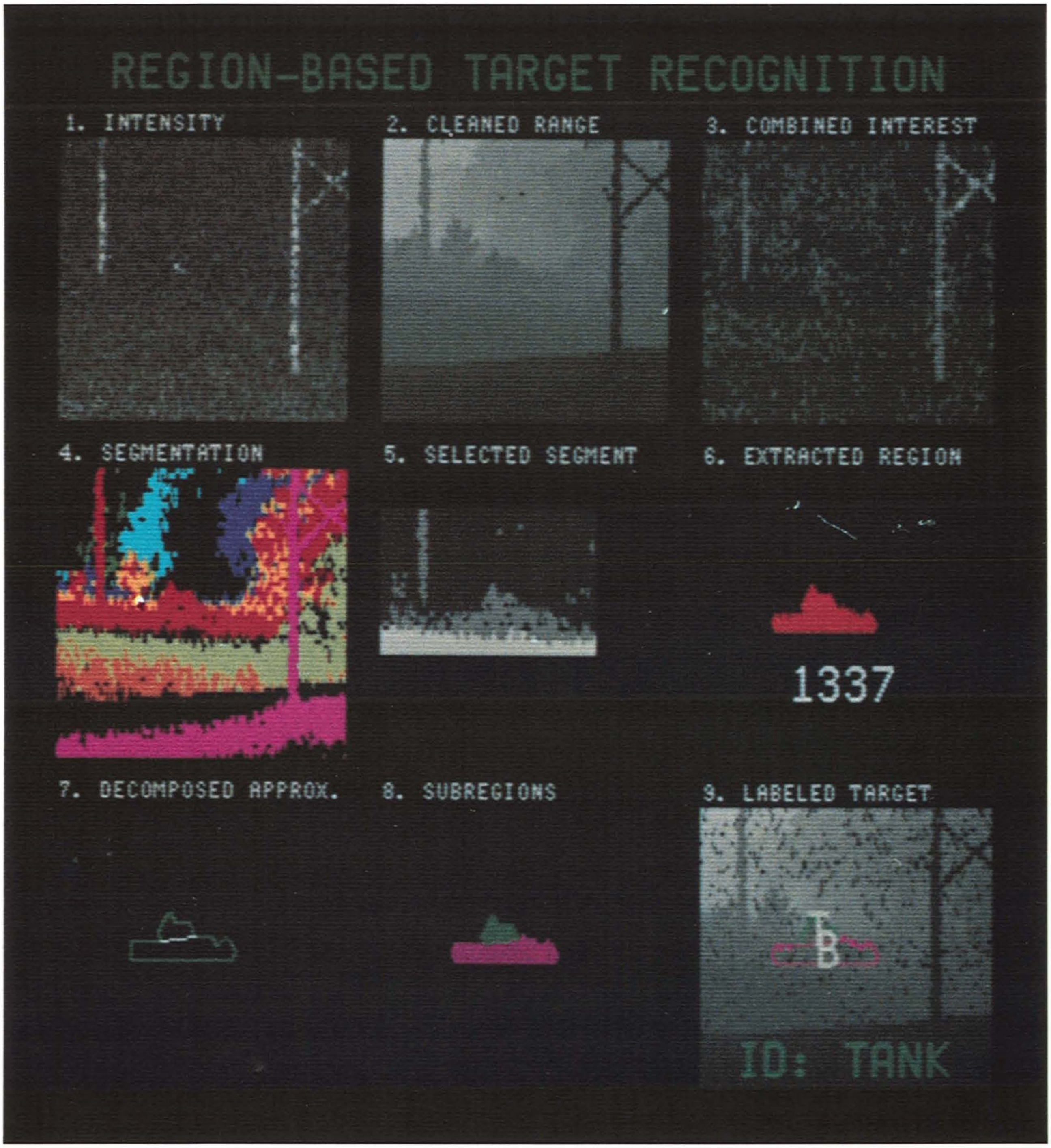


Fig. 15—Recognition example for an image with cluttered background (XTRS-R).

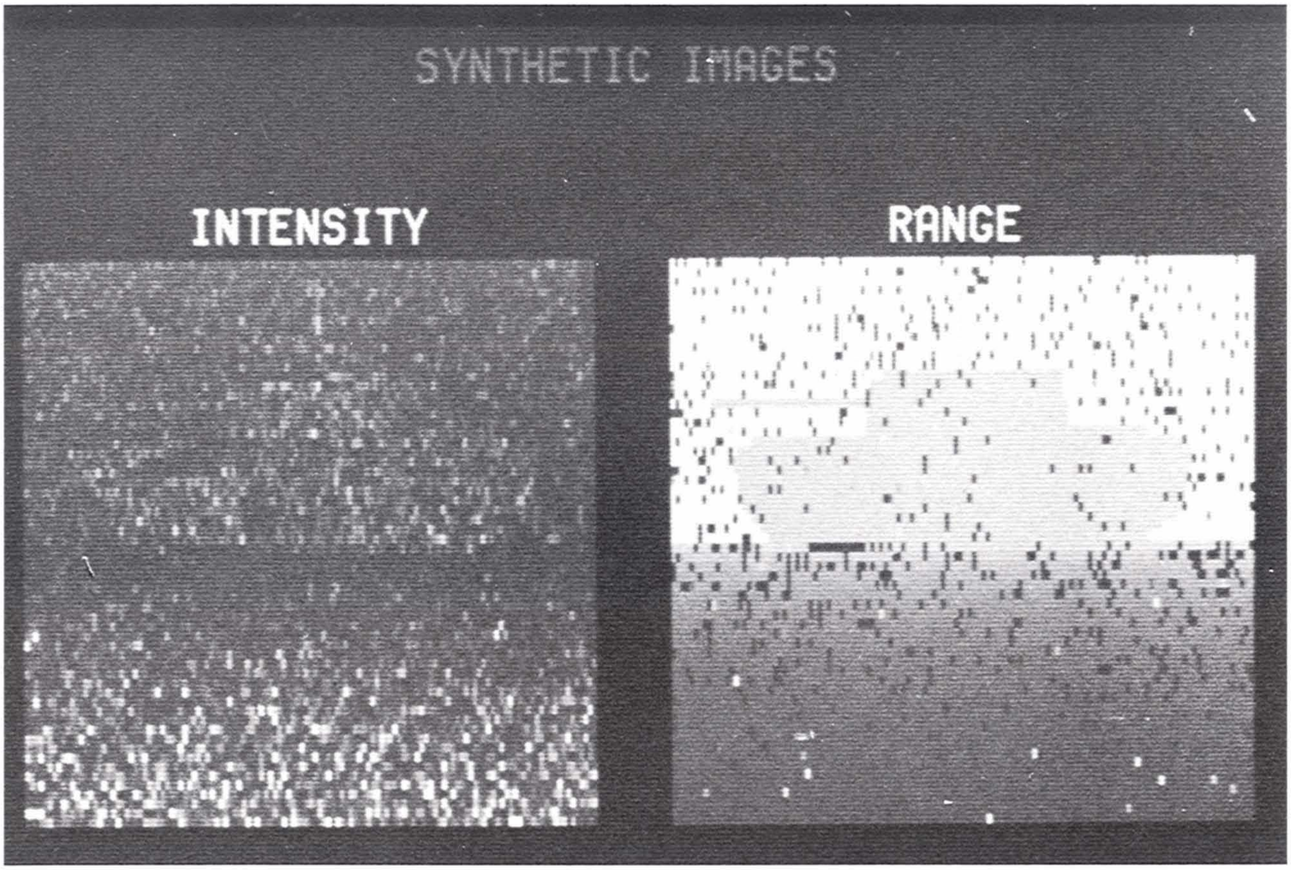


Fig. 16—Ground-based synthetic images of a tank.

that even at a range of 2 km, the imagery contains enough target pixels to allow for consistent recognition of the target.

XTRS can also be used to investigate the effects of altering some of the basic characteristics of the sensor. For example, we are interested in learning how XTRS's performance degrades with increasing levels of noise and how it improves with better range precision. Once again, we can use the synthetic-image generator for compiling appropriate data sets to address these questions. For example, Fig. 19 shows how performance degrades as the percentage of outliers in the range image increases. (The percentage of outliers is related to the carrier-to-noise ratio of the laser's return signal. Our primary data set has an outlier percentage of about 1%.)

A sensor with increased range precision will generate images with better contrast between target and background and will begin to convey some measure of the target's third spatial dimension. This capacity becomes especially important in the airborne ATR application in which large depression angles reduce the range discontinuity between the top of a target and the background behind it. Figure 20 shows how the recognition performance of XTRS-C varies with depression angle for range precisions of 2 m and 6 m. (The range precision of our primary data set is 6 m.) Figure 21 contains a similar graph for XTRS-R. Both graphs show that higher range precisions can markedly improve the recognition of targets in situations of low range contrast, which are commonly found in airborne target-recognition applications.

Areas for Future Development

There is room for improvement in all of the constituent subsystems of the generic model-based recognition system (Fig. 1). We expect

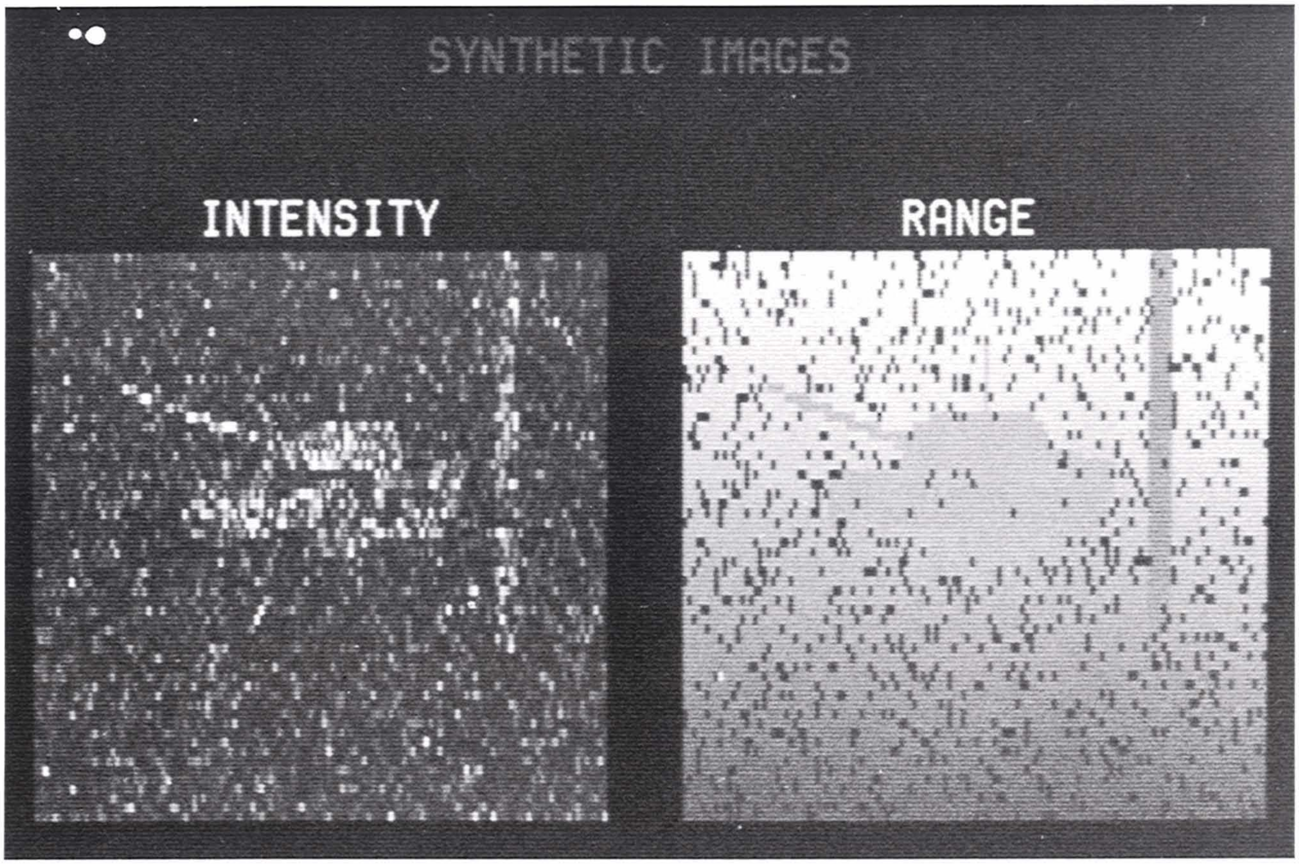


Fig. 17—Airborne synthetic images of a tank when viewed at a 5° depression angle at a distance of 1 km by a sensor with a range precision of 1 m.

improvements in the event-characterization subsystem from the following two sources. New sensors might offer new features that could be exploited by new characterization algorithms. And new algorithms might be developed with the ability to combine measurements, at the pixel or feature level, from different sensors. For example, we used the pixel-registered intensity and range measurements together in XTRS-R to extract target regions. A more sophisticated algorithm for region extraction may use several bands of visible and/or infrared imagery in conjunction with laser range and intensity images.

The realization of the model library is crucially important. Robust recognition systems must use a variety of clues to deduce the identity of an event detected in imagery. This requires the use of diverse pieces of information at the right time

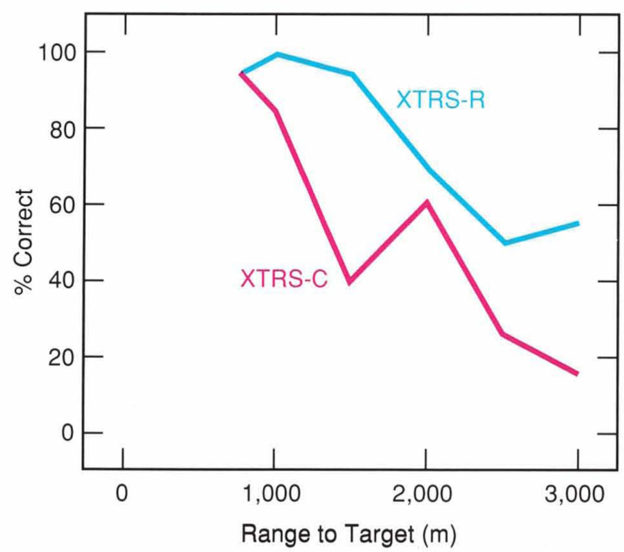


Fig. 18—Recognition performance of XTRS as a function of range to target.

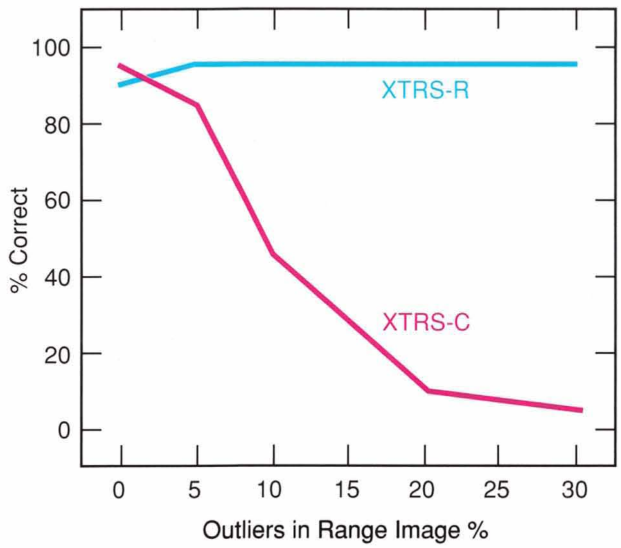


Fig. 19—Recognition performance of XTRS as a function of noise.

in the matching process. The information represented in the model library must be easily accessible to the matching subsystem in order to keep the computational cost reasonable.

The control subsystem needs significant improvement to provide greater flexibility and to implement more effectively the principle of attention focusing described earlier. The subject of control is probably the most complex and least studied area of ATR and, in general, of the computer-vision field. However, promising organization principles are beginning to emerge, such as blackboards and schemas [22].

Although building the individual components of a large ATR system draws on powerful mathematical and theoretical concepts, assembling these components into a system is currently more of an art than a science. No global theoretical framework for building a complete vision system currently exists. The only guiding principle adopted by virtually all systems is the division of the problem into three stages, often referred to as low-level vision (silhouette extraction in XTRS), intermediate-level vision (silhouette decomposition), and high-level vision (silhouette matching). However, within each level is a variety of well-developed techniques that can be applied to particular aspects of the problem.

Progress on the general ATR problem has been slow for two reasons. Mathematically analyzing the performance of a target-recognition system is difficult. So is making progress experimentally because of the large amount of imagery that needs to be processed in order to investigate ATR performance under many conditions.

Fortunately, technology continues to make cheaper and more powerful computational resources available. Computer science, moreover, is beginning to provide mechanisms for the rapid conversion of ideas into executable algorithms. For example, our productivity has benefited greatly from the use of Sketch [23], a LISP- and C-based algorithm-development environment for mixed numeric and symbolic computations. Developed at the Machine Intelligence Technology Group of Lincoln Laboratory, Sketch allows us to develop new computer-vision algorithms quickly and to test these algorithms on large data sets.

Summary

In this article, we have discussed model-based target recognition and, in particular, the XTRS under development in the Machine Intelligence Technology Group of Lincoln Laboratory. In

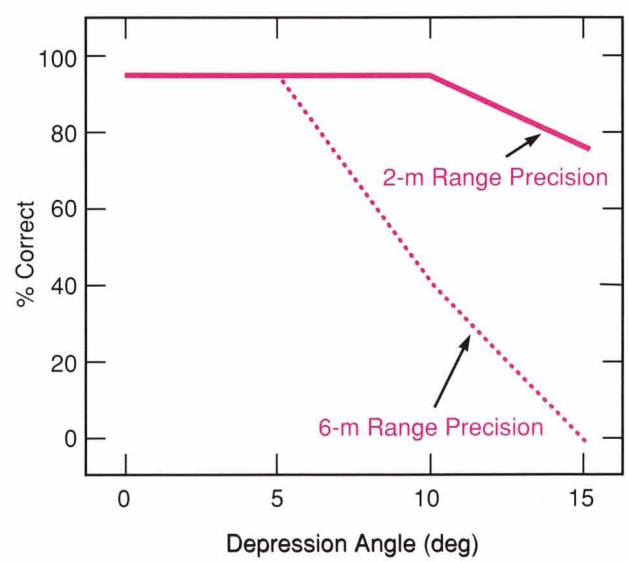


Fig. 20—Recognition performance of XTRS-C as a function of depression angle with range precision as a parameter.

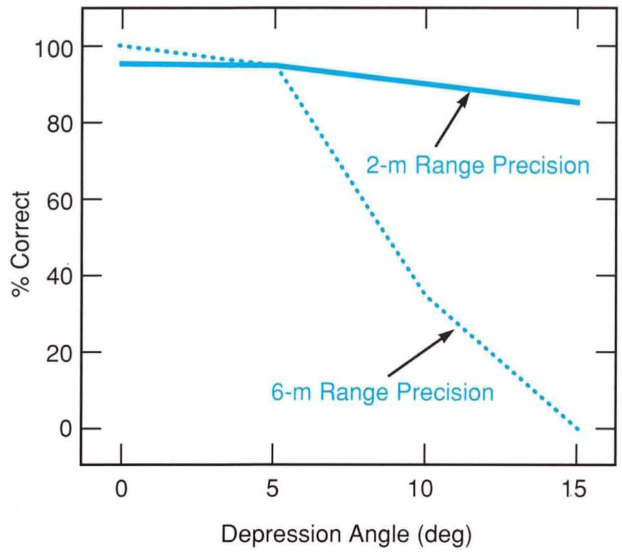


Fig. 21—Recognition performance for XTRS-R as a function of depression angle with range precision as a parameter.

XTRS, the model library takes the form of an AM hierarchy. The input to the matching subsystem consists of an AM hierarchy and the set of event characterizations derived from the imagery. Each event characterization consists of a symbolic and numeric description of a single image event—in our case, either a contour or a region silhouette—extracted from the imagery, and the descriptions of the primitives that resulted from decomposing the extracted image event. The procedure used to match the image-derived descriptions to the AM hierarchy is application independent. In fact, the only system components that change from one application to another are the AM hierarchy, which contains knowledge about the targets, and the algorithms necessary to extract the desired image events and decompose them into primitives. Thus, our approach has a wide variety of potential applications.

XTRS recognizes targets by using silhouettes extracted from range imagery of coarse range precision. We have discussed the important steps of silhouette extraction, decomposition, and matching for both the contour- and region-based systems. Both systems operate in a fully autonomous fashion, from raw data to target recognition. We observed that the system performs extremely well when dealing with images similar to those of Figs. 11 through 13. We demonstrated that the concept of AMs processed in an evidential formalism can be used to achieve good recognition of articulated, imprecisely defined targets present in real-life laser radar images with significant noise and coarse range precision.

Our approach to the design of XTRS was to use established and mathematically well-understood techniques wherever feasible (e.g., the Dempster-Shafer Theory of Evidence). However, when necessary, we did use *ad hoc* techniques, notably in the silhouette decomposition step. When we had to rely on *ad hoc* processing, we created a modular building block that satisfied three conditions: its operation had to be easily explainable in lucid terms, its behavior had to be predictable, and its domain of applicability had to be readily extendable. The AM-matching phase of XTRS illustrates our attempt to impose modularity and rigor. The system is application independent and our algorithm for the calculation of the degrees of matching has a sound mathematical foundation. The use of a modular, well-understood building block approach in XTRS allows us to reconfigure the system quickly and to replace any algorithm with an alternate processing scheme. This flexibility enables us to test, improve, and extend the system with relative ease.

In our work, we have tried to address the whole ATR problem. Our goal from the beginning was to build an end-to-end system, one that uses sensor images as its input and makes a recognition decision as its output. An end-to-end system permits us to experiment with this system, augment it, reconfigure it, and use it to discover and evaluate new approaches to the ATR problem. In the computer-vision field, performance improvements can often arise from a synergistic combination of different suboptimal processing algorithms. Conversely, replacing a particular algorithm with a better, but computationally more expensive, alternative may not significantly improve the overall performance of the end-to-end system. XTRS provides an excellent testing ground for observing and understanding such effects.

Acknowledgment

The authors thank their sponsor, the Tactical Technology Office of the Defense Advanced Research Projects Agency, and their colleagues in the Opto-Radar Group of Lincoln Laboratory.

References

1. R.A. Brooks, *Model-Based Computer Vision* (UMI Research Press, Ann Arbor, MI, 1984).
2. A.B. Gschwendtner, R.C. Harney, and R.J. Hull, "Coherent IR Radar Technology," *Series in Optical Sciences: Optical and Laser Remote Sensing*, eds. D.K. Killinger and A. Mooradian (Springer-Verlag, Berlin, 1983).
3. T. Kanade, *Three-Dimensional Machine Vision* (Kluver Academic Publishers, Boston, 1987).
4. P.L. Van Hove, "Model-Based Silhouette Recognition," *Proc. of the IEEE Computer Society Workshop on Computer Vision, Miami Beach, FL, Nov.–Dec. 1987*, p. 88.
5. P.L. Van Hove and R.A. Jaenicke, "Model-Based Silhouette Recognition," *MIT Lincoln Laboratory Technical Report TR-817* (25 Nov. 1988) AD-A202540.
6. A.R. Hanson, E.M. Riseman, and T.D. Williams, "Sensor and Information Fusion from Knowledge-Based Constraints," *SPIE* **931:** *Sensor Fusion, Orlando, FL, 4-6 Apr. 1988*, p.186.
7. J.G. Verly, P.L. Van Hove, R.L. Walton, and D.E. Dudgeon, "Silhouette Understanding System," *Proc. of 1986 IEEE-IECEJ-ASJ Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP), Tokyo, 8–11 Apr. 1986* **2,** 1457.
8. J. Serra, *Image Analysis and Mathematical Morphology* (Academic Press, London, 1982).
9. T.R. Esselman and J.G. Verly, "Some Applications of Mathematical Morphology to Range Imagery," *Proc. of 1987 IEEE Int. Conf. on Acoust., Speech, and Signal Processing (ICASSP), Dallas, TX, 6–9 Apr. 1987*, p. 245.
10. T.R. Esselman and J.G. Verly, "Applications of Mathematical Morphology to Range Imagery," *MIT Lincoln Laboratory Technical Report TR-797* (23 Dec. 1987) AD-A189316.
11. D. Marr, *Vision* (W.H. Freeman and Co., San Francisco, 1982).
12. U. Ramer, "An Iterative Procedure for the Polygonal Approximation of Plane Curves," *Comput. Graphics and Image Proc.* **1,** 244 (1972).
13. T. Pavlidis and F. Ali, "A Hierarchical Syntactic Shape Analyzer," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-1,** 2 (1979).
14. R.L. Delanoy, J.G. Verly, and B.D. Williams, *MIT Lincoln Laboratory Technical Report* in preparation.
15. M.A. Eshera and K.S. Fu, "An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph-Matching," *IEEE Trans. Patt. Anal. Mach. Intell.* **PAMI-8,** 604 (1986).
16. R.M. Quillan, "Semantic Memory," in *Semantic Information Processing*, ed. M. Minsky (MIT Press, Cambridge, MA, 1968), pp. 216–270.
17. P.G. Selfridge, "Reasoning about Success and Failure in Aerial Image Understanding," *Ph.D. Thesis* (University of Rochester, Rochester, NY, 1982).
18. J.G. Verly, B.D. Williams, and R.L. Delanoy, *MIT Lincoln Laboratory Technical Report* in preparation.
19. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, NJ, 1976).
20. J. Gordon and E.H. Shortliffe, "The Dempster-Shafer Theory of Evidence," in *Rule-Based Expert Systems*, eds. B. G. Buchanan and E.H. Shortliffe (Addison-Wesley, Reading, MA, 1984), pp. 272–292.
21. J.G. Verly, *MIT Lincoln Laboratory Technical Report* in preparation.
22. R. Engelmore and T. Morgan, *Blackboard Systems* (Addison-Wesley, Reading, MA, 1988).
23. R. Walton, J. Verly, and P. Van Hove, *Sketch 4B: An Image Understanding Operating System* (MIT Lincoln Laboratory, Lexington, MA, 1989).

Appendix: Elements of the Dempster-Shafer Theory of Evidence

This appendix contains a brief introduction to the Dempster-Shafer Theory of Evidence [1, 2]. We begin with the theory and later discuss its application.

Theory

A *frame of discernment*, represented by Θ, is defined as the set of all possible outcomes of a decision problem. The elements of Θ are assumed to be mutually exclusive and exhaustive. In the case of XTRS, the elements are "tank," "howitzer," "APC," and "none," so we can write $\Theta = \{T, H, A, N\}$. Note that the element "none" (i.e., none of the others) makes the elements of Θ exhaustive.

The power set of Θ, generally denoted by 2^{Θ}, is the set of all subsets of Θ. The power set corresponding to XTRS is shown in Fig. A.

A *state of belief* with respect to a given Θ can be represented by a *basic probability assignment*, or *bpa*, defined as the function m: $2^{\Theta} \rightarrow [0, 1]$, such that $m(\phi) = 0$ and

$$\sum_{A \subseteq \theta} m(A) = 1 \ ,$$

where ϕ denotes the empty set. This definition indicates that the *bpa* assigns a number in the range [0, 1], called *basic probability number*, or *bpn*, to each subset A of Θ or, equivalently, to each element of 2^{Θ}. The above equations respectively indicate that no belief ought to be committed to ϕ and that the total belief has a value of 1. Each subset A for which $m(A) > 0$ is called a *focal element*.

The *bpn* for A, or $m(A)$, is the measure of the belief that is committed *exactly* to A. It is not the *total* belief committed to A. Each of the subsets of A also has a belief committed exactly to it, and therefore adding all these beliefs to get the total belief in A seems reasonable. This assumption leads to the introduction of the *belief function*, or *Bel*, defined as

$$Bel(A) = \sum_{B \subseteq A} m(B) \ .$$

It follows that the values of *Bel* are also in the range [0, 1]. Figure B shows the subsets that contribute to $Bel(X)$ for the given subset $X = \{T, H, N\}$ of $\Theta = \{T, H, A, N\}$. In the simple but important case in which the subset A is a *singleton* (i.e., it consists of a single element), the equation linking beliefs and *bpns* reduces to $Bel(A) = m(A)$. (This case is important in XTRS since the decision rule we use considers only the belief in singleton target hypotheses, e.g., "the target is a tank." For composite hypotheses, e.g., "the target is either a tank or a howitzer," the more general formula should be used to assign belief.)

An example of a belief function is the *vacuous-belief function*, for which $m(A)$ is 1 if $A = \Theta$ and is 0 otherwise. Another example is the *simple support function*, for which $m(A)$ is s if $A = X$, $1 - s$ if $A = \Theta$, and 0 otherwise. (A "support" s is assigned to X and the balance $1 - s$ is assigned to Θ. X is sometimes referred to as the *focus* of the belief function). The vacuous-belief and simple-support functions play a key role in the matching subsystem of XTRS.

The orthogonal sum $m_1 \oplus m_2$ of two *bpas* m_1 and m_2 is given by

$$m_1 \oplus m_2(A) = \begin{cases} 0 & \text{if } A = \phi \\ \text{"undefined"} & \text{if } \sum\limits_{B \cap C = \phi} m_1(B)\, m_2(C) = 1 \\ \dfrac{\sum\limits_{B \cap C = A} m_1(B)\, m_2(C)}{1 - \sum\limits_{B \cap C = \phi} m_1(B)\, m_2(C)} & \text{otherwise,} \end{cases} \tag{1}$$

where the result "undefined" really means that m_1 and m_2 correspond to totally conflicting pieces of evidence. It can be shown that the orthogonal sum $m_1 \oplus m_2$ is itself a *bpa*. Therefore, using the orthogonal sum operation $\oplus$ to combine *bpas* is legitimate.

This method for combining two *bpas* (or, equivalently, two belief functions) into a new *bpa* (or belief function) is known as *Dempster's rule of combination*. Note that the interpretation of the orthogonal sum as a rule for combining evidence is valid only if the pieces of evidence whose *bpas* are combined are independent. (In the Dempster-Shafer Theory of Evidence, the notion of independence is not as precisely defined as in the Bayesian case. Common sense appears to be the best guide in this matter.)

If many *bpas* need to be combined, then they must be done so pairwise. Because the orthogonal sum is both commutative and associative, the order in which the *bpas* are combined is irrelevant.

As Shafer himself indicated [1], there is no *a priori* justification for Dempster's rule of combination. However, the *a posteriori* justification is that the rule generally, but not always, produces intuitively meaningful results if the underlying pieces of evidence being combined are independent.

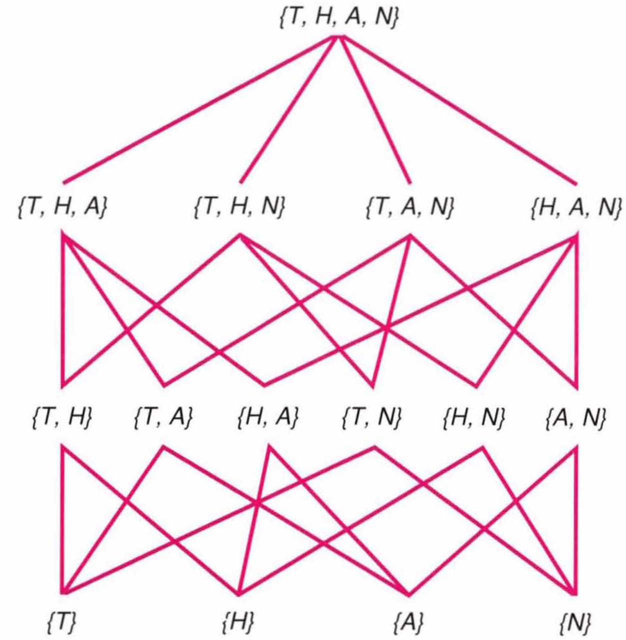


Fig. A—Power set 2^Θ of the frame of discernment Θ = {T, H, A, N}. The empty set ϕ has been omitted.

Application

Now we consider an example of evidence accumulation that illustrates the above concepts in a context similar to that of XTRS [2, 3].

To simplify the discussion, we ignore the issue of pruning. We assume that we start in a state of complete ignorance with respect to Θ, which the vacuous-belief function represents. It is further assumed that each piece of evidence either confirms or does not confirm a singleton hypothesis, i.e., one of the elements of Θ. If some evidence confirms an element a of Θ to a degree s, then the evidence is represented by a *simple-support function* whose support s is focused on the subset $\{a\}$. If the evidence does not confirm a to a degree s, then the evidence is represented by a *simple-support function* with support focused on the subset that is the *complement* of $\{a\}$ with respect to Θ. In this example, we suppose that the first piece of evidence confirms T with a support 0.7; the second piece does not confirm H with a degree 0.4 (i.e., confirms $\overline{\{H\}} = \{T, A, N\}$ with support 0.4); and the third confirms A with support 0.3.

Table 1 shows the evolution of the belief accumulation for this example. In the table, each row corresponds to a new piece of evidence. Each column corresponds to that subset of Θ which is one of the focal elements of the belief function that describes the final result of the accumulation.

For the first piece of evidence, the bpns for the focal elements of the corresponding simple support function are $m_1(\{T\}) = 0.7$ and $m_1(\Theta) = 0.3$. By applying Dempster's rule of combination, we can easily see that the result of combining the initial vacuous-belief function with this first simple-support belief function yields this first function. The result is shown in the second row of Table 1.

For the second piece of evidence, the relevant *bpns* are $m_2(\{T, A, N\}) = 0.4$ and $m_2(\Theta) = 0.6$. After using Eq. 1 to combine the result of the previous operation with the new belief function, we obtain the result shown in the third row of Table 1. Note that some belief is assigned to the triplet $\{T, A, N\}$, but none to the singleton $\{A\}$. Also, note that the belief in $\{T\}$ remains unchanged and the fraction of belief assigned to the triplet actually comes from that for $\Theta = \{T, H, A, N\}$. (Refer to Ref. 2 for a more detailed description of the calculations involved.)

The case of the last piece of evidence is particularly interesting because it illustrates a partial conflict. Conceptually, the new piece of evidence partially contradicts the current state of belief. Indeed, while the current state shows some belief for T and none for A, the new evidence shows belief for A but none for T. Mathematically, this partial conflict manifests itself by a nonunity denominator in the formula for the orthogonal sum (Eq. 1), which leads to a final scaling of all *bpns*. By looking at the formula, the reader will see that the reduced *bpn* for $\{T\}$ is due to this rescaling alone, and the reduced *bpn* for $\{T, A, N\}$ involves both an addition of support and the rescaling.

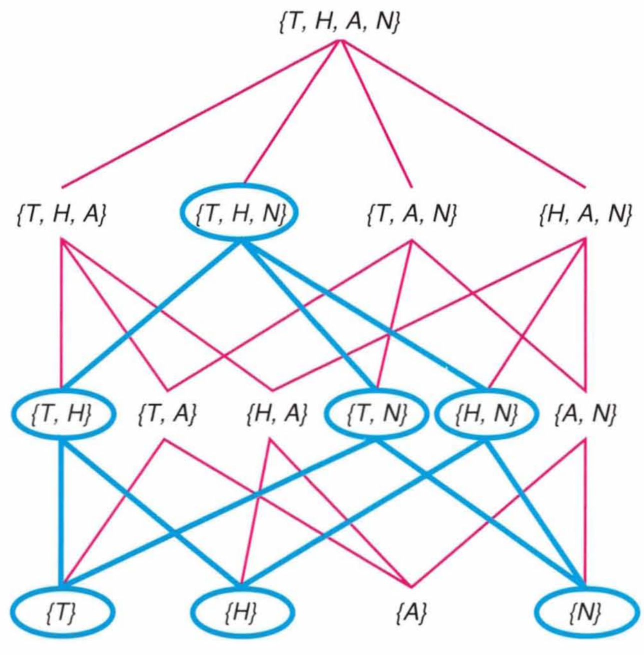


Fig. B—Subsets of Θ = {T, H, A, N} that contribute to the value of Bel ({T, H, N}).

The conclusion drawn from this example is that the belief in T (i.e., "the object is a tank") is 0.620, the belief in A (i.e., "the object is an APC") is 0.114, and the belief in all other singletons is 0.0. Note that although the *bpn* for the compound hypothesis "the object is either a tank, an APC, or an unknown vehicle" is 0.106, the total belief in that hypothesis is actually 0.840, which is the sum of 0.106, 0.620, and 0.114. However, the current version of XTRS does not consider such compound hypotheses.

References

1. G. Shafer, *A Mathematical Theory of Evidence* (Princeton University Press, Princeton, NJ, 1976).
2. J.G. Verly, *MIT Lincoln Laboratory Technical Report* in preparation.
3. J.G. Verly, B.D. Williams, and R.L. Delanoy, *MIT Lincoln Laboratory Technical Report* in preparation.

Table 1. Evolution of Basic Probability Assignments

	BASIC PROBABILITY ASSIGNMENT			
EVIDENCE ADDED	$\{T, H, A, N\}$	$\{T, A, N\}$	$\{T\}$	$\{A\}$
None	1.0	-	-	-
$\{T\}$ (0.7)	0.3	-	0.7	-
$\{H\}$ (–0.4)	0.18	0.12	0.7	-
$\{A\}$ (0.3)	0.159	0.106	0.620	0.114

JACQUES G. VERLY was born in Liège, Belgium. He received the Ingénieur Electronicien degree from the University of Liège in 1975. Through a sponsorship of the Belgian American Educational Foundation, he came to the United States in 1975 and received an M.S. and a Ph.D. in electrical engineering from Stanford University in 1976 and 1980, respectively. At Stanford he performed doctoral research in the fields of image reconstruction and restoration. Upon graduation, he joined Lincoln Laboratory, where he now works in the Machine Intelligence Technology Group on multisensor image understanding applied to laser radar data. He has published more than 30 papers in the areas of image processing, optical signal processing, distributed signal processing, mathematical morphology, computer vision, automatic target recognition, and image reconstruction from tomographic projections. Jacques is a CRB Fellow of the Belgian American Educational Foundation.

RICHARD L. DELANOY is a staff member of the Machine Intelligence Technology Group of Lincoln Laboratory. His work spans the fields of learning, machine intelligence, and computer vision. From 1980 to 1983, he was a research scientist at the University of Virginia Department of Psychology, where he investigated the effects of stress-related hormones on electrophysiological models, and biochemical correlates of learning. Before joining the Laboratory in 1987, he worked for GE Fanuc Automation N.A., Inc. as a software analyst. Dick received a B.A. in biology from Wake Forest University in 1973, a Ph.D. in neuroscience from the University of Florida's College of Medicine in 1979, and an M.S. in computer science from the University of Virginia in 1987. He was an NSF Predoctoral Fellow and an NIMH Postdoctoral Fellow.

DAN E. DUDGEON is an assistant leader of Lincoln Laboratory's Machine Intelligence Technology Group. He is involved in studies of image processing, computer vision, and computer architectures for algorithm development and real-time implementations. Before joining the laboratory in 1979, he worked at Bolt, Beranek, and Newman, Inc. in Cambridge, Mass. Dan was educated at MIT, where he received an S.B. and an S.M. in electrical engineering, an E.E., and an Sc.D. in signal processing. In 1976, he was corecipient of IEEE's Browder J. Thompson Memorial Prize for best paper by an author under the age of 30. He is the coauthor of *Multidimensional Digital Signal Processing*, published by Prentice-Hall. Because of his contribution to the field of multidimensional signal processing, he was elected an IEEE Fellow in 1987. He is also a charter member and past chairman of the IEEE Acoustics, Speech, and Signal Processing Society's Technical Committee on Multidimensional Signal Processing. He is currently serving as secretary of the ASSP Society Administrative Committee, and he was named a Distinguished Lecturer of the ASSP Society in 1988.