Image Understanding Workshop, 1996 Proceedings
Palm Springs, CA, sponsored by ARPA, Morgan Kaufmann
Publishers, Inc., Vol. II, pp. 1263-1276, February 12-15, 1996

Model-based SAR ATR System*

Katsushi Ikeuchi, M. D. Wheeler, Taku Yamazaki, and Takeshi Shakunaga

School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213

WWW node: **http://www.cs.cmu.edu:8001/afs/cs/project/vision/www/vasc.html**

Abstract

*Recognizing a target in synthetic-aperture radar (SAR) images is an important, yet challenging, application of the model-based vision technique. This paper describes a model-based SAR recognition system based on invariant histograms and deformable template matching techniques. An invariant histogram is a histogram of invariant values defined by geometric features such as points and lines in SAR images. Although a few invariances are sufficient to recognize a target, we build a histogram of all invariant values given by all possible target feature pairs. This redundant histogram enables robust recognition under severe occlusions typical in SAR recognition scenarios. Multi-step deformable template matching examines the existence of an object by superimposing templates over potential energy field generated from images or primitive features. It determines the template configuration which has the minimum deformation (deformation energy) and the best alignment of the template with features (potential energy). The deformability of the template absorbs the instability of SAR features. We have implemented the system and evaluated the system performance using hybrid SAR images, generated from synthesized model signatures and real SAR background signatures.**Introduction*

*This research was sponsored in part by the U.S. Advanced Research Project Agency under Wright Research and Development Center, U.S. Air Force under Contract F33615-93-1-1282, in part by the U.S. Advanced Research Project Agency under Army Research Office, the Department of the Army under Grant DAAH04-94-G-0006, and in part by the Office of Naval Research. the Department of the Navy under Grant N00014-93-1-1220. Views and conclusions contained in this document are those of the authors and should not be interpreted as necessary representing official policies or endorsements, either expressed or implied, of the Department of the Air Force, the Department of the Army, the Department of the Navy or the United States Government.

1. Introduction

Recognizing targets in synthetic-aperture radar (SAR) images [Tomiyasu, 1978, Chellapa et al., 1992] is a difficult problem for conventional computer vision systems. First, all SAR features are non-attached; they are not tightly related with surface markers nor explicit object geometry such as edges. Rather, they are floating over a target surface. Thus, they suddenly appear, disappear, and abruptly change their shapes due to tiny movements by an observer. Secondly, in SAR image recognition, objects are often intentionally hidden from an observer. For example, enemy tanks are often hidden under trees. A whole tank may be camouflaged completely with a camouflage net.

Historically, target recognition in SAR images is attacked using three different approaches: statistical pattern recognition [Novak at al., 1993], model-based [Kuno, 1988, Sato, 1992], and artificial neural network [Waxman, 1993]. Among these three approaches, model-based approach [Bolles and Cain 1982, Brooks, 1983, Low, 1985, Huttenlocher and Ullman, 1987, Grimson, 1990, Gremban and Ikeuchi, 1993, Wheeler and Ikeuchi, 1995] is the most promising, because of its potential for the robust recognition. In essence, a model-based system analyzes each image in detail and identifies each part of a signature contribution toward recognition, while pattern recognition and artificial neural network based recognition system handle a target signature as a whole. This capability of part analysis in the model-based vision approach provides the potential for the robustness with respect to partial occlusion of target, and cluttered background.

Promising features from SAR appearances are isolated peaks. Among several proposed model-based techniques, pose clustering is suitable for determining the object pose (and identifying the object) from such sparse features. Representative pose

clustering techniques include: Hough transform and geometric hashing. Ballard [1981] generalized the Hough transform to detect arbitrary patterns. Recently, several alternative techniques have been proposed by Lamdan and Wolfson [19881, Dhome et al. [1986], and Stockman [1987]. Grimson [1990] reported that searching pose space with Hough transforms is very effective for 2D object recognition. Wolfson and Lamdan [1992] also reported an effective recognition system using geometric hashing. These pose clustering techniques are highly optimized so that each relation among a pair of features can reduce the possible interpretations as much as possible. However, pose clustering becomes unstable when relative relationships among features vary as is the case of non-attached SAR features.

Recently, several model-based recognition systems have been designed using geometric invariants [Mundy ans Zisserman, 19921. Geometric invariants such as the cross-ratio provides very efficient clue for identifying 3D objects. In this paper, we will denote those geometric invariants as *strong invariants.* Those strong invariants require the correspondence problem to be solved prior to applying such invariants to recognition. Although this may be an easy problem when an object contains a few feature points, combinatorial explosion occurs when handling cluttered images typical of SAR images.

This paper introduces an invariant histogram based on weak *invariants,* defined by a pair of features, to avoid the difficult correspondence problem. Though each invariant is weak for constraining possible object classes (and their poses), we demonstrate that a histogram of weak invariants can be used to identify the object uniquely. Moreover, utilizing all of the weak invariants in an image is highly redundant, and provides robust recognition under severe occlusion with unstable SAR features.

We have built a recognition system that consists of indexing and verification. The indexing module quickly reduces the number of candidates using the invariant histogram technique. To select the correct candidate, the verification module employs deformable template matching to test for the existence of each feature. Here, each SAR feature is non-attached and can vary its position. Deformations are necessary for fine-tuning each feature positions locally.

The system is designed under the vision algorithm compilation paradigm [Ikeuchi, 19881. The system has *two* modes: off-line and on-line. In off-line mode, model invariant histograms and deformable templates are generated from target models using XPATCH SAR simulator. In on-line mode, an image invariant and potential fields are computed from an input image and our indexing and verification algorithms are applied.

Section 2 will introduce the concept of our invariant histogram technique, and section 3 describes how to use the technique for designing the indexing module. Our deformable template matching method will be discussed in Section 4. Section 5 presents our experimental results, and in Section 6 we present our conclusions.

2. Invariant Histogram

In order to achieve robust recognition under severe occlusion or camouflage with unstable SAR feature, our system introduces an invariant histogram based on weak invariants, defined by a pair of features. The indexing module employs this invariant histogram of weak invariants. The indexing quickly reduces the number of the possible candidates before expensive candidate verification. It employs a dictionary lookup method. The dictionary consists of the invariant histograms, distributions of invariant values of an object. By comparing the observed invariant histogram with model histograms in the dictionary, the module decides which candidates are the most likely ones. This process requires to measure similarity between an input and a model invariant histogram. The section will also discusses on the similarity measure defined on the invariant histogram.

2.1. Concept of invariant histogram

An invariant histogram stores many invariant values from a target model, though only a few invariants are actually necessary for recognition. Thus, this invariant histogram is a redundant space. It is robust against variation in computed invariant value typical of SAR data.

This paper employs weak invariants, such as distance of two points or the slope of the bisecting

line of two lines. Strong invariants such as a cross ratio of four points on a line are convenient for object recognition yet difficult to reliably extract from real data. We instead relay on weak invariants defined using only pairs of primitive features in this system. This is because known strong invariants requires too many primitive to be extracted reliably; for example a cross ratio needs four points to be identified in an image. Detecting a group of features is rarely possible in the SAR domain, where most features are quite unstable.

In our SAR recognition system, two kinds of primitive features are extracted from an image: points and line segments. All feature points are detected by applying an interest operator. All line segments are detected by a line detector based on the Canny's edge detecting technique.

When a target rotates in 3D space, the appearance of the target in SAR images drastically changes. On the other hand, even though a target translates along the ground plane, the appearance is not significantly altered. Thus, we decide to use translation invariants to construct invariant histograms. Figure 1 shows six translation invariants which are used for constructing our invariant histograms.[1]

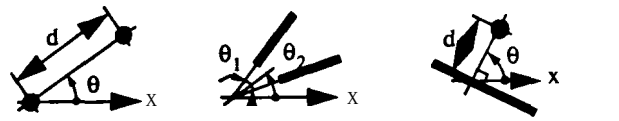
2.1.1. Point-Point (PP) histogram **(Figure 1(a))**
Distance and direction between a pair of points is a translation invariant. This invariant is calculated for a pair of primitive feature points, and histograms are made in the 2D displacement space[2].

2.1.2. Line-Line (LL) histogram **(Figure 1(b))**

An angle between two line segments and a slope of their bisecting line are invariant to translation. We use these two invariants for characterizing a line pai?.

2.1.3. Point-Line (PL) histogram **(Figure 1(c))**

An orthogonal distance from a point to a line is invariant to translation and rotation. An orthogonal direction from a point to a line segment is also invariant to translation. We use these orthogonal distance and its direction for characterizing a point-line pair[4].



(a) displacement & direction (b) angle & slope of bisecting line (c) orthogonal direction & orthogonal distance

Figure 1 Six invariants used in our implementation

2.2 Implementation of invariant histogram

A two dimensional invariant histogram is implemented as a collection of tessellated bins. Each pair of geometric features provides a pair of invariant values. Those values are then histogrammed to the corresponding bins in the corresponding 2D invariant histogram. At the same time, the bin maintains pointers to keep track of the original primitive pairs that vote for it. Since several pairs may lie in the same bin, each bin may contain multiple pointers. These pointers will be utilized later for establishing initial correspondences for verification between image and model features.

Figure 2 shows a procedure for generating an invariant histogram from an image. First, primitive features, point features in this example, are extracted from an image. From point pairs, invariant values are obtained: distance and direction. When making point pairs, the system consider only local feature pairs, those within a certain threshold, indicated as a circle

1. In order to increase the robusmess of the system against camouflage and surrounding noise, we do not use properties of peaks or edges (such as brightness values of a peak or area size of a peak); we only use spatial relations among peaks and edges.

2. The 2D space is composed of a 2D array, of which each cell has widths of 2 or 4 pixels along x- and y-axes in our implementation.

3. We do not use all the line segment pairs to make an LL histogram. A nearby subset is first generated from all the line segment pairs, so that the minimum distance between two line segments is less than a threshold. The coupled invariant is then calculated over the line pair subset. The resulting invariant histogram is in 2D space whose dimensions correspond to the angle and the slope of bisecting line. Both the angle and the slope are quantized to 10 degree intervals in our implementation.

4. To construct a PL histogram, a subset of point-line pairs is first made up from all the pairs, so that the foot of the perpendicular is included in the line segment. Then the coupled invariant is calculated over the point-line pair subset. PL histograms are made in the 2D space of which two axes correspond to the distance and the direction. The distance is quantized to intervals of 2 or 4 pixels, and the direction is quantized to 10 degree intervals in our implementation

in the figure. The horizontal axes in the indicate values of distance and direction, and the vertical axis denotes the number of votes for each bin.

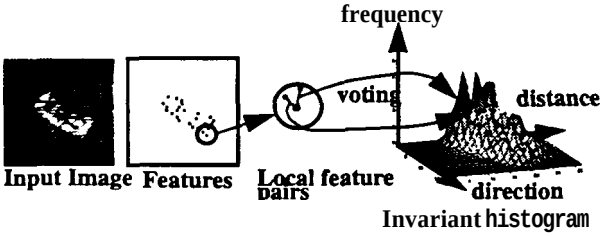


Figure 2 Invariant histogram generation

To minimize problems due to quantization, we smooth the histogram over a small neighborhood. We use weighted voting for the four nearest neighbor bins of the real point in the invariant space. The weights are calculated so as to be inversely proportional to the distance from the centers of the bins to the real point. The sum of four weights is normalized for each occurrence. At the same time, the pointers to the feature pair also copied to the four bins.

3. Indexing by Dictionary Lookup

We have described the details of the invariant histogram representation. Now we will describe how we utilize these histograms to eliminate candidate hypotheses in our recognition algorithm in the indexing module. We can build invariant maps for each representative view, and use these maps to compute distance measures between an invariant map computed from the image and each candidate. The candidates can be ranked by this distance measure and then pruned accordingly. In this process we refer to the collection of invariant histograms as a dictionary which represents how a target object appears, and thus, invariant values change depending on pose parameters. This dictionary is constructed from model appearances at off-line.

3.1. Structure of a dictionary

Pose parameters can be decomposed into two categories: invariant and variant pose parameters with respect to a weak invariant. Invariant pose parameters do not alter the invariant value; variant pose parameters do. In our current implementation. translation of a target does not change our invariant values, while rotation does change their values. Thus, translation and rotation parameters are invariant and variant pose parameters respectively.

We will construct a dictionary, a collection of invariant histograms, to cover all of the variant parameter space. Rotation parameter spaces are evenly sampled, and invariant histograms are constructed at these sampled rotation values. Here, each sampled rotation value is denoted as a representative view.

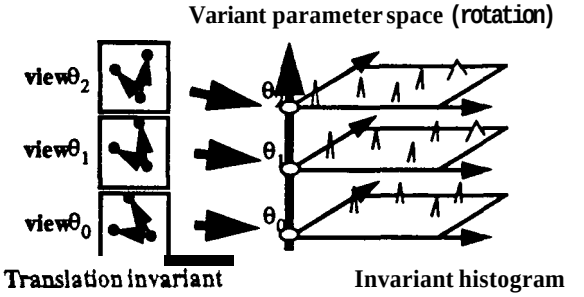


Figure 3 Invariant and variant space

For each interval, we compute the average and variance histograms over the interval. Some features appear and disappear abruptly, while other features may be observable from a wide range of viewing directions. Histogram values voted by such abrupt pairs are unstable and unreliable for indexing, while others are stable and reliable. A variance histogram conveys this reliability measure. We take a large number of neighboring images around each representative view and generate histograms for each. Then the average and variance histograms are computed from these surrounding histograms; this cumulative invariant histogram is used as a histogram of the particular dictionary entry.

3.2. Similarity measure for invariant histograms

This section will describe a similarity measure for comparison between image and model histograms in a dictionary. A model histogram comprises average and variance histograms. Basically, average values in a model histogram are compared with those from input image and, then, the difference will be weighted using variance values.

One simple similarity measure is L1 norm as follows:

$$L = \sum_{i,j} |x_{i,j} - m_{i,j}|, \qquad (1)$$

where $x_{i,j}$ is an image value in bin (i,j), while $m_{i,j}$ is a model value. This difference will be calculated

over the all of the histogram.

Some values in bins are less reliable than other values depending on the reliability of values at bins; we will adjust the difference using a variance, value $\sigma_{i,j}$ at each bin:

$$L_1 = \sum_{i,j} \frac{|x_{i,j} - m_{i,j}|}{\sigma_{i,j}} \quad (2)$$

The L_1 norm provides severe penalty, when some features are occluded and values disappear from a histogram. In order to avoid such effect from occlusion, we further modify the measure by introducing the saturation factor, $k\sigma_{i,j}$. Namely, if the difference between the observed and model values are larger than this saturation factor, the penalty imposed is the saturation factor instead of the real distance:

$$L_{1,sat} = \sum_{i,j} min\left(\frac{|x_{i,j} - m_{i,j}|}{\sigma_{i,j}}, k\sigma_{i,j}\right). \quad (3)$$

When a histogram does not have a value in one bin, the variance, $\sigma_{i,j}$ is zero; we cannot evaluate the value. Thus, we will add a constant variance σ_o :

$$L_{1,sat} = \sum_{i,j} min\left(\frac{|x_{i,j} - m_{i,j}|}{\sigma_{i,j} + \sigma_o}, k\sigma_{i,j}\right). \quad (4)$$

Here, two constants, σ_o and k are obtained empirically.

3.3. Implementation of indexing algorithm

Using this similarity measure, we will design the following four step indexing algorithm. Since there are three different histograms, PP, LL, and PL, their relative weights are adjusted using normalization factors given by the maximum distance values over the bins of the histogram.

3.3.1. Step 1: Absolute distance

For each of the PP, LL and PL histograms, the absolute distance is calculated between an image and each model histograms. The absolute distance is given by the L1 norm with saturation given by the equation (4).

332. Step 2: Relative distance

For obtaining relative distance, the maximum distance between corresponding bins of the image and model histograms is determined for each of the PP, LL and PL histograms using

$$a_{max} = max_{i,j}\left\{min\left(\frac{|x_{i,j} - m_{i,j}|}{\sigma_{i,j} + \sigma_o}, k\sigma_{i,j}\right)\right\}. \quad (5)$$

For example, we will use $\frac{L^{PP}_{1,sat}}{a^{PP}_{max}}$ as the relative distance between two PP histograms.

333. Step 3: Total distance

The total distance is, thus, defined by:

$$L^{total}_{1,sat} = \frac{L^{PP}_{1,sat}}{a^{PP}_{mor}} + \frac{L^{LL}_{1,sat}}{a^{LL}_{max}} + \frac{L^{PL}_{1,sat}}{a^{PL}_{mar}} \quad (6)$$

33.4. Step 4: Candidate screening by total distance

The most likely representative view is determined by the total distance between the input and model histograms. Since the indexing is not to determine one particular view but to select multiple possible candidate views, we select those with distance less than a certain threshold value.

4. Pose Clustering using Invariant Histogram

After obtaining variant pose parameters (rotation parameters), we will determine the invariant pose parameters (translation parameters) using the correspondences between image and dictionary features through an invariant histogram. First, we will explain how to establish these correspondences using invariant histograms. Then, we will describe our method for obtaining invariant pose parameters by pose clustering.

4.1. Sampling correspondences

Each bin of an invariant histogram has pointers to the primitive features that vote for this bin. By retrieving the pointers of corresponding bins of the input and model histograms, we can establish correspondences between image and model primitive features.

Let us consider a case of the LL histogram as an example as shown in Figure 4. By tracking pointers, three line pairs are retrieved in a image as candidates of one line pair of a model. The two translation values are computed for each candidate. These translation values are computed by comparing between the middle points of lines. If these two translation values are near to each other, the corre-

spondence can be established. Then their average is combined, yielding the average translation parameters and the transformation is given to the pose clustering algorithm. Otherwise, the line pair correspondence is removed as a false correspondence.

Assume that n image feature pairs are referenced from a bin, and m model feature pairs are referenced from the corresponding bin. We will consider mn possible correspondences in this bin. It is noted that no more than $Min(m,n)$ ones are correct among the mn correspondences, if the one-to-one mapping holds between the input and the model features. These correct correspondences generate the correct pose candidates in the invariant parameter space, while the other correspondences generate false pose candidates. When the number of possible correspondences is too large, we do not have to consider all of them. The possibility that the values of the model's invariants randomly occur in the input image is very low compared to actual occurrences due to the model's presence in the image. Thus, random sampling can achieved a large reduction of the computation time with little or no loss in the detection rate.

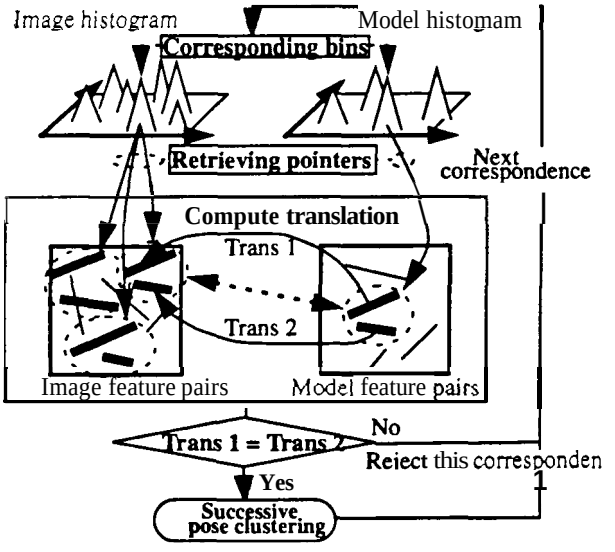


Figure 4 Pose clustering through an invariant histogram

4.2. Successive pose clustering

Several kinds of techniques are used for pose clustering [Ballard, 1981, Dhome et al, 1986, Lamdan and Wolfson 1988, Clemens and Jacobs, 1991]. The most popular technique is the generalized Hough transform in which voting is applied to the quantized pose space. Although these voting methods are easy to implement, we have to determine the size of each cell in the quantized space before execution. The quantization is closely related to uncertainty which is difficult to estimate.

We implement a successive clustering algorithm to avoid the difficulty of quantization. This method successively generates clusters without voting. By tracking pointers in bins, some feature correspondences are established and a candidate pose can be obtained. This pose candidate is examined whether it is within a certain distance to one of the existing clusters. If it is within distance from several clusters, the largest cluster will be selected to include the new candidate. The average pose and the size of the cluster will be updated at each iteration.

The clustering process terminates either when the size of the largest cluster is large enough or when the total number of generated pose candidates reaches a threshold. In both cases, the largest cluster is selected as giving the final results.

Pose clustering provides a rough estimate of translation parameters (invariant parameters), while a dictionary lookup gives a rough estimate of rotation parameters (variant parameters). Using these estimates of the pose parameters, the pose is refined using deformable template matching before final verification.

5. Verification through Deformable Template Matching

Deformable template matching examines the existence of a target by superimposing deformable templates over potential energy fields given by features by obtaining the template configuration of the minimum deformation (deformation energy) and the best alignment of the template with features (potential energy). A SAR image often contains a large number of non-attached features. We employ multi-step deformable template matching to avoid local minima given by these erroneous features. We start to examine the existence of a target object in the position given by the translation parameter from the previous pose clustering. Then, our recognition system uses multiple-level template and potential fields progressively from the coarse to the fine level.

5.1. Template generation

Templates are generated from model appearances generated by XPATCH simulator in off-line mode. The first two level of template matching, the coarse and medium level, shares the same non-deformable template, while the fine level matching employs a deformable template. These templates are generated at each representative view over the evenly sampled rotation space as used to sample the rotation parameter for the indexing dictionaries.

5.1.1. Coarse/Medium level Template

Non-deformable templates *are* generated from binarized model images. First, we threshold noise-free model images and binarize the output, $I(x,y)$. Then, we repeat this process eleven times around a representative view and superimpose them taking the union. The resulting superimposed binarized point distribution, $T^{\theta_o}(x,y)$ is the template at the central viewing direction, θ_o (representative view).

$$T^{\theta_0}(x,y) = I^{\theta o - 5\Delta}(x,y) \cup \ldots I^{00}(x,y) \ldots \cup I^{\theta o + 5\Delta}(x,y) . \quad (7)$$

Combining the templates is necessary to absorb all unstable *SAR* non-attached features in one template.

For the coarse and medium templates, only translation (x_t, y_t) is allowed; there is no relative movement of each point. The total energy is provided as the sum of potential energy values at each point position and the translation energy of the entire template:

$$E_{total} = E_{potential} + E_{trans} , \quad (8)$$

where

$$E_{potential} = \iint P(x,y)\, T^{\theta_o}(x + x_t, y + y_t)\, dxdy , \quad (9)$$

$$E_{trans} = k_t \left(x_t^2 + y_t^2 \right)^{\frac{1}{2}} . \quad (10)$$

$P(x,y)$ is the potential field function given from an input image and k_t is a spring constant. Both the coarse and medium level matching uses the same value for this spring constant. This translation term is introduced to give the priority to positions close to the one given by feature correspondences.

5.1.2. Fine level template

The fine level matching employs deformable templates: each feature point moves freely relative to other points. At this level, there is no translation of the entire template. The deformations are necessary to account for typical perturbations in position of *SAR* features.

These deformable templates are generated using a point feature extractor. First, a noise-free model image of a target object, $I(x,y)$, is convolved with a Gaussian filter. From this smoothed image, $I_{gauss}(x,y)$, we extract isolated brightness peaks, $I_{point}(x,y)$ using our regular point feature extractor. This is a binary distribution; $I_{point} = 1$ at a peak and *0* otherwise. In the same way as the non-deformable templates, eleven such point distributions around a representative view, θ_o, are superimposed.

$$D^{\theta_o}(x,y) = \sum_{i=1}^{p} \delta(x - x_i, y - y_i) , \quad (11)$$

where p is the total number of points over eleven point distributions.

The total energy of this template is:

$$E_{total} = E_{potential} + E_{deform} , \quad (12)$$

where

$$E_{potential} = \sum_{i=1}^{p} \iint P(x,y)\, \delta(x - x_i - \Delta x_i, y - y_i - \Delta y_i)\, dxdy , \quad (13)$$

$$E_{deform} = \sum_{i=1}^{p} k_d \{ (\Delta x_i)^2 + (\Delta y_i)^2 \}^{\frac{1}{2}} . \quad (14)$$

5.1.3. Difference template

Confusion often occurs between one pose and its counter pose (rotated 180 degrees from the original pose). In order to avoid confusion between a pair of poses, our system employs a difference-template as the fourth step of matching. This template suppresses common parts and emphasizes conflicting parts between the pair. This difference templates are used only when it is necessary *to* disambiguate a pair candidates of close score.

In order to make a difference template prior to execution, first, the best possible alignment of a pair of coarse-level templates is obtained. Let us denote a

pair of poses as A and B. A coarse-level potential field, P_B is generated from a template T_B. Then, the ternplate T_A will be applied to the potential field, P_B to obtain necessary translation $(\Delta x_A, \Delta y_A)$ for optimally superimposing template, T_A over the template, T_B. See Figure 5.

Using this translation value, we superimpose a pair of fine-level templates to extract the common points in fine-level template, D_A. Then, the common points are suppressed in the template and the difference template for pose A is:

$$S_A(x+\Delta x_A, Y+\Delta y_A) = D_A(x+\Delta x_A, y+\Delta y_A) - D_B(x,y) \otimes D_A(x+\Delta x_A, y+\Delta y_A) .$$

By exchanging A and B, we will also obtain the difference template of pose B.

5.2. Generating the potential fields

Three potential fields, coarse, medium, and fine, are generated at on-line mode from an input image. For all these three potential fields, a threshold operation is applied to the original intensity distribution of the input image and then, a Gaussian filter is applied to the threshold image I_{th}. Figure 5 shows the overview of this module

$$I_{gauss}(x,y) = \iint I_{th}(x-u, Y-v) e^{-\frac{1}{2}\frac{u^2+v^2}{\sigma^2}} dudv$$

5.2.1. Coarse level potential field

To generate the coarse level potential fields, we first apply the median filter; the median value is obtained among nine neighboring pixels, and then, is assigned to the central pixel I_{median}. This process removes isolated bright pixels. Finally, we apply an exponential function with the width k_{coarse}, to this output:

$$(x,y) = \iint I_{median}(x-u, y-v) e^{-k_{coarse}} dudv . \quad (15)$$

We prefer to the exponential function than the Gaussian function, for emphasizing the central value and suppressing peripheral areas.

5.2.2. Medium level potential field

For this level, we directly apply the exponential function to the output of the Gaussian filter. k_{medium} is selected to make this exponential function narrower:

$$I_{medium}(x,y) = \iint I_{gauss}(x-u, y-v) e^{-k_{medium}\left(u^2+v^2\right)^{\frac{1}{2}}} dudv . \quad (16)$$

5.23. Fine level potential field

The third step is deformable template matching. This step allows each point to move to further reduce the potential energy. For this step, we extract isolated brightness peaks, $I_{point}(x,y)$ using our regular point feature extractor. Then, we apply the exponential function to the binary distribution:

$$P_{fine}(x,y) = \iint I_{point}(x-u, y-v) e^{-k_{fine}\left(u^2+v^2\right)^{\frac{1}{2}}} dudv \quad (17)$$

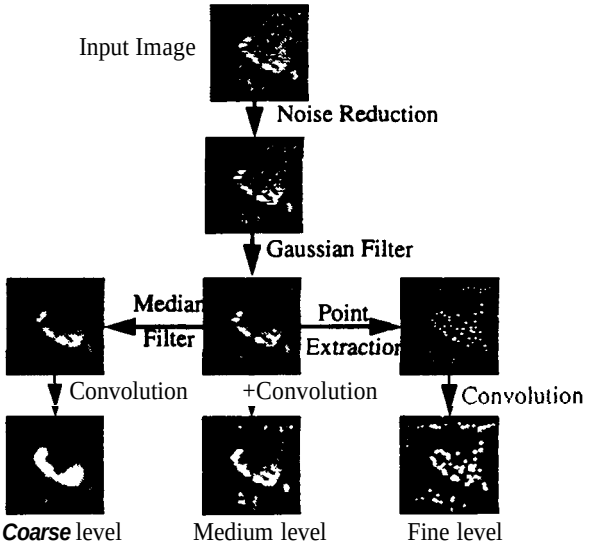


Figure 5 Potential Field Generation

6. Experiments

6.1. Outline of the system

Figure 6 shows the overview of the SAR recognition system. It has two modes: off-line and on-line mode. In off-line mode, the system generates dictionaries for targets using XPATCH SAR simulator and target models. It also generates templates for verification module. In on-line mode, the system generates an invariant histogram and potential fields from the input image. By using the invariant histogram, the indexing module selects possible candidates. Then, the final decision is made by the verification module using the potential fields and

templates.

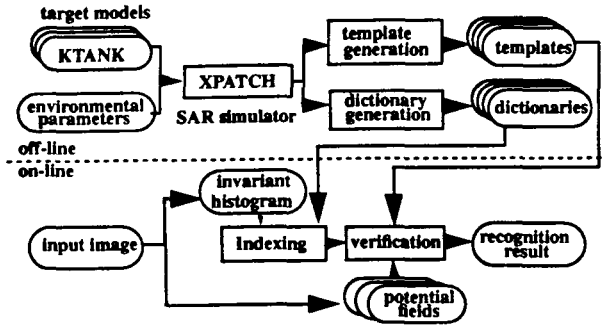


Figure 6 System overview

6.1.1. Off-line mode

In typical *SAR* image scenarios, the depression angle and resolution are fixed during image acquisition. In this paper, we use 22.5 degrees as the depression angle and 30cm/pixel as a scale. We employ the **XPATCH** *SAR* simulator, developed at Wright-Patterson Air Force Base, to generate simulated *SAR* images for the dictionary generation. Figure 7 shows three series of 36 images: KTANK, BMP, and BTR60 generated. A dictionary is constructed for 36 views rotated around the axis perpendicular to the ground plane, sampled every 10 degrees.

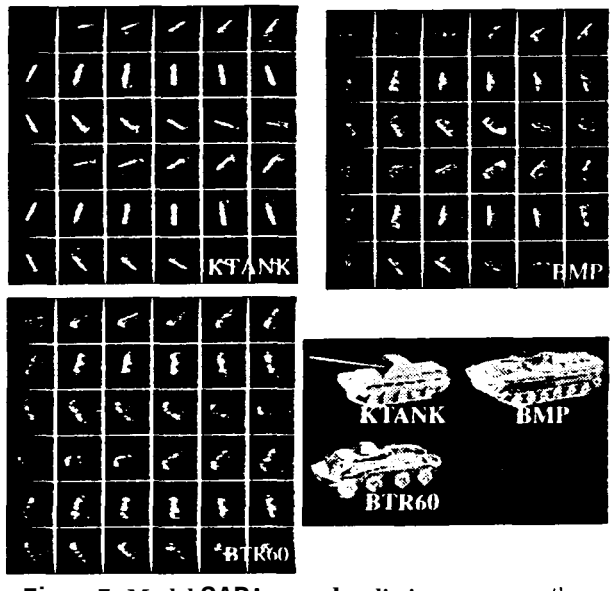


Figure 7 Model SAR images lor dictionary generation

In order to obtain an estimate of the variance of each invariant value around a representative view, 19 images around each representative views are generated within 1 degree (0.1 degree intervals). Each representative view of a dictionary consists of three invariant histograms: point-point, line-line, and point-line as shown in Figure 8.

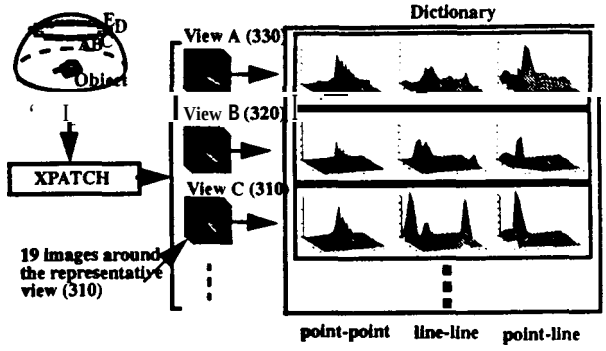


Figure 8 Generating a dictionary

In off-line mode, coarse, medium and fine templates are also generate at each representative view. For each template, eleven images around a representative view are utilized.

6.1.2. On-line model

For the recognition experiments, hybrid SAR images, synthesized *SAR* simulated signatures of target objects on real *SAR* background signatures (Lincoln Stockbridge Data), are used. The ratio of signal level between simulated and real signatures are determined using a car parked in a parking lot observed in the Stockbridge Data (M90P5F8HH). Figure 9 shows the **KTANK** model at a rotation of 312 degrees superimposed in the lawn area in the Stockbridge Data [Bessette, 1991].

Figure 9 Hybrid *SAR* image

In on-line mode, the feature detector and invariant generator are used to create the invariant histograms from an input image. The indexing module eliminates impossible candidates by measuring the distance between an input and dictionary images and prunes the number of possible objects for recognition using the similarity measure.

Our similarity measure function,

$L_{1,saturation} = \sum_{i,j} min\left(\frac{|x_{i,j} - m_{i,j}|}{\sigma_{i,j} + \sigma_o}, k\sigma_{i,j}\right)$, has two parameters: k and σ_o. We use $k = 10$ and $\sigma_o = 0.5$.

Figure 10 shows the flow of the indexing. From an input hybrid image (rotation of 312 degrees), the module generates three invariant histograms and then compares them with the dictionary. In this example, the module selects five candidates, including 110,250 and 310 (the correct pose) of **KTANK as** the possible candidates for verification.

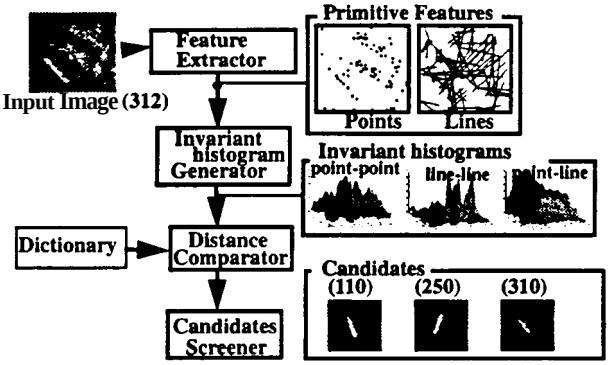


Figure 10 Indexing module

Figure 11 shows the flow of the verification module. It determines the initial template position using the feature correspondences. Then, the module evaluates each candidate pose through a three-step matching over potential energy fields given by images/features. It determines the template configuration that has the minimum deformation (deformation energy) and the best alignment of the template with features (potential energy). In this example, the module correctly identifies- the template of 310 degrees, as the minium energy template (most likely pose).

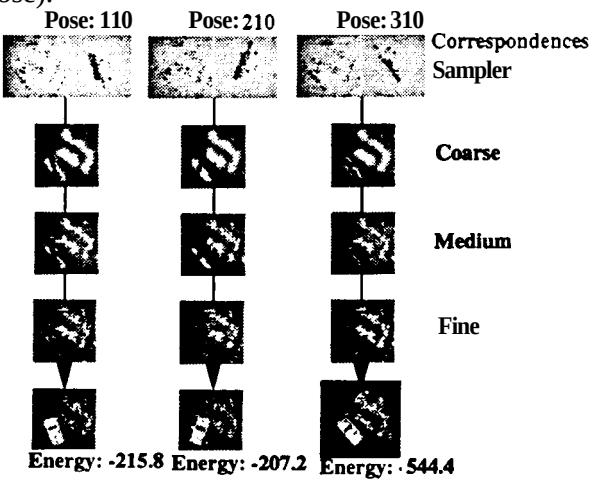


Figure 11 Verification module

Figure 12 shows the KTANK model at 310 degree aspect angle superimposed on the hybrid ***SAR*** images containing the signature from KTANK rotated 312 degree.

Figure 12 Recognition result

6.2. Recognition experiments

In order to examine the performance of our recognition system, we have generated 180 hybrid SAR images, from viewing directions sampled every 2 degrees around **360** degrees using the following models: KTANK, **BMP,** and BTR60.

We have tested the indexing module which determines the possible candidates for the verification. For each test, 180 images of the object are given to the system with the single object's dictionary being used **by** the indexing. At each viewing direction, **5** to 9 candidates are selected on average. The first row in the table denotes the results for the object. When the candidate set contains the direction nearest to the input direction, the indexing is considered **as** success. The second column in the table represents the correct indexing ratio.

Then, the verification module is executed using the templates of the candidate poses selected by the indexing module. Here candidate templates are sampled evenly ten degrees. When the template nearest to the input direction has the least energy, we consider that the correct recognition (in the fourth column) as well **as** the correct verification (in the third column) is achieved.

In case that the candidate set given by the indexing does not contain the correct direction, this is the failure of the recognition (in the fourth column), However, for the calculation of the correct verification ratio (in the third column), we discard this case.

Table 1 Recognition Results

Vehicle	Indexing	Verification	System
KTANK	*97.8%*	90.3%	88.3%

Vehicle	Indexing	Verification	System
BMP	92.8%	88.0%	81.7%
BTR60	99.4%	97.2%	96.7%

6.3. Occlusion

In order to evaluate the effect of occlusion, we generated five series of 180 images (64×64 pixels) with: 0 pixel shift, 8 pixel shift, **16** pixel shift, and 20 pixel shift. Each image is shifted a certain amount from the center position; if pixels move outside of the window (64×64) , we consider them to **be** occluded. Thus, for example, for a 20 pixel shift, roughly one half of the original pixels are lost in the worst case. We evaluate the effect using three targets, KTANK, BMP, and BTR60. Tables 2-4 shows the results, while Figures **13** summarize these results in a graphical display.

Table 2 KTANK

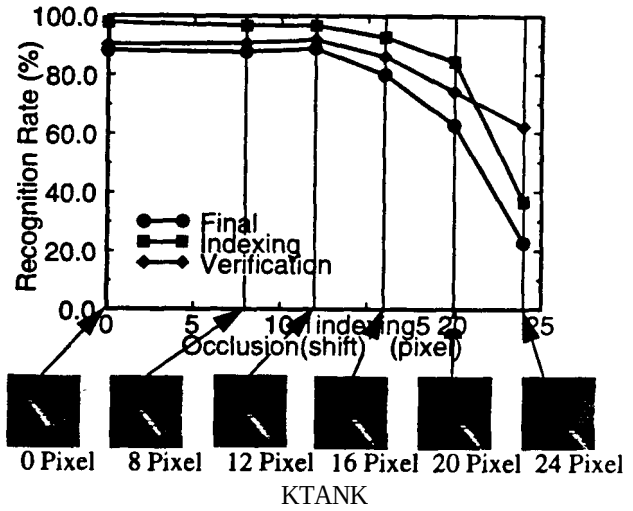
Occlusion	Indexing	Verification	System
0 pixel	97.8%	90.3%	88.3%
8 pixel	96.7%	90.8%	87.8%
12 pixel	96.7%	92.0%	88.9%
16pixel	92.8%	86.7%	80.0%
20 pixel	84.4%	74.3%	62.8%
24 pixel	36.7%	62.1%	22.8%

Table 3 BMP

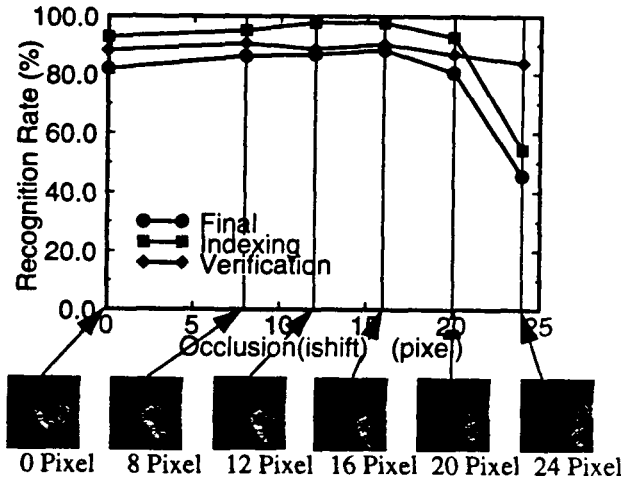
Occlusion	Indexing	Verification	System
0 pixel	92.8%	88.0%	81.7%
8 pixel	95.0%	90.6%	86.1%
I2 pixel	97.8%	88.6%	86.7%
16pixel	97.8%	90.3%	88.3%
20 pixel	92.8%	86.8%	80.6%
24 pixel	54.4%	83.7%	45.6%

Table4 BTR60

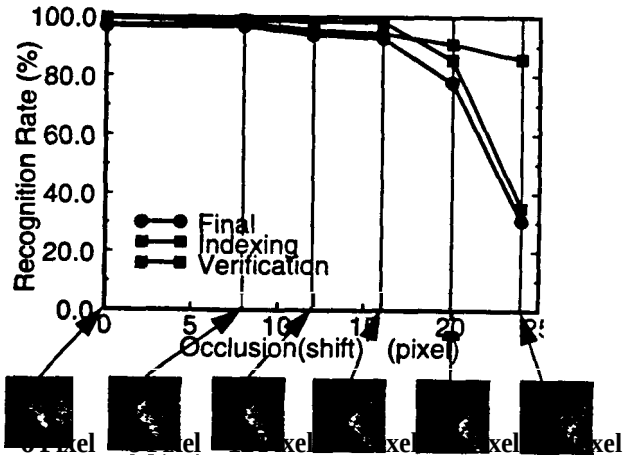
Occlusion	Indexing	Verification	System
0 pixel	99.4%	97.2%	96.7%
8 pixel	98.9%	97.8%	96.7%
12 pixel	98.3%	95.5%	93.9%
16pixel	98.3%	94.4%	92.8%
20 pixel	85.6%	90.9%	77.8%
24 pixel	35.0%	85.7%	30.6%



KTANK



BMP



BTR60

Figure 13 Occlusion

6.4. Camouflage

In order to simulate the effect of camouflage, we reduce the intensity of a target while maintaining the same background intensity. In this experiment in Tables 5-7, the effect of the camouflage is represented by the ratio of the number of features extracted from the reduced and original intensity images. See Figures 14 for the graphical display of these results.

Table 5 KTANK

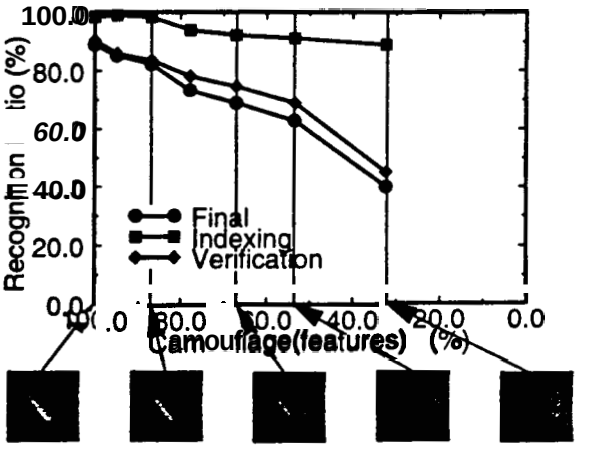
Camouflage	Indexing	Verification	System
100.0%	98.3%	90.4%	88.9%
94.8%	98.9%	86.0%	85.0%
86.8%	98.3%	83.6%	82.2%
77.7%	93.9%	78.1%	73.3%
67.0%	92.2%	74.7%	68.9%
53.4%	91.1%	68.9%	46.6%

Table 6 BMP

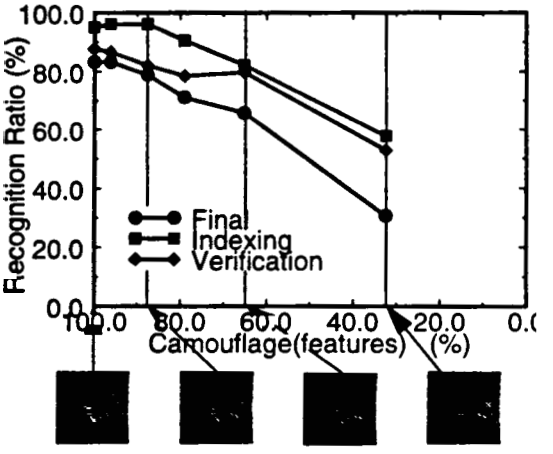
Camouflage	Indexing	Verification	System
100.0%	95.0%	87.7%	83.3%
96.2%	96.1%	86.7%	83.3%
87.6%	96.1%	82.1%	78.9%
79.0%	90.6%	78.5%	71.1%
65.1%	82.2%	79.7%	65.6%
32.4%	57.8%	52.9%	30.6%

Table 7 BTR60

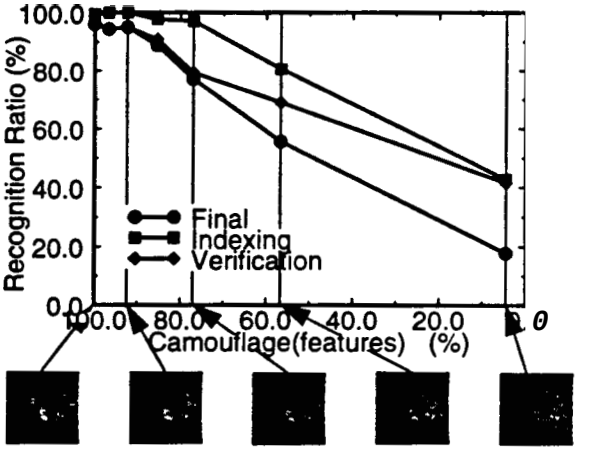
Camouflage	Indexing	Verification	System
100.0%	100.0%	96.6%	96.6%
96.7%	100.0%	94.4%	94.4%
92.3%	100.0%	95.0%	95.0%
85.4%	97.8%	90.9%	88.9%
77.7%	97.2%	79.4%	77.2%
56.6%	80.6%	69.0%	55.6%



KTANK



BMP



BTR60

Figure 14 Camouflage

6.5. Multiple object databases

We evaluated the effect of confusion among ground vehicles using BMP, BTR60 and KTANK. We generated 90 hybrid SAR images of these vehicles, at 90 viewing directions sampled every 4 degrees over 360 degrees. Here, we use a model dictionary containing BMP, BTR60 and KTANK. In the table, each row and column represents input image and system response, respectively. The number indicates the classification ratio by the system for each vehicle, while the numbers in parentheses represent the ratio of the correct pose as well as correct vehicle class. For example, the BMP was correctly identified in 90% of the tests, but only 75.6% found the correct pose.

Input\Res	BMP	BTR60	KTANK
BMP	90%(76%)	10%	0%
BTR60	7%	93%(87%)	0%
KTANK	10%	11%	79%(69%)

7. Conclusion

This paper proposes to use invariant histograms and deformable templates for SAR recognition. An invariant histogram is a histogram of geometric invariants given by primitive feature sets. Deformable template matching examines the existence of an object by superimposing templates over potential energy field generated from images so that it generates the minimum deformation (deformation energy) and the best alignment of the template with features (potential energy).

We have develop a SAR recognition system using these two techniques, and demonstrated the effectiveness of these two techniques for robust SAR recognition through extensive evaluation of the system using occluded and camouflaged target images.

This system has two modes: off-line and on-line. In off-line mode, the system generates a dictionary for indexing and deformable templates for verification. Currently, it takes a half hour for this compilation on SPARC 20. In on-line mode, by calculating an invariant histogram from an input image, the system performs the indexing to reduce the number of possible candidates. Then, from the potential fields from an input image and the deformable templates, the system determines the most likely pose and class of the target. Indexing takes about 2 to 3 seconds, and verification takes a few seconds per candidate pose. The run times include time to build invariant histograms and compute potential fields.

Recently, several researchers have begun to develop appearance-based recognition systems. From a large number of images, they effectively extract compress essential features, eigen-values in an orthogonal eigen space, and use those eigen-values for object recognition. Turk and Pentland [1991] recognized human faces using eigen-vectors and Murase and Nayar [1995] applied an eigen-space analysis for illumination planning. The main focus of these techniques are how to effectively reduce the size of necessary features for recognition.

In contrast to these compression-oriented approach, this paper proposes a redundancy oriented approach; by using a redundant representation of image features, this work shows it is possible to build a robust recognition system, in particular for SAR recognition. These characteristics are particularly important when handling occluded target images consisting of unstable non-attached SAR features.

Reference

[Ballard, 1981] Ballard, D. H., "Generalizing the Hough Transform to detect arbitrary patterns," *Pattern Recognition,* vol. 13, no. 2, pp. 111-122, 1981.

[Bessette, 1991] Bessette, L. A., "Stockbridge Mission 85 Pass 5 Data Package," *MIT Licoln Laboratory Project Report TT-80*, 1991

[Bolles and Cain, 19821 Bolles, R. C. and R. A. Cain, "Recognizing and locating partially visible objects: The local feature focus method," *Int. J. Robotics Res.,* vol.1, no. 3, pp.56-82, 1982.

[Brooks, 19831 Brooks, R., "Symbolic reasoning among 3-dimensional models and 2-dimensional images," *Artif. Intell.*, vol.5, no.5, pp.493-505, 1983.

[Chellapa et al., 1992] Chellapa, R., E. Zelnio, and E. Rignot, "Understanding synthetic aperture radar images," Proc. *IUW*, pp. 229-245, 1992.

[Clemens and Jacobs, 1991] Clemens, D. T. and D. W. Jacobs, "Model group indexing for recognition," ***IEEE Trans. PAMI***, vol.13, no.10, pp.1007-1017, 1991.

[Dhome et al., 19861 Dhome, M., M. Richetin and G. Rives, "Model-based recognition and location of local patterns in polygonal contours via hypothesis accumulation," ***Pattern Recognition in Practice II,*** E. S. Gelsema and L. N. Kanal (eds.), North-Holland, 1986.

[Gremban and Ikeuchi, 1993] Gremban, K. D. and K. Ikeuchi, "Appearance-based vision and the automatic generation of object recognition program," ***Three-dimensional object recognition*** sys***tems,*** A.K. Jain and P.J. Flynn (eds.), Elsevier Publishers, pp.229-258, 1993.

[Grimson, 1990] Grimson, W. E. L., ***Object*** recog***nition by computer,*** MIT Press, 1990.

[Huttenlocher and Ullman, 1987] Huttenlocher, D. P. and S. Ullman, "Object recognition using alignment," ***Proc. ICCV,*** pp. 102-111, 1987.

[Ikeuchi, 1987] Ikeuchi, K., "Generating an interpretation tree from a CAD model for 3d-object recognition in bin-picking tasks," ***Int. J. Comp. Vision,*** vol.1, no.2, pp.145-165, 1987.

[Ikeuchi and Kanade, 1988] Ikeuchi, ***T.*** and T. Kanade, "Toward automatic generation of object recognition programs," ***Proc. of IEEE,*** vol. 76, no.8, pp.1016-1035, 1988.

[Kuno et al., 1988] Kuno, **Y.,** K. Ikeuchi, and T. Kanade, "Model-based vision by cooperative processing of evidence and hypotheses using configuration spaces," ***Proc. SPIE,*** Vol. 938, pp.444-453, 1988.

[Lamdan and Wolfson, 1988] Lamdan, **Y** and H. J. Wolfson, "Geometric Hashing: A general and efficient model-based recognition scheme," ***Proc. ICCV,*** pp.238-249, 1988.

[Low, 1985] Lowe, D. G., ***Perceptual organization and visual recognition,*** Kluwer Academic Publishers, 1985.

[Mundy, 1992] Mundy, J. L. and A. Zisserman, edit, ***Geometric Invariance in Computer Vision,*** The MIT Press, 1992.

[Novak et. al., 19931 Novak, L.M., G.J. Owirka, and C. M. Netishen, "Performance of a high-resolution polarimetric ***SAR*** automatic target recognition system," ***Lincoln Lab Journal,*** vol.6, no. 1, pp.11-23, 1993.

[Sato et al., 1992] Sato, **K.,** ***K.*** Ikeuchi, and ***T.*** Kanade, "Model **based** recognition of specular objects using sensor models," ***CVGIP: Image Understanding,*** vol. **55,** no.2, pp.155-169, 1992.

[Stockman, 1987] Stockman, G. **S.,** "Object recognition and localization via pose clustering," ***Computer: Vision, Graphics, Image Proc.,*** vol. 40, pp.361-387, 1987.

[Tomiyasu, 1978] Tomiyasu, K., "Tutorial review of synthetic-aperture radar ***(SAR)*** with applications to imaging of the ***ocean*** surface," ***Proc. of IEEE,*** vol. 66, no.5, pp. 563-583, 1978.

[Turk and Pentland, 1991] Turk, M. A. and Pentland, **A.** P., "Face recognition using eigenfaces," ***Proc. CVPR,*** pp. 586-591, 1991.

[Murase and Nayar, 1995] Murase, H. and S. K. Nayar, "Visual learning and recognition of 3-D objects **from** appearance," ***Int. J. Computer Vision,*** vol. 14, no. 1, pp. 5-24, 1995.

[Waxman et al., 1993] Waxman, **A.** M., M. Seibert, A.M. Bemardon, and D. **A.** Fay, "Neural systems for automatic target learning and recognition," ***Lincoln Lab Journal,*** vol. 6, no. 1, pp.77-116, 1993.

[Wheeler and Ikeuchi, 1995] Wheeler, M. and K. Ikeuchi, "Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition," ***IEEE Trans. PAMI,*** vol. 17, no.3, pp.252-265, 1995.

[Wolfson and Lamdan, 1992] Wolfson, **H.** J. and Y. Lamdan, "Transformation invariant indexing," ***Geometric Invariance in Computer Vision,*** J. Mundy and A.Zisserman (eds.), MIT Press, pp.335-353, 1992.