PERGAMON

Pattern Recognition 36 (2003) 2855–2873

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Stochastic models for recognition of occluded targets

Bir Bhanu*, Yingqiang Lin

Center for Research in Intelligent Systems, College of Engineering, University of California, Riverside, CA 92521, USA

Received 18 February 2002; accepted 20 May 2003

Abstract

Recognition of occluded objects in synthetic aperture radar (SAR) images is a significant problem for automatic target recognition. Stochastic models provide some attractive features for pattern matching and recognition under partial occlusion and noise. In this paper, we present a hidden Markov modeling based approach for recognizing objects in SAR images. We identify the peculiar characteristics of SAR sensors and using these characteristics we develop feature based multiple models for a given SAR image of an object. The models exploiting the relative geometry of feature locations *or* the amplitude of SAR radar return are based on sequentialization of scattering centers extracted from SAR images. In order to improve performance we integrate these models synergistically using their probabilistic estimates for recognition of a particular target at a specific azimuth. Experimental results are presented using both synthetic and real SAR images.

Keywords: Hidden Markov modeling; Object recognition; Multiple recognition models and their integration; Rotation invariance; Synthetic aperture radar images

1. Introduction

One of the critical problems for object recognition is that the recognition approach should be able to handle partial occlusion of an object and spurious or noisy data [1–4]. In most of the object recognition approaches, the spatial arrangement of structural information of an object is the crucial part with the most important information. Under partial occlusion situations the recognition process must be able to work with only portions of the *correct* spatial information. Rigid template matching and shape-based recognition approaches depend on good prior segmentation results. But the structural primitive (e.g., line segments, point-like features, etc.) extracted from occluded and noisy images may not have sufficient reliability, which will directly undermine the performance of those recognition approaches.

We suggest an object recognition mechanism that effectively makes use of all available structural information. Based on the nature of the problems caused by occlusion and noise, we view the spatial arrangement of structural information as a whole rather than view the spatial primitives individually. Because of its stochastic nature, a hidden Markov model (HMM) is quite suitable for characterizing patterns. Its nondeterministic model structure makes it capable of collecting useful information from distorted or partially unreliable patterns. Many applications of HMM in speech recognition [5–7] and character recognition [8,9] attest to its usefulness. Thus, it is potentially an effective tool to recognize objects with partial occlusion and noise.

However, the limit of traditional HMMs is that they are basically one-dimensional models. So how to appropriately apply this approach to two-dimensional image problems becomes the key. It has been largely an unsolved problem. In this paper we use the features based on the image formation process to encode the 2-D image into 1-D sequences. We use information from *both* the relative positions of the scattering centers and their relative magnitude in synthetic aperture radar (SAR) images to address the fundamental issues of building object models and using them for robust recognition of objects in SAR images.

[*] Corresponding author. Tel.: +1-909-787-3954; fax: +1-909-787-3188.

E-mail address: bhanu@cris.ucr.edu (B. Bhanu).

0031-3203/03/$30.00 © 2003 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/S0031-3203(03)00182-1

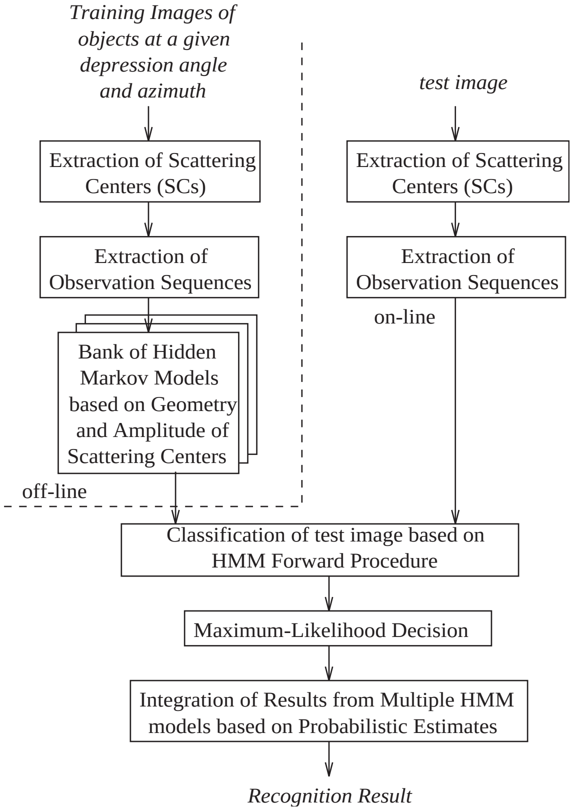

Training Images of objects at a given depression angle and azimuth
test image
Extraction of Scattering Centers (SCs)
Extraction of Scattering Centers (SCs)
Extraction of Observation Sequences
Extraction of Observation Sequences
on-line
Bank of Hidden Markov Models based on Geometry and Amplitude of Scattering Centers
off-line
Classification of test image based on HMM Forward Procedure
Maximum-Likelihood Decision
Integration of Results from Multiple HMM models based on Probabilistic Estimates
Recognition Result

Fig. 1. The HMM-based approach for recognition of occluded objects.

1.1. Overview of the approach

Fig. 1 provides an overview of the HMM-based approach for recognition of occluded objects in SAR imagery. During an off-line phase, scattering centers are extracted from SAR images by finding local maxima of intensity. Both locations and magnitudes of these peak features, viewed as *emitting patterns*, are used to generate multiple observation sequences. These multiple observation sequences, based on both the *relative* geometry and amplitude of SAR return signal (obtained as a result of the physics of the SAR image formation process), are used to build the bank of stochastic models. From each training SAR image containing a particular object under a particular azimuth angle, five kinds of observation sequences are extracted to build five HMM models with one model taking one kind of observation sequence as input. These models provide robust recognition in the presence of occlusion and unstable features caused by scintillation phenomena where some of the features may appear/disappear at random in an image. At the end of the off-line phase, hidden Markov recognition models for various objects and azimuths are obtained. Similar to the off-line phase, during the on-line phase features are extracted from SAR images and observation sequences based on these features are matched by the HMM forward process with the stored models obtained previously. Then Maximum likelihood decision is made on the classification results. Finally the results obtained from multiple models are combined in a voting kind of approach that uses both the object, azimuth label and its probability of classification. This produces a rank ordered list of classifications of the test image and associated confidences.

2. Related work and our contributions

In this paper, the features used in recognition are scattering centers, which are high magnitude points extracted from SAR images. During the training, the magnitude and relative positions of scattering centers are used to build HMM models; during testing, the scattering centers extracted from the testing image are fed into each HMM model and the model with the maximum probability of match is selected as the recognition result. It is actually a point matching problem in which the magnitude and geometric distribution of scattering centers in the testing image are matched with those in the training images via HMM models.

2.1. Point matching

Point matching is widely used in object recognition, motion analysis, image retrieval and image registration [10–12]. In the following, we provide a brief review of point matching methods, representative techniques are summarized in Table 1.

- *Local-feature-focus*: Each object contains a small number of focus (salient) features. The correspondence between focus features of the model and testing images is established to determine the affine transformation to map the model into the image. After the transformation is obtained, the minor features around each of the focus features are used to verify the transformation and evaluate the quality of the match.
- *Pose clustering*: This approach computes an affine alignment for all possible feature point pairs and then checks a cluster of similar parameter sets. If the feature points from a model and an image can be matched, a cluster exists in the parameter space. Considering each possible point pair, the number of pairs could be very large, leading to a significant computational cost.
- *Geometric hashing*: It is designed to work with many models and it considers a large amount of off-line preprocessing and a large amount of space for a potentially fast online recognition. The off-line preprocessing creates a hash table indexed by a pair of affine coordinates of point features with respect to a particular basis. Each cell of the hash table stores a list of model-basis pairs where some point of the model in the list has affine coordinates with respect to the basis. The recognition step creates an

Table 1
Summary of representative point matching techniques

Techniques	References	Comments
Local-feature-focus	Shen and Palmer [27]	Feature focus: junctions, a matching algorithm matches junctions between two images.
Pose clustering	Stockman [28]	Uses pose clustering in both 2D and 3D recognition.
	Bhanu and Ming [29]	Uses clusters of transformations to recognize occluded Objects.
	Moss and Hancock [30]	Primitive parts determine a set of alignment parameters; Global alignment is determined by the maximum vote.
	Olson [31]	Recognizes object by finding clusters of poses from a space of legal object positions using 2D feature points.
Geometric hashing	Lamden and Wolfson [32]	Represent object as sets of points or lines; matching procedure is based on geometric hashing technique.
	Jones and Bhanu [19]	Scattering centers are sorted by their magnitude; hash table, indexed by the range and cross range distances between feature points, is queried to accumulate the evidence.
	Surendro and Anzai [33]	Calculates the statistics of point features to determine possible matches; hypothesize the existence of a model if its features score hits the vote count.
	Au and Tsang [34]	Geometric hashing are used to recognize object under partial occlusion; salient points are features used to form hash table that is used to record the vote.
Hausdorff measure	Olson and Huttenlocher [35]	Feature points are edge pixels; both the location and orientation of edge pixel are used to measure the Hausdorff distance.
Interpretation tree	Albus [36]	Used for matching 3D model features to 2D image features; tolerant to incomplete features; makes optimal use of geometric constraints.
Discrete relaxation	Tu and Dubuison [37]	Detect high contrast points and find point matches between images; after a few iterations, point pattern and good corresponding points are obtained.
	Rangarajan and Chui [38]	Solve the point matching problem with deterministic annealing to avoid local minima.
Continuous relaxation	Bhanu and Faugeras [39]	Shape matching is viewed as segment matching where a segment is formed by two or three feature points; the technique is based on stochastic relaxation.
	Chen et al. [40]	Find motion vectors of features for object tracking.
Relational distance matching	Sharghi et al. [41]	Matches straight lines; triplets of matched points are used to construct a model polygon in one image to search a matching polygon in the other.
	Myers et al. [42]	Describe a framework to match corrupted relational graph; develop edit distance for graph matching.

accumulator array indexed by model-basis pairs. Each cell of the array is incremented by 1 if the model-basis indexing the accumulator cell is in the list associated with a cell of the hash table, where the hash table cell is indexed by the affine coordinates of an image feature point with respect to an affine basis. The accumulator is used to vote for the hypothesis that there is a particular affine transformation placing the model in the image. The model-basis pairs that achieve a high number of votes are selected as possible transformations.

- *Hausdorff measure*: The Hausdorff measure yields the maximum distance of a model feature point (for example, edge pixels) from its nearest image feature point. It is used to determine the quality of a match between the transformed feature points of a model and those of an image and thus, it determines the goodness of the transformation mapping a model to an image. To address the occlusion problem, a partial Hausdorff measure is proposed to determine the Hausdorff measure among part of model feature points (salient feature points that are expected to be found in the image) from its nearest image feature point.
- *Interpretation tree and graph matching*: An interpretation tree represents all possible matches between a model and an image, where both are described by using feature points. Every path in the tree is terminated either because it represents a complete consistent match or because the partial match fails some relation. Graph matching is a classical relational matching algorithm used in object recognition. After feature points are extracted from a model and an image, two graphs representing the model and the image, respectively, are built in which the nodes of graph represent the feature points and the edges of graph represent the relations between feature points. Thus, matching between model and image becomes matching between two graphs representing them. To deal with partial occlusion, subgraph matching is proposed to find the matching between the image and a portion of a model. The maximal clique or minimum spanning tree algorithms that match between two graphs are also used to reduce computation.
- *Discrete relaxation*: Unlike an interpretation tree, discrete relaxation uses only local constraints rather than all the constraints available. Initially, an image feature can match with any model feature permitted by its type. Then it examines the relations between features of model and image and reduces possible matches. Local constraints from one image feature neighborhood can propagate across the object to another image feature on a path N edges distant after N iterations. Local constraints are weaker than those used in an interpretation tree, but they can be applied in parallel, thus making faster and simpler processing. Since only local constraints are considered, it is more robust in case of partial occlusion of the object in the image.
- *Continuous relaxation*: In discrete relaxation, a feature point in an image is either matched or not matched with a model feature point. Continuous relaxation associates each image-model feature point pair to a real number representing the probability that the image feature and model feature can be matched. Discrete relaxation iterates to remove possible matches, continuous relaxation iterates to update the probabilities associated with each possible match.
- *Relational distance matching*: Due to feature extraction errors, noise and occlusion of the object, an image may have missing or extra features and required relationships may not hold. Continuous relaxation may be used, but it may not find the good solution. Relational distance matching determines the relations between features to obtain a relational description of an image or object and performs a search to find the best mapping from a model to an image in the sense that it preserves most of the relationships. The goodness of a matching is evaluated by relational distances between the transformed model and the image.

In this paper, HMM is used as a novel way to perform point matching and it is different from all the above methods. In the local-feature-focus, graph matching, interpretation tree, discrete and continuous relaxation and relational distance matching methods, correspondence between salient feature points must be established to determine the affine transformation or obtain initial matching. Pose clustering considers every possible match between model and image feature points and geometric hashing computes the affine coordinates of each feature point with respect to different coordinate systems. Geometric hashing also needs significant preprocessing and space. However, in the HMM method, the correspondence between model and image feature points is not obtained directly. The magnitude and distribution of feature points are encoded in HMM models and the characteristics of the feature points in the testing image are reflected by the probability of a HMM model generating the observation sequence obtained from the feature points. The observation sequence is a sequence of symbols and the HMM model is used to compare the observation sequences from the training images with those from testing images. It is essentially a string matching problem discussed in the next subsection.

2.2. String matching

String matching algorithms compare strings and determine the similarity or dissimilarity between them [13]. This is nontrivial since, in general, the strings can be of different lengths. The key is to define dissimilarity and there are many definitions. The Levensthein distance defines the dissimilarity as the number or cost of string edit operations (insertion, substitution and deletion of symbols) needed to convert a string into a reference string. Structural dissimilarity measures the extent to which two strings differ in terms of rules of composition of the string pattern. Error-correcting parsing defines the distance between a string and a reference string as the weighted Levensthein distance between the string and the nearest (in term of edit operations) string generated by the grammar inferred from the reference string. Grammar complexity-based dissimilarity is formed on the assumption that the joint description of two similar strings is more compact than their isolated description due to the sharing of rules of symbol composition and quantifies the compactness of the representation by the grammar complexity. Minimum code length-based similarity explores the compressibility of strings. Since low compressibility is related with high complexity and randomness and since strings with inter-symbol dependencies are likely to have compact description, strings sharing subpattern regularities will, therefore, produce more compact codes than the gathering of the codes for the individual strings. The quantification of this reduction in code length form the basis of similarity measure. Strings can also be represented by a model such as a Markov chain or a HMM. Thus, similarity between strings are computed based on their model representation. Fielding and Ruck [14] have used HMM models for spatio-temporal pattern recognition to classify moving objects in image sequences. Rao and Mersereau [15] have attempted to merge HMM and deformable template approaches for image segmentation. Kottle et al. propose a HMM-Based SAR automatic target recognition system [16]. They first segment a SAR image, and then extract features followed by Radon transforms. The feature sequences so obtained are input to HMMs.

In this paper, HMM is used to capture the characteristics of the observation sequences from the training images and the observation sequences from the testing images are compared with HMM models. Template matching [17] and major axis based approaches [18] have been used to recognize and index objects in SAR images, however, they are not suitable to recognize occluded objects. Our HMM-based approach to recognizing occluded objects in SAR images uses significantly fewer models (typically 5) of an object at some aspect than our geometric hashing approach [19], which uses 20–40 models with different scatterers as basis points.

2.3. Our contributions

The contributions of this paper are: (a) HMM approach commonly used for recognizing 1-D speech signals is applied in a novel manner to 2-D SAR images to solve the occluded object recognition problem. (b) Unlike most of the work for model building in pattern recognition and computer vision, our recognition models using HMM concepts are based on the peculiar characteristics of SAR images where the number of models used for recognition is justified by the quantification of the azimuthal variance in SAR images. (c) Multiple models derived from various observation sequences, based on both the relative geometry and signal amplitude (four based on geometry and one based on amplitude) are used to capture the unique characteristics of patterns to recognize objects. (d) Extensive amounts of data are used to test the approach for recognition of objects with various amounts of occlusion for both synthetic and real data.

3. Hidden Markov modeling approach

It is well known that HMM can model speech signals well [5–7]. It is a model used to describe a doubly stochastic process which has a set of states, a set of output symbols and a set of transitions. Each transition is from state to state and associated with it are a probability and an output symbol. The word 'hidden' means that although we observe an output symbol, we cannot determine which transition has actually taken place. At each time step t, the state of the HMM will change according to a transition probability distribution which depends on the previous state and an observation y_t is produced according to a probability distribution which depends on the current state. Formally, a HMM is defined as a triple $\lambda=(A,B,\pi)$, where a_{ij} is the probability that state i transits to state j, $b_{ij}(k)$ is the probability that we observe symbol k in a transition from state i to state j, and π_i is the probability of i being the initial state. Fig. 2 shows an example of a N states HMM.

Recognition problem—forward procedure: The HMM provides us a useful mechanism to solve the problems we face for robust object recognition. Given a model and a sequence of observations, the probability that the observed sequence was produced by the model can be computed by the forward procedure [20]. Suppose we have a HMM $\lambda = \{A, B, \pi\}$ and an observation sequence $y_1^T = \{y_1, y_2, \ldots, y_t, \ldots, y_T\}$. We define $\alpha_i(t)$ as the probability that the Markov process is in state i, having generated $y_1^t = \{y_1, y_2, \ldots, y_t\}$.

$$\alpha_i(t) = 0 \quad \text{when } t = 0 \text{ and } i \text{ is not an initial state.}$$

$$\alpha_i(t) = 1 \quad \text{when } t = 0 \text{ and } i \text{ is an initial state.}$$

$$\alpha_i(t) = \Sigma_j[\alpha_j(t-1)a_{ji}b_{ji}(y_t)] \quad \text{when } t > 0. \tag{1}$$

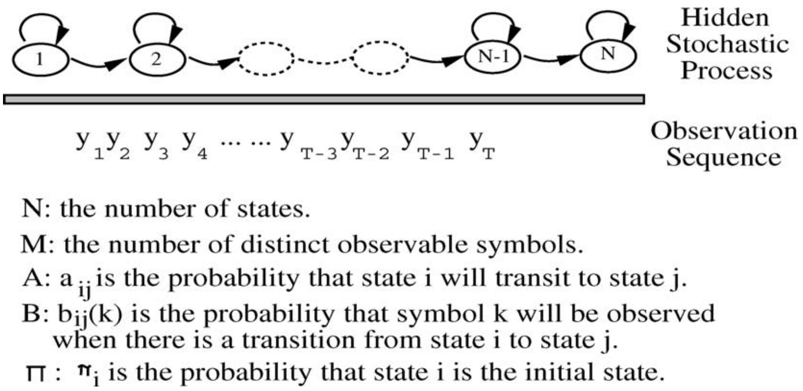
The probability that the HMM stopped at the final state and generated y_1^T is $\alpha_{S_F}(T)$. The forward procedure is given below:

Let T be the length of an observation sequence and N is the number of states in the HMM.

1. Initialize $\alpha_i(0)$, where $i = 1, 2, \ldots, N$.
2. Compute $\alpha_i(t)$ inductively (Eq. (1)), where $t=1, 2, \ldots, T$. At each step, the previously computed $\alpha_i(t-1)$ is used. Repeat this process until t reaches T.
3. Output $\alpha_{S_F}(T)$, where $\alpha_{S_F}(T)$ is the probability that the HMM stopped at the final state and generated the observation sequence.

Usually, α becomes too small to be represented in a computer after several iterations. We take the logarithm of the α value in the computation.

Training problem—Baum–Welch algorithm: To build a HMM is actually an optimization of the model parameters so that it can describe the observation better. This is a problem of training. The Baum–Welch re-estimation algorithm is used to calculate the maximum likelihood model. But before we use the Baum–Welch algorithm, we must introduce the counterpart of $\alpha_i(t)$; $\beta_i(t)$, which is the probability that the Markov process is in state i and will generate


1
2
N-1
N
Hidden Stochastic Process
Observation Sequence
y 1 y 2 y 3 y 4 ... ... y T-3 y T-2 y T-1 y T

N: the number of states.
M: the number of distinct observable symbols.
A: a_{ij} is the probability that state i will transit to state j.
B: $b_{ij}(k)$ is the probability that symbol k will be observed when there is a transition from state i to state j.
Π : π_i is the probability that state i is the initial state.

Fig. 2. A N states forward-type HMM.

$y_{t+1}^T = \{y_{t+1}, y_{t+2}, \dots, y_T\}$.

$\beta_i(t) = 0$ when $t = T$ and i is not a final state.

$\beta_i(t) = 1$ when $t = T$ and i is a final state.

$$\beta_i(t) = \Sigma_j[a_{ij}b_{ij}(y_{t+1})\beta_j(t+1)] \quad \text{when } 0 \leqslant t < T. \tag{2}$$

The probability of being in state i at time t and state j at time $t+1$ given observation sequence y_1^T and the model λ is defined as follows:

$$\gamma_{ij}(t) = P(X_t = i, X_{i+1} = j \mid y_1^T) = \frac{\alpha_i(t-1)a_{ij}b_{ij}(y_t)\beta_j(t)}{\alpha_{S_F}(T)}. \tag{3}$$

Now the expected number of transitions from state i to state j given y_1^T at any time is simply $\Sigma_{t=1}^T \gamma_{ij}(t)$ and the expected number of transitions from state i to any state at any time is $\Sigma_{t=1}^T \Sigma_k \gamma_{ik}(t)$. Then, given some initial parameters, we could recompute $\overline{a_{ij}}$, the probability of taking the transition from state i to state j as

$$\overline{a_{ij}} = \frac{\Sigma_{t=1}^T \gamma_{ij}(t)}{\Sigma_{t=1}^T \Sigma_k \gamma_{ik}(t)}. \tag{4}$$

Similarly, $\overline{b_{ij}(k)}$ can be re-estimated as the ratio between the frequency that symbol k is emitted and the frequency that any symbol is emitted:

$$\overline{b_{ij}(k)} = \frac{\Sigma_{t:y_t=k} \gamma_{ij}(t)}{\Sigma_{t=1}^T \gamma_{ij}(t)}. \tag{5}$$

It can be proved that the above equations are guaranteed to increase $\alpha_{S_F}(T)$ until a critical point is reached, after which the re-estimate will remain the same. In practice, we set a threshold as the ending condition for re-estimation.

So the whole process of training a HMM is as follows:

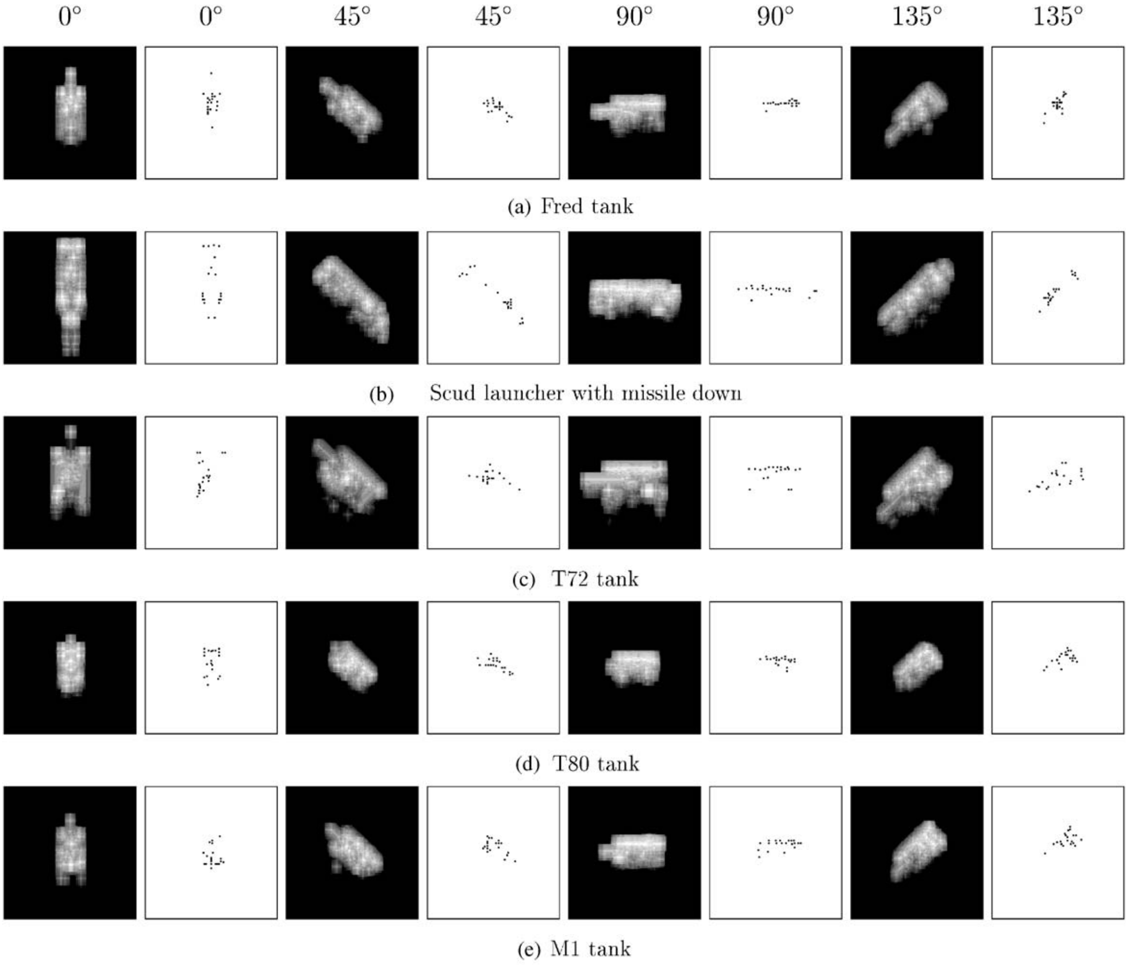
1. Initially, we have only an observation sequence y_1^T and blindly set (A, B, π).
2. Use y_1^T and (A, B, π) to compute α and β (Eqs. (1) and (2)).
3. Use α and β to compute γ (Eq. (3)).
4. Use y_1^T, (A, B, π), α, β and γ to compute A and B (Eqs. (4) and (5)). Go to *step* 2.

A HMM is able to handle pattern distortions and the uncertainity of locally observed signals, because of its non-deterministic nature. However, a HMM is primarily suited for sequential, one-dimensional patterns and it is not obvious that how a HMM can be used on 2-D patterns in object recognition. The basic ideas to apply a HMM for our purpose are (a) training the HMM λ by samples of SAR images of a certain object, and (b) recognizing an unknown object in a given SAR image. These two problems are addressed in the following. The key questions are what we shall use as observation data and how we get the observation sequences.

4. Hidden Markov models for SAR object recognition

4.1. Extraction of scattering centers

Scattering centers (location and magnitude) extracted from SAR images are used to train and test models for recognition. At 6–12 in. resolution, there exist a large number of peaks corresponding to scattering centers. We select peaks as features in this work since we wanted to evaluate the limits of our approach using six inch resolution synthetic SAR images (generated using the XPATCH code [21]) and real SAR images at one foot resolution from the MSTAR data [22] before more complicated features are used. A pixel is a scattering center if the magnitude of SAR return at this pixel is larger than all its eight neighbors. Fig. 3 shows some examples of top 20 scattering centers (based on their magnitude) extracted from XPATCH generated SAR images of five objects at 15° depression angle and azimuths at 0°, 45°, 90°, and 135°. The squint angle



(a) Fred tank

(b) Scud launcher with missile down

(c) T72 tank

(d) T80 tank

(e) M1 tank

Fig. 3. XPATCH SAR images of five targets and the corresponding scattering centers.

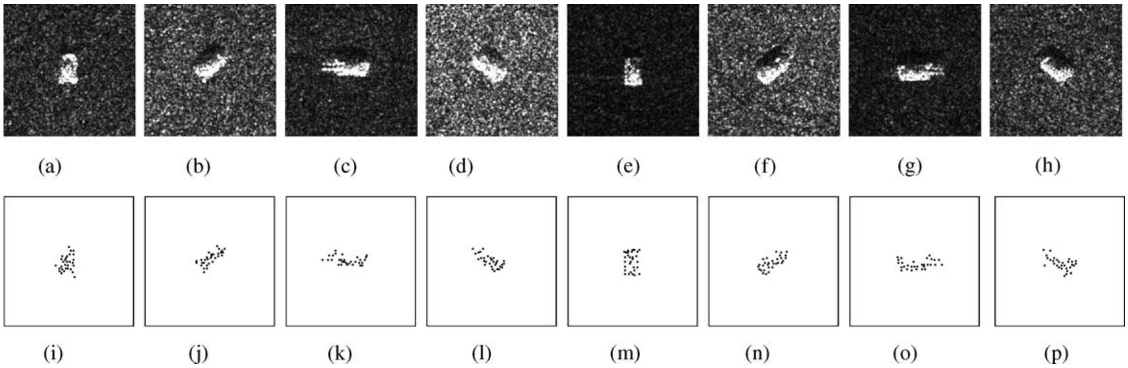


Fig. 4. Real SAR images and extracted scattering centers of T72 tank #a64: (a) 2°; (b) 45°; (c) 93°; (d) 135°; (e) 180°; (f) 226°; (g) 269°; (h) 315°; (i) 2°; (j) 45°; (k) 93°; (l) 135°; (m) 180°; (n) 226°; (o) 269°; and (p) 315°.

between the flight path and radar beam is 90° and kept fixed for all the images used in this paper. Figs. 4 and 5 show examples of scattering centers (top 30) extracted from real MSTAR SAR data. The real SAR images are obtained for depression angle 30° and their corresponding azimuths are shown in the figure captions.

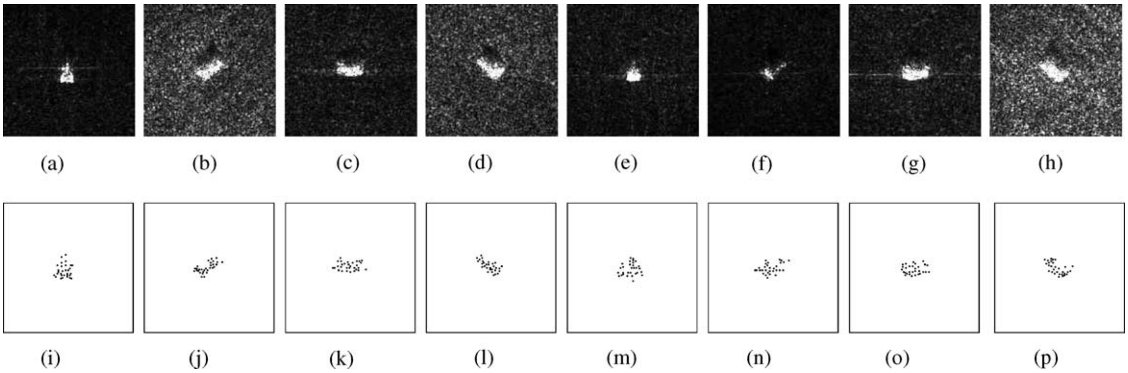


Fig. 5. Real SAR images and extracted scattering centers of ZSU anti-aircraft gun #d08: (a) 3°; (b) 45°; (c) 93°; (d) 135°; (e) 180°; (f) 226°; (g) 269°; (h) 315°; (i) 3°; (j) 45°; (k) 93°; (l) 135°; (m) 180°; (n) 226°; (o) 269°; and (p) 315°.

4.2. Rotation variance of scattering centers and representation of 3-D objects

In order to determine the number of azimuths needed to represent a 3-D object in a systematic manner, we evaluate the characteristics of SAR images. Unlike the visible images, SAR images are extremely sensitive to slight changes in viewpoint (azimuth and depression angle) and are not affected by scale [23]. We evaluate [19] the characteristics of scattering centers to find out what kind of location invariance exists among scattering centers. Fig. 6 shows the rotation invariance of the T72 tank with XPATCH data. The data is obtained by rotating the image at azimuth i° (for a fixed depression angle) by x° (x from 1 to 10), and comparing the rotated image with the image of $(i+x)^\circ$ to see how many scattering centers do not change their location. Since the object chip is 256×256 pixels, we rotate the image with respect to the center point $(127.5, 127.5)$. The distance measurement criteria "exact match" and "within one pixel" are defined in the following:

$\mathbf{x_r}$ **exactly matches x** if

$$MAX(|x - x_r|, |y - y_r|) < \tfrac{1}{2} \text{ pixel},$$

$\mathbf{x_r}$ **and x are within one pixel**

$$\text{if } MAX(|x - x_r|, |y - y_r|) < 1\tfrac{1}{2} \text{ pixel}.$$

Fig. 6(a) shows the average (over all the 360 azimuth angles) results for the number of scattering centers that remain unchanged within some error tolerance (as defined in the above) for T72 tank images for angular span of 1–10°. The 50 strongest scattering centers are used for each image. Note that for a 1° change in azimuth (angular span), approximately 1/3 and 3/4 scattering centers remain unchanged for the exact match case and within one pixel case, respectively. Fig. 6(b) gives the percentage of scattering center locations unchanged vs. azimuth angle, for each of the 360 azimuths, with 1° angular span for the exact match and within one pixel match. These results show that scattering centers for SAR images vary greatly with relatively small changes of azimuth angles. Similar results are obtained with real SAR data at one foot resolution [24]. Therefore, to recognize occluded objects, we need SAR images of an object at a given depression angle under various finely sampled azimuth angles. Ideally, we can have 360 SAR images of an object with one image corresponding to one and only one azimuth angle between 0° and 359°. We treat an object under different azimuth angles separately and build a model for each azimuth angle based on the SAR image taken under this azimuth angle whenever possible. When building models for an object azimuth, a single HMM model is inadequate because of noise, articulation, occlusion, etc. To increase robustness, we build multiple HMM models for a given object at a specific azimuth with one model taking a particular kind of observation sequence as input, as discussed in the next subsection.

4.3. Extraction of observation sequences

After the scattering centers are extracted, we need to encode the data into a 1-D sequence as the input to a model based HMM process for recognition. It is one of the *key* factors which affects the performance of HMM approach for object recognition. There are many ways to choose observation sequences, but we want to use information from both the magnitude and the relative spatial location of the scattering centers extracted from a SAR image. Also the sequentialization method should not be significantly affected by distortion, noise, or partial occlusion and should be able represent the image efficiently. Based on the above considerations, we employ two approaches to obtain the sequences.

- Sequences based on amplitudes: $O_1 = \{Magnitude_1, Magnitude_2, \ldots, Magnitude_n\}$, where $Magnitude_i$ is the amplitude of ith scattering center and $Magnitude_i \geqslant Magnitude_{i+1}$, $\forall\, i = 1, \ldots, n-1$.

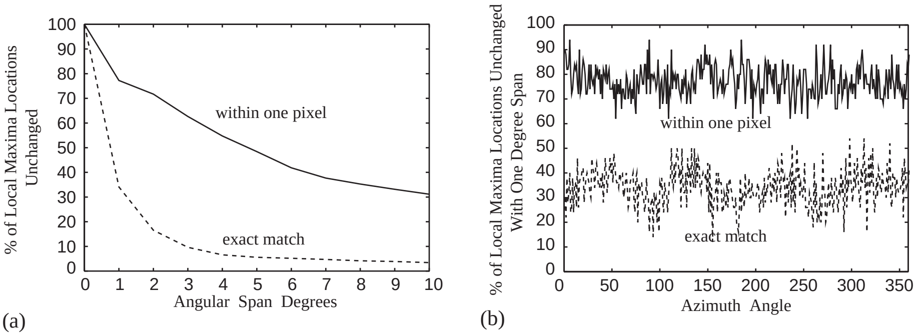


Fig. 6. XPATCH Data: (a)T72 tank rotational invariance, and (b) T72 tank rotational invariance with 1° angular span.

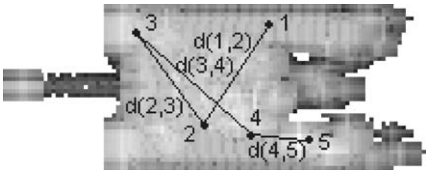


Fig. 7. Example of an observation sequence superimposed on an image of T72 tank.

- Sequences based on *relative* geometrical relationships:
  $O_2 = \{d(1,2), d(2,3), \dots, d(n,1)\}$ $(length\ n)$,
  $O_3 = \{d(1,2), d(1,3), \dots, d(1,n)\}$ $(length\ n-1)$,
  $O_4 = \{d(2,1), d(2,3), \dots, d(2,n)\}$ $(length\ n-1)$,
  $O_5 = \{d(3,1), d(3,2), \dots, d(3,n)\}$ $(length\ n-1)$,
  where $d(i,j)$ is the Euclidean distance between scattering centers i and j.

Fig. 7 gives an example to illustrate how we get the sequences. Sequence O_1 is obtained by sorting the scattering centers by their magnitude. We label the scattering centers 1 through n in descending order. So in this approach, we do not use the location information and thus can avoid the instability caused by the error in localization of scattering centers. Sequences O_2 through O_5 are obtained based on the relative locations of the scattering centers and magnitude of the scattering centers only affects the ordering. Note that they are not scale dependent, since the SAR images are scale independent [23]. In experiments described in Section 5, we only consider the top 20 scattering centers for XPATCH data and the top 30 scattering centers for MSTAR data (sorted in descending order of their magnitude). This method of ordering is done for convenience, for repeatability and because we expect that the scattering centers with larger magnitude are generally more stable than the weaker ones.

Since we use discrete HMMs, each element in the sequence should be converted to an observation symbol. Labels from 1 to M represent the symbols which can be observed for a HMM. We use the K-means algorithm [25] to classify the magnitude values (or distance values) of all the scattering centers in the database into K classes (here $K = M$, it is experimentally determined, see Section 5.1.1). Once we know to which class each of the elements of a sequence belongs, we label the element with the label of its class. Thus, for a given sequence, we obtain a sequence of observation symbols.

4.4. Off-line training and on-line recognition phases

The procedure for building the model based on a particular kind of observation sequence is described as follows:

1. Loop (for a given depression angle) lines 2–4 for each object and each azimuth angle.
2. Generate images which simulate occlusion with scattering centers occluded from different directions (see Section 5).
3. Loop line 4 for each image generated by line 2.
4. Run Baum–Welch algorithm on all the observation sequences generated in (2) to estimate the HMM parameters. First, randomly set the parameter values and run Baum–Welch algorithm on the first observation sequence (use Eqs. (1)–(5)). Then the parameter values resulting from the first observation sequence are used as the initial parameter values when the second observation sequence is input and so on. Exit the loop when there is no further change in parameter values.

The recognition procedure for a particular kind of observation sequence, is given below:

1. Extract the observation sequence from the testing image.
2. Loop line 3 for all the models in the model base.
3. Feed the observation sequence into the model, $(A,B,\Pi)_{(M_i^*,a_j^*)}$, where (M_i^*,a_j^*) represents the model of object i with azimuth angle j. Use Forward algorithm to compute the probability that this sequence is produced by this model.
4. The models with the H highest probabilities of generating the observation sequence are selected as the best

possible matches to be used by the next integration step. The model with maximum probability is considered the best match.

4.5. Integration of results from multiple sequences

Since not all models based on various sequences for a particular object and azimuth will provide optimal recognition performance under occlusion, noise, etc., we improve the recognition performance by combining the results obtained from multiple kinds of models. Thus, when testing image cannot be recognized correctly by a model based on a particular sequence, say O_1, it can be recognized correctly by models based on other kinds of sequences. We have developed a voting-like method shown in Fig. 8 to integrate the results from models based on a given number of sequences. The algorithmic steps are:

1. For each kind of observation sequence, we collect the H highest possibilities in the test results. Each possibility is the probability that the test image is the image of that object at that azimuth.
2. A normalization is done to the H probabilistic estimates for each kind of observation sequence so that estimates from five kinds of observation sequences can be compared.
3. We draw a histogram that sums the normalized probabilities of the various object/azimuth instances for the results obtained in step (2).
4. If the object associated with the highest frequency in the histogram is the same as the ground truth, we count it as one correct recognition.

5. Experiments

5.1. XPATCH SAR data

We use the XPATCH SAR simulator [21] to generate the data to perform controlled experiments. XPATCH is considered the best radar signature prediction code available; it has been extensively validated against primitive objects and complex full scale vehicles. In our research, we selected 6 in. resolution data to get a significant number of “pixels-on-target” to facilitate solving the occluded target recognition problem. It is possible to collect real data at this resolution and also super resolution techniques [26] exist to achieve such data. We generate SAR images of five objects (Fred tank, SCUD missile launcher, T72 tank, T80 tank and M1a1 tank, shown in Fig. 9) at 15° depression angle, and 90° squint angle (fixed), at each of the azimuth angles from 0° to 359°. We extract the 20 scattering centers (local maxima) with largest magnitudes. In the experiments, since we want to test the performance of our approach under partial occlusion and spurious data, we simulate realistic occlusion situations and generate images for training and testing.

Simulating occlusion: There are no accepted empirical models of object occlusion in SAR imagery. So, we consider the occlusion to occur possibly from nine different directions as shown in Fig. 10. Scattering centers being occluded are not available. Moreover, we add some spurious data into the image to simulate noise or clutter. For instance, 20 scattering centers are shown in each image of Fig. 10. They are obtained by removing four scattering centers (20% occlusion) from the center of one object or from one particular direction (simulated occlusion) and adding four spurious scattering centers into the image. The spurious scattering centers are added based on the following two rules: (1) The location of the scattering center is generated as a pair of random numbers. (2) The magnitude of the scattering center depends on a random number r between 1 and 50. We use the magnitude of the rth brightest scattering center as the magnitude of the spurious scattering center. Top 50 vs top 20 biases clutter to low values.

For the training data, based on the method of simulating occlusion described above, we generate 90 images from the original image (10 samples for each of nine directions) at 5% occlusion and another 90 images at 10% occlusion. Including the original image, we have 181 images per object per azimuth angle to train multiple HMM models. Thus, we have a total of 325,800 (five objects, 360 azimuths, 181 occluded images) samples for training. For the testing data we generate one image with o scattering centers occluded ($o = 2, 4, 6, 8$ or 10) from direction d ($d = 0, 1, \ldots, 8$) per azimuth angle per object. So there are 1800 images (five objects × 360°) generated for testing of occlusion with o scattering centers occluded from direction d. Thus, we have a total of 81,000 (five objects, 360 azimuths, five different occlusions 10%–50%, and nine directions) samples for testing. Fig. 11 shows the XPATCH SAR images (not to scale) of five objects taken under depression angle 15° and azimuth 225°. The scattering centers under various percentage of occlusion (shown on the top of the figure) are also shown.

5.1.1. Training—building bank of HMM models for recognition

The results of experiments to choose the optimum of number of states and number of symbols of the HMM are shown in Table 2. We use data from five azimuth angles of five objects (Fred tank, SCUD missile launcher, T80 tank, T72 tank, and M1a1 tank). Table 2 shows that with the increase in the number of states and symbols, recognition performance increases. Considering both the recognition performance and the computational cost, we choose eight states and 32 symbols as the optimal number of states and symbols for our HMM models. Fig. 12 illustrates example parameters of a five state, four symbol HMM. Using the algorithm in Section 4.4, we built recognition models. For a selected sequence type we have 1800 (=360 azimuths × five object classes) HMM models. Since we have defined five kinds of observation sequences for each image (O_1, O_2, O_3, O_4, O_5), we get models based on each kind of observation sequence.

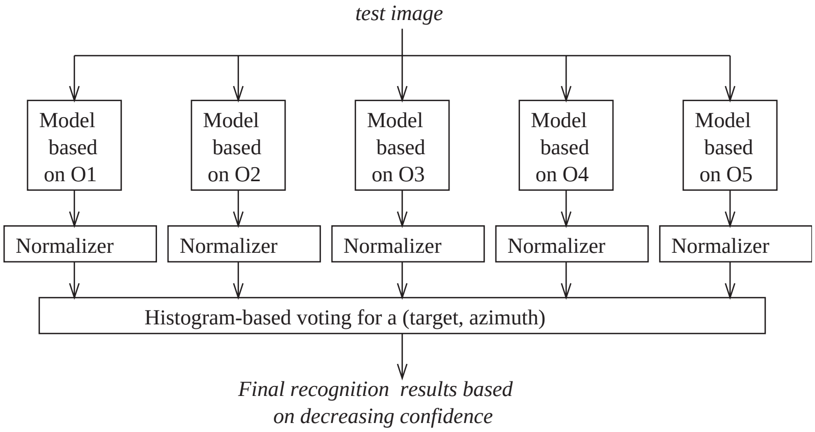


Fig. 8. Integration of results by histogram-based method.

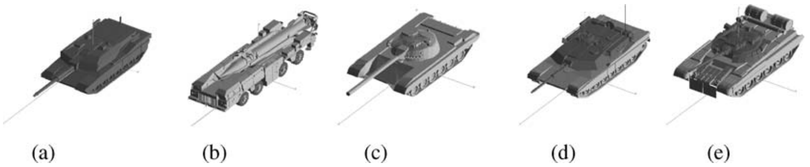


Fig. 9. Objects: (a) Fred tank; (b) SCUD missile launcher; (c) T72 tank; (d) T80 tank; and (e) M1a1 tank.

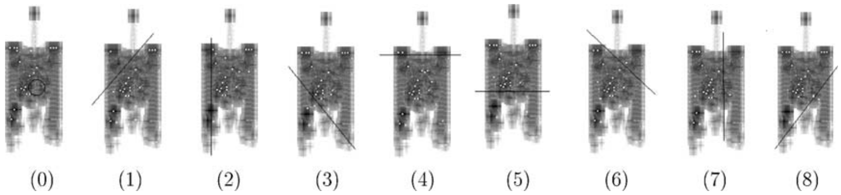


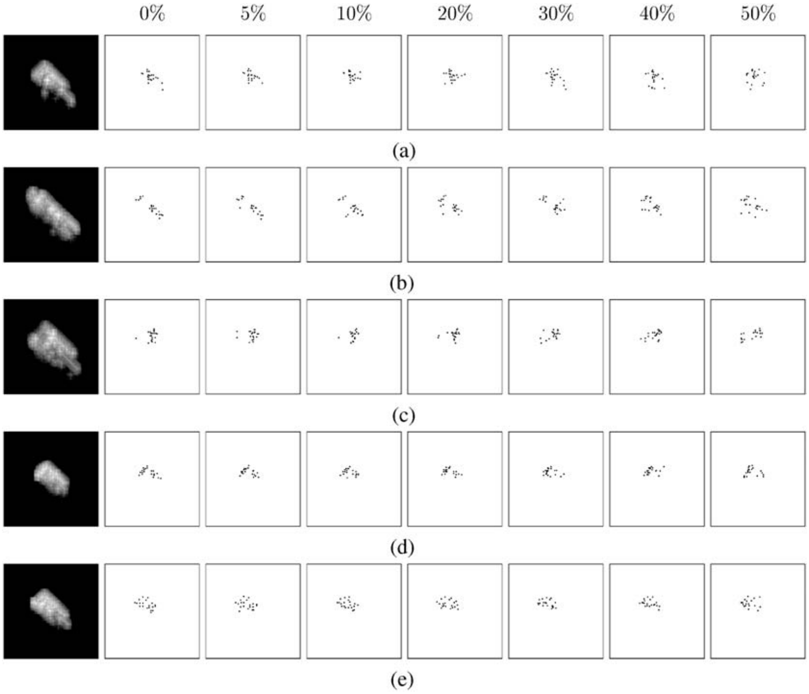
Fig. 10. Scattering centers of T72 tank at azimuth 0°, part of scattering centers are occluded from a particular direction (0–8, left to right).

5.1.2. Testing results

During testing phase, for a given observation sequence type each of the 81,000 testing images is tested against all models (1800 models: five objects, each has 360 models for each azimuth angle). If the model with the maximum probability is associated with the object and azimuth angle from which the observation sequence is extracted, we count it as one correct recognition (object type and its pose). Otherwise, we count it as one incorrect recognition. After we get the results on images with scattering centers occluded from all nine directions, we average these results and associate this recognition performance with the selected model for this particular percentage of occlusions. Fig. 13 shows the testing results for each of the five kinds of sequences: $O_1, O_2, \ldots, O_5$ (Section 4.3). The top curve, a dotted line, is the percentage that the test case object and pose is among the top 10 recognition results, and the lower curve, in solid line, indicates the percentage that the recognition result with the highest probability is the same as the test case object and pose.

5.1.3. Integration of results from multiple sequences

In order to understand the integration process better, we have done a set of testing experiments to show how integration improves the performance. Both geometric (O_2–O_5) and magnitude (O_1) observation sequences are used and the results are shown in Table 3. The table shows how many incorrect recognitions, made by using models based on sequence O_2 can be correctly recognized (“captured”) by models based on other sequences. We *define* the “upper bound” as the highest possible recognition performance that can be achieved using the five observation sequences, considering *only* the top candidate for recognition from each of the models. The percentage of correct recognition corresponding to “upper bound” is shown in the 8th column. The results shown in column 9 are obtained by integrating the



(a)

(b)

(c)

(d)

(e)

Fig. 11. Features of five targets under various occlusion from direction 8: (a) Fred tank; (b) Scud launcher with missile down; (c) T72 tank; (d) T80 tank; and (e) M1 tank.

Table 2
Recognition rate of HMM with different number of states and symbols

		id only		id with pose				id only		id with pose	
N	*M*	*R*	*I*	*R*	*I*	*N*	*M*	*R*	*I*	*R*	*I*
4	8	76.1	96.5	62.6	79.9	6	32	98.9	100.0	98.5	99.9
4	16	89.6	98.4	85.4	93.1	6	64	100.0	100.0	100.0	100.0
4	24	95.1	99.3	91.8	97.3	8	8	84.3	97.6	77.4	87.6
4	32	96.6	99.9	94.8	99.0	8	16	96.4	99.8	94.6	98.3
4	64	99.7	100.0	99.6	100.0	8	24	99.4	100.0	99.0	99.9
5	8	80.1	97.4	67.3	84.0	8	32	99.8	100.0	99.8	100.0
5	16	91.9	98.6	86.7	93.7	8	64	100.0	100.0	100.0	100.0
5	24	96.6	99.7	94.6	98.6	10	8	100.0	100.0	100.0	100.0
5	32	97.8	99.8	96.7	99.3	10	16	98.3	99.9	97.3	99.6
5	64	99.9	100.0	99.9	100.0	10	24	99.9	100.0	99.9	99.9
6	8	82.5	96.9	71.7	84.8	10	32	100.0	100.0	99.9	100.0
6	16	93.8	99.5	90.1	96.7	10	64	100.0	100.0	100.0	100.0
6	24	98.5	99.8	97	99.7						

N—No. of states. *M*—No. of symbols. R—Recognition rate % (top answer is correct). I—Indexing rate % (correct answer is in the top 5).

results from sequences O_1 to O_5. The integrated recognition performance is 86.8%. Note that the result are for both object and azimuth (object/azimuth), that is, both the object identity and pose should be correctly recognized.

In the rest of this section, we demonstrate the results using five kinds of observation sequences (O_1–O_5). We draw two curves (Fig. 14(a)) to show the “upper bound” and “lower bound” of recognition rate. The curve on the top is obtained by considering all five kinds of observation sequences, if a model based on one of them correctly recognizes the test data, we count it as a correct recognition. The “lower bound” or the bottom curve is the worst recognition result out of the five models. The second curve from the bottom in Fig. 14(b) is the result for integrated recognition. The

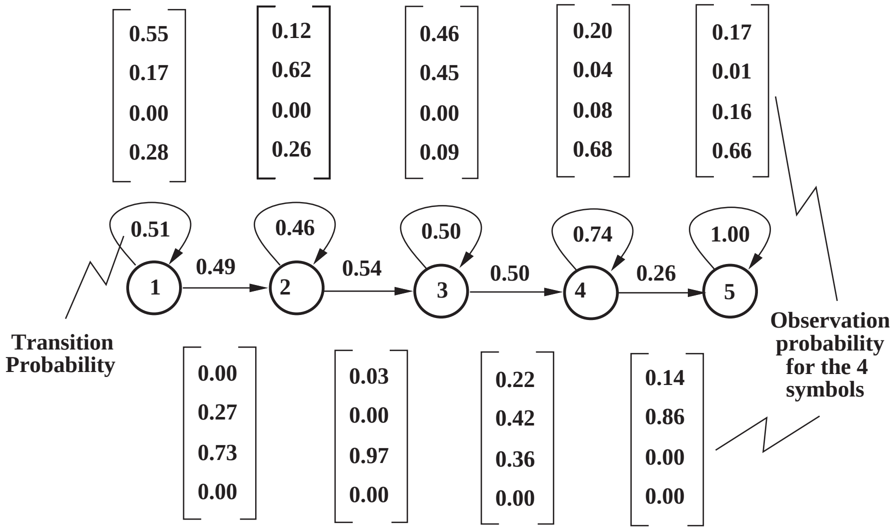


Fig. 12. An example: parameters of a five states, four symbols HMM. The number on edges represents the transition probability, and the vector associated with each transition represents $b_{ij}(k)$. In our case, we use HMM with eight states, 32 symbols.

corresponding confusion matrix for various amounts of occlusion is shown in Table 4. On the average, we find 86.8% correct recognition performance when the objects are occluded from 10% to 50%. The second curve from the top in Fig. 14(b) is obtained by counting a correct indexing result when the ground-truth object/azimuth is in the objects/azimuth associated with the highest five probabilities in the histogram. For the purpose of comparison, we have also superimposed the curves of Fig. 14(a) into Fig. 14(b) with “lower/upper” bounds. Considering the correct indexing answer in the top five responses, the average performance is 94.4% for five objects occluded from 10% to 50%. The difference between top 5 and top 1 recognition performance is 7.6%.

5.2. Real SAR data

The methods used here are the same as those used in the previous subsection. The only difference is that here we experiment on real data instead of XPATCH data. We use MSTAR public real SAR images (at one foot resolution and depression angle 30°) of two objects (T72 tank with serial number #a64 and ZSU anti-aircraft gun with serial number #d08). Ideally, we should have 360 object models for each azimuth for each object. However, we do not have 360 SAR images for each object in the MSTAR data set. For the T72 tank, there are 288 images available for different azimuths. Also for the ZSU gun, 288 images are available. Thus, each object consists of 288 azimuths (or aspects) which we call object models. Each object model consists of HMM models based on observation sequences (O_1–O_5). We extract 30 scattering centers with the largest magnitudes from each SAR image. Figs. 4 and 5 show some examples of SAR imagery and scattering centers extracted from these SAR images [22,24].

We consider the occlusion to occur possibly from nine different directions (center, four sides and four corners of the image). Scattering centers being occluded are not available. Moreover, we add back into the image at random locations a number of spurious scattering centers, equal to the number of occluded scatterers, of random magnitude. The random magnitude could be equal to the magnitude of any of the top 30 scatterers. This simulates noise or clutter from other objects. For example, for 30% occlusion, we remove nine scattering centers from the center of one object or from one particular direction and add randomly nine spurious scattering centers back into the image with magnitude randomly determined as stated above. Fig. 15 shows the SAR images of T72 tank and ZSU gun under depression angle 30° and azimuth 93°. The scattering centers under various occlusions are also shown. We compute the observation sequences based on the scattering centers available after the occlusion process has taken place. We assume there is only one object present in one image, but our technique to synthetically generate occluded data adds an equivalent number of clutter features which could be viewed as a surrogate for multiple objects. No real SAR data with multiple objects exist in the public domain.

5.2.1. Training and testing data

For the training data we generate 91 training images from each SAR image. The first one is the original SAR image without occlusion. Then we occlude the SAR image from nine directions. For each direction, the occlusion level is 5% and 10%. For each occlusion level, we generate five sample images. So 91 images are obtained from each original

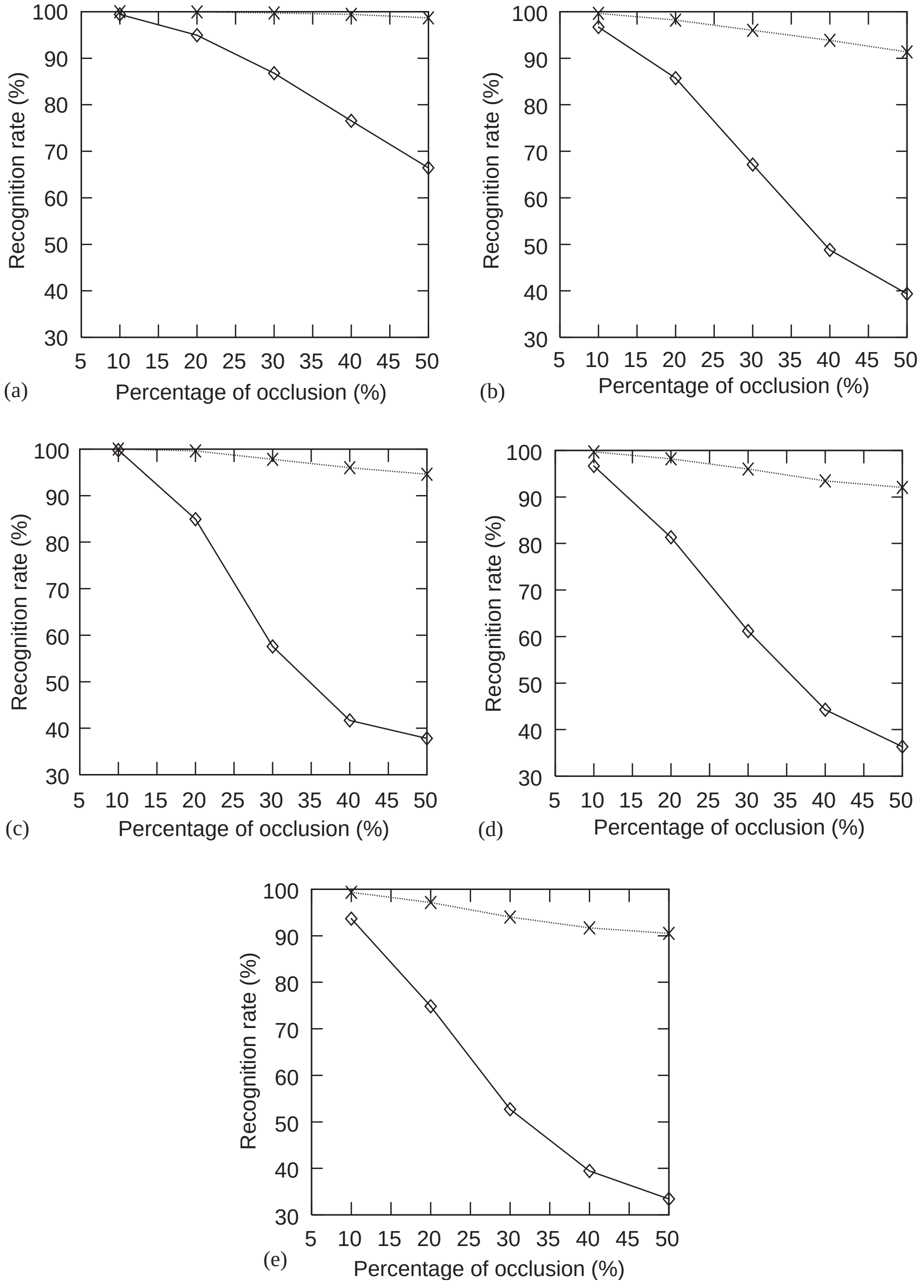


Fig. 13. Recognition rate vs. percentage of occlusion for HMM models based on (a) O_1, (b) O_2, (c) O_3, (d) O_4, and (e) O_5.

training SAR image. We have two objects and 288 SAR images of each object, thus, the number of training images is 52,416. Since there are five kinds of observation sequences that are extracted from each image, the total number of sequences is 262,080. For the testing data, from each SAR image, we generate 36 testing images. We occlude the SAR image from nine directions. For each direction, the occlusion level is from 20% to 50% with 10% increments. That is, the numbers of occluded scattering centers are 6, 9, 12, and 15, respectively. Thus, we have total 20,736 ($2 \times 288 \times 36$) testing images and corresponding to each image, there are five types of sequences.

5.2.2. Experimental results

During the training, for consistency, we use the same eight states and 32 symbols used for XPATCH data for our

Table 3
Testing results for occluded object recognition using 81,000 testing cases. Results based on integration of O_1–O_5

Percent. occlusion	Errors with model O_2	Errors captured by models				Errors using models O_1–O_5	% correct recognition ("upper bound")	% based on integrated recognition	% based on integrated indexing
		O_1	O_3	O_4	O_5				
10%	4	0	1	0	1	2	100.0	99.9	99.9
20%	271	19	53	74	101	7	99.6	98.9	99.6
30%	763	111	294	339	418	25	98.6	93.4	97.6
40%	1050	265	580	629	675	79	95.6	79.4	91.8
50%	1119	397	726	755	784	148	91.8	62.2	83.3
Average recognition rate							97.1	86.8	94.4

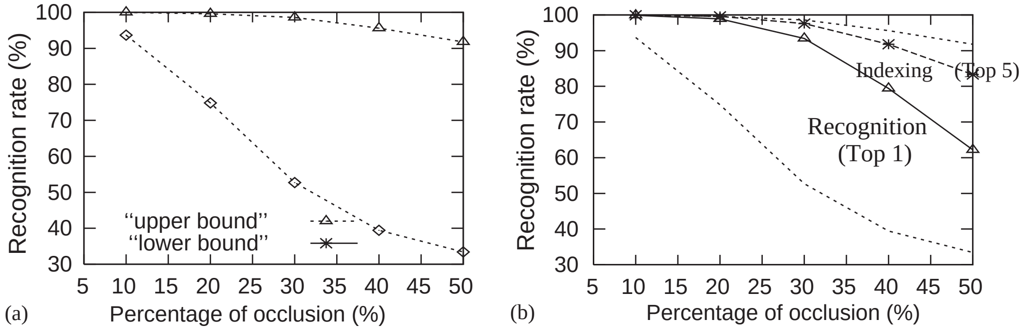


Fig. 14. (a) "Upper" and "lower" bound of recognition rate vs. percentage of occlusion. (b) Performance of integrated models: using integrated models O_1–O_5. The results for recognition (top 1) and indexing (top 5 candidates) are superimposed on the figure shown in (a).

HMM models for the real SAR data. Using the algorithm presented in Section 4.4, we build recognition models using the real SAR training data described in Section 5.2.1. For one sequence type we have 576 (288 azimuths × 2 object classes) HMM models. Since we have defined five kinds of observation sequences for each image (O_1, O_2, O_3, O_4, O_5), we get models based on each kind of observation sequence. During the testing phase, we apply the recognition algorithm (described in Section 4.5) to the real SAR test data (described in Section 5.2.1). Here, we consider only the kind of an object, which we call ID. We count a recognition result as a correct recognition if the HMM model with the maximum probability is associated with the object from which the testing sequence was extracted, we do not consider the corresponding azimuth angles of the HMM models.

Tables 5 and 6 show the experimental results on MSTAR real data. These results are obtained by integrating the results from five different type of sequences O_1, O_2, O_3, O_4, and O_5. Table 5 shows the results for recognizing 20–50% occluded T72 tank and ZSU gun. The confusion matrix shows how many of them are correctly/incorrectly identified. Table 6 shows the results similar to Table 5. The difference is that here we use "probability ratio threshold" instead of considering only the maximum probability. Only when the ratio between the maximum and second maximum (other object type) probabilities is greater than the probability ratio threshold (1.01 used here) we accept the recognition result. Otherwise, it is rejected and the test data is labeled as unknown object.

We also performed an additional experiment to determine the optimal number of observation sequences. Table 7 shows the results when only sequences O_1 and O_2 are used in integration. The results of Table 7 based on O_1 and O_2 are slightly better than the results of Table 5 based on all the five sequences (O_1 to O_5). The reason is that the results from sequences O_3, O_4, and O_5 are less reliable than others, since each of these sequences measures distances from a specific scattering center, for example, O_3 measures distances of other scattering centers from (reference) scattering center 1, O_4 measures distances of other scattering centers from scattering center 2, and so on. There is a significant probability that these prominent scattering centers (used as a reference) get occluded at higher levels of occlusion. If the first scattering center is occluded, then the entire sequence (O_3) is subject to error, which is reflected in the integration results. An alternative is to use all the scattering centers as reference points to build models to achieve increased performance at the expense of increased

Table 4
Confusion Matrix for five objects classes at varying amounts of occlusion (10–50%)

	% Occlusion	Fred	SCUD	T72	T80	M1a1
Fred	10	100.0	0.0	0.0	0.0	0.0
	20	99.2	0.0	0.1	0.4	0.3
	30	95.9	0.2	0.6	1.9	1.4
	40	87.1	0.7	2.8	5.5	3.9
	50	73.2	1.6	7.1	12.1	6.0
SCUD	10	0.0	100.0	0.0	0.0	0.0
	20	0.0	99.7	0.2	0.1	0.0
	30	0.9	97.3	1.2	0.4	0.3
	40	3.1	88.8	4.9	1.9	1.3
	50	5.6	77.9	11.9	2.7	1.9
T72	10	0.0	0.0	100.0	0.0	0.0
	20	0.4	0.2	99.2	0.1	0.2
	30	2.4	0.5	95.3	1.1	0.6
	40	9.1	2.1	82.5	3.8	2.4
	50	16.8	5.2	65.9	6.8	5.4
T80	10	0.0	0.0	0.0	100.0	0.0
	20	1.2	0.0	0.1	98.6	0.1
	30	6.9	0.0	0.6	91.1	1.4
	40	21.5	0.1	1.6	72.6	4.2
	50	37.4	0.8	3.1	50.9	7.8
M1a1	10	0.0	0.0	0.0	0.0	100.0
	20	1.6	0.0	0.1	0.3	98.0
	30	8.5	0.2	0.7	2.9	87.8
	40	22.5	0.8	2.0	8.5	66.1
	50	36.9	1.1	5.2	13.8	42.9

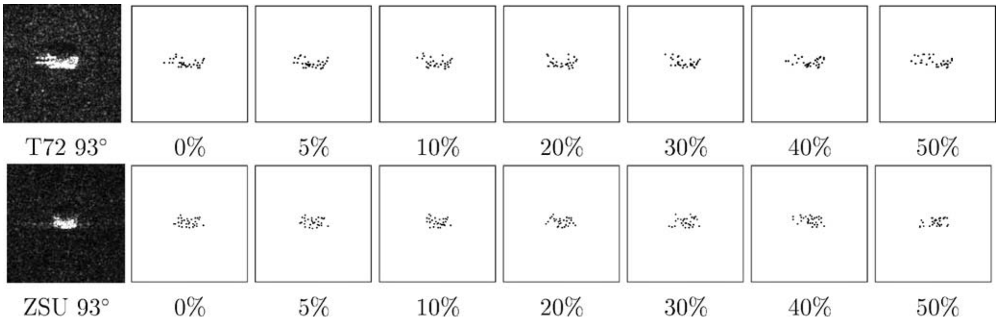


Fig. 15. Features of T72 tank #a64 and ZSU anti-aircraft gun #d08 under various occlusions from direction 2.

Table 5
Confusion matrix for various occlusions using MSTAR data

Target type	Occlusion level							
	20%		30%		40%		50%	
	T72	ZSU	T72	ZSU	T72	ZSU	T72	ZSU
T72	280(97.2%)	8(2.8%)	233(80.9%)	55(19.1%)	200(69.4%)	88(30.6%)	164(56.9%)	124(43.1%)
ZSU	5(1.7%)	283(98.3%)	64(22.2%)	224(77.8%)	98(34.0%)	190(66.0%)	117(40.6%)	171(59.4%)

Table 6
Recognition results for various occlusions using MSTAR data

Target type	Occlusion level											
	20%			30%			40%			50%		
	Correct	Error	Reject	Correct	Error	Reject	Correct	Error	Reject	Correct	Error	Reject
T72	279(96.9%)	7(2.4%)	2(0.7%)	230(79.9%)	47(16.3%)	11(3.8%)	188(65.3%)	75(26.0%)	25(8.7%)	156(54.2%)	110(38.2%)	22(7.6%)
ZSU	281(97.6%)	4(1.4%)	3(1.0%)	216(75.0%)	59(20.5%)	13(4.5%)	168(58.3%)	91(31.6%)	29(10.1%)	164(56.9%)	102(35.5%)	22(7.6%)

computation, not just the top three scattering centers that we used in sequences O_3, O_4, and O_5.

From Tables 5–7, we can see that the performance degrades as the occlusion level increases and reasonable recognition performance can be achieved when the occlusion is as much as 30–40%. The time needed to recognize a test case is 6ms on a SUN Ultra 2.

6. Conclusions

Recognition of occluded objects has been a significant problem for automatic target recognition. In this paper, we have presented a novel conceptual approach based on hidden Markov modeling for the recognition of occluded objects in SAR images. The approach uses multiple models for various observation sequences that are chosen based on the SAR image formation and account for both the geometry and magnitude of SAR image features. We have shown the results on both XPATCH and real SAR data. The number of observation sequences and the number of features are design parameters which can be optimized by following the approach presented in the paper. We have demonstrated that the HMM approach makes use of available structural information to solve the problem caused by occlusion and noise. It takes the spatial arrangement of structural information as a whole and is able to collect useful information from distorted or partially unreliable patterns. Reasonable recognition can be achieved for up to 30–40% occlusion with both the simulated and real SAR data.

7. Summary

Recognition of occluded objects has been a significant problem for automatic target recognition. Stochastic models provide some attractive features for pattern matching and recognition under partial occlusion and noise. In this paper, we present a novel conceptual approach based on hidden Markov modeling for the recognition of occluded objects in SAR images. The approach uses multiple models for various observation sequences that are chosen based on the SAR image formation and account for both the geometry and magnitude of SAR image features. The number of observation sequences and the number of features are design parameters which can be optimized by following the approach presented in the paper. To improve performance, we integrate these models synergistically using their probabilistic estimates for recognition of a particular object at a specific azimuth. We show the results on both XPATCH and real SAR data and demonstrate that the HMM approach makes use of available structural information to solve the problem caused by occlusion and noise. It takes the spatial arrangement of structural information as a whole and is able to collect useful information from distorted or partially unreliable patterns. Reasonable recognition can be achieved

Table 7
Results of integration (sequences O_1 and O_2)

Occlusion level	No. of correct recognition				No. of test
	20%	30%	40%	50%	sequences
T72 tank	284(98.6%)	249(86.5%)	208(72.2%)	186(64.6%)	288
ZSU gun	285(99.0%)	238(82.6%)	194(67.4%)	175(60.8%)	288

for up to 30–40% occlusion with both the simulated and real SAR data.

Acknowledgements

This research is supported in part by grant F49620-97-1-0184 and F49620-02-1-0315. The contents and information do not necessarily reflect the position and policy of the US Government.

References

[1] B. Bhanu, Automatic target recognition: state of the art survey, IEEE Trans. Aerospace Electron. Syst. 22 (1986) 364–379.

[2] B. Bhanu, D.E. Dndgeon, E.G. Zelnio, A. Rosenfeld, D. Casasent, I.S. Reed, Introduction to the special issue on automatic target detection and recognition, IEEE Trans. Image Process. 6 (1) (1997) 1–6.

[3] B. Bhanu, T.L. Jones, Image understanding research for automatic target recognition, Proceedings of the ARPA Image Understanding Workshop, San Diego, CA, USA, January 26–29, 1992, pp. 249–254.

[4] D.E. Dndgeon, R.T. Lacoss, An overview of automatic target recognition, Lincoln Laboratory J. 6 (1) (1993) 3–9.

[5] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–285.

[6] L.R. Rabiner, B.H. Juang, An introduction to hidden Markov models, IEEE Acoust. Speech Signal Process. Mag. 3 (1) (1986) 4–16.

[7] L.R. Rabiner, B.H. Juang, S.E. Levinson, M.M. Sondhi, Recognition of isolated digits using hidden Markov models with continuous mixture densities, AT& T Tech. J. 64 (6) (1985) 1211–1233.

[8] O.E. Agazzi, S.S. Kuo, Hidden Markov model based optical character recognition in the presence of deterministic transformations, Pattern Recognition 26 (12) (1993) 1813–1826.

[9] O.E. Agazzi, S.S. Kuo, Pseudo two-dimensional hidden Markov models for document recognition, AT& T Tech. J. 72 (5) (1993) 60–72.

[10] W.E.L. Grimmson, Object Recognition by Computer, MIT Press, Cambridge, MA, 1990.

[11] L.G. Shapiro, G.C. Stockman, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 2001.

[12] R.M. Haralick, L.G. Shapiro, Computer and robot vision, Vols. 1 and 2, Addison-Wesley Publishing Company, Reading, MA, 1992.

[13] D. Chen, X. Cheng (Eds.). Pattern Recognition and String Matching. Kluwer Academic Publishers, Dordrecht, 2002.

[14] K.H. Fielding, D.W. Ruck, Spatio-temporal pattern recognition using hidden Markov models, IEEE Trans. Aerospace Electron. Syst. 31 (4) (1995) 1292–1300.

[15] R.R. Rao, R.M. Mersereau, On merging hidden Markov models with deformable templates, Proceedings of the International Conference on Image Processing, Los Alamitos, CA, USA, 1995, pp. 556–559.

[16] D.P. Kottle, P.D. Fiore, K.L. Brown, J.K. Fwu, A design for HMM-based SAR ATR, SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery V, Orlando, FL, USA, Vol. 3370, April 1998, pp. 541–551.

[17] L.M. Novak, G.J. Owirka, C.M. Netishen, Performance of a high-resolution polarimetric SAR automatic target recognition system, Lincoln Laboratory J. 6 (1) (1993) 11–24.

[18] J.H. Yi, B. Bhanu, M. Li, Target indexing in SAR images using scattering centers and Hausdorff distance, Pattern Recognition Lett. 17 (1996) 1191–1198.

[19] G. Jones III, B. Bhanu, Recognition of articulated and occluded objects, IEEE Trans. Pattern Anal. Mach. Intell. 21 (7) (1999) 603–613.

[20] K.F. Lee, Automatic Speech Recognition—The Development of the SPHINX System, Kluwer Academoc Publishers, Boston, 1989.

[21] D.J. Andersh, S.W. Lee, H. Ling, C.L. Yu, XPATCH: a high frequency electromagnetic scattering prediction code using shooting and bouncing ray, in: Proceedings of the Ground Target Modeling and Validation Conference, Houghton, MI, USA, August 1994, pp. 498–507.

[22] T. Ross, S. Worrell, V. Velten, J. Mossing, M. Bryant, Standard SAR ATR evaluation experiments using the MSTAR public release data set, SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery V, Orlando, FL, USA, Vol. 3370, April 1998, pp. 566–573.

[23] J.P. Fitch, Synthetic Aperture Radar, Springer, New York, 1988.

[24] B. Bhanu, G. Jones III, Recognizing target variants and articulations in synthetic aperture radar images, Opt. Eng. 39 (3) (2000) 712–723.

[25] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, Reading, MA, 1974.

[26] L. Novak, G. Benitz, G. Owirka, L. Bessette, ATR performance using enhanced resolution SAR, SPIE Conference on Algorithms for Synthetic Aperture Radar Imagery, Orlando, FL, USA, Vol. 2757, April 10–12, 1996, pp. 332–337.

[27] X. Shen, P. Palmer, Uncertanty propagation and the matching of junctions as feature groupings, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1381–1395.

[28] G. Sockman, Object recognition and localization via pose clustering, Computer Vision Graphics Image Process. 40 (3) (1987) 361–387.

[29] B. Bhanu, J. Ming, Recognition of occluded objects: a cluster-structure algorithm, Pattern Recognition 20 (2) (1987) 199–211.

[30] S. Moss, E.R. Hancock, Structural constraints for pose clustering, Proceedings of the Eighth International Conference on Computer Analysis of Images and Patterns, Berlin, Germany, September 1999, pp. 632–640.

[31] C.F. Olson, Efficient pose clustering using a randomized algorithm, Int. J. Comput. Vision 23 (2) (1997) 131–147.

[32] Y. Lamden, H. Wolfson, Geometric hashing: a general and efficient model-based recognition scheme, Proceedings of the International Conference on Computer Vision, Washington DC, USA, December 1998, pp. 238–249.

[33] K. Surendro, Y. Anzai, Non-rigid object recognition using principal component analysis and geometric hashing, Proceedings of the Seventh International Conference, Computer Analysis of Images and Patterns, Berlin, Germany, September 1997, pp. 50–57.

[34] A. Au, P. Tsang, Affine invariant recognition of 2D occluded objects using geometric hashing and distance transformation, Proceedings of the IEEE TENCON Digital Signal Processing Applications, New York, NY, USA, Vol. 1, November 1996, pp. 63–67.

[35] C. Olson, D. Huttenlocher, Automatic target recognition by matching oriented edge pixels, IEEE Trans. Image Process. 6 (1) (1997) 103–113.

[36] J. Albus, Applications of an efficient algorithm for locating 3D models in 2D images, Proceedings of the SPIE Conference on Automatic Target Recognition VIII, Orlando, FL, USA, Vol. 3371, April 1998, pp. 416–427.

[37] X. Tu, B. Dubuisson, Contrast pixel enhancement and matching in an image sequence by relaxation, Proc. SPIE 1613 (1991) 296–301.

[38] A. Rangarajan, H. Chui, A mixed variable optimization approach to non-rigid image registration, DIMACS Workshop on Discrete Mathematical Problems with Medical Applications, 2000, pp. 105–123.

[39] B. Bhanu, O. Faugeras, Shape matching of two-dimensional objects, IEEE Trans. Pattern Anal. Mach. Intell. 6 (2) (1984) 137–156.

[40] Y. Chen, Y. Lin, S. Kung, A feature tracking algorithm using neighborhood relaxation with multi-candidate pre-screening, Proceedings of the International Conference on Image Processing, New York, NY, USA, Vol. 2, 1996, pp. 513–516.

[41] S. Sharghi, F. Kamangar, Stereo correspondence using geometric relational matching. Proceedings of the SPIE Conference on Visual Communication and Image Processing, San Jose, CA, USA, Vol. 3653, January 1999, pp. 582–592.

[42] R. Myers, R. Wilson, E. Hancock, Efficient relational matching with local edit distance, Proceedings of the Fourth International Conference on Pattern Recognition, Los Alamitos, CA, USA, Vol. 2, August 1998, pp. 1711–1714.

About the Author—BIR BHANU received the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, the Ph.D. degree in electrical engineering from the Image Processing Institute, University of Southern California, Los Angeles, and the M.B.A. degree from the University of California, Irvine.

Currently Prof. Bhanu is the Director of the Center for Research in Intelligent Systems (CRIS) at the University of California, Riverside where he has been a Professor and Director of Visualization and Intelligent Systems Laboratory (VISLab) since 1991. Previously, he was a Senior Honeywell Fellow at Honeywell Inc. in Minneapolis, MN. He has been on the faculty of the Department of Computer Science at the University of Utah, Salt Lake City, UT, and has worked at Ford Aerospace and Communications Corporation, CA, INRIA-France and IBM San Jose Research Laboratory, CA. He has been the principal investigator of various programs for DARPA, NASA, NSF, AFOSR, ARO and other agencies and industries in the areas of learning and vision, image understanding, pattern recognition, target recognition, navigation, image databases, and machine vision applications. He is the co-author of books on *Computational Learning for Adaptive Computer Vision* (Forthcoming), *Genetic Learning for Adaptive Image Segmentation* (Kluwer 1994), and *Qualitative Motion Understanding* (Kluwer 1992). He has received two outstanding paper awards from the Pattern Recognition Society and has received industrial awards for technical excellence, outstanding contributions and team efforts. He has been the associate editor/guest editor of various journals and transactions. He holds 11 US and international patents and over 230 reviewed technical publications in the areas of his interest. He has been General Chair for IEEE Workshops on Applications of Computer Vision. Chair for the DARPA Image Understanding Workshop, General Chair for the IEEE Conference on Computer Vision and Pattern Recognition, Program Chair for the IEEE Workshops on Computer Vision Beyond the Visible Spectrum and Chair for IEEE Workshop on Learning in Computer Vision and Pattern Recognition.

Prof. Bhanu is a Fellow of IAPR, IEEE, AAAS and SPIE, and a member of ACM and AAAI.

About the Author—YINGQIANG LIN received his B.S. and M.S. degrees in computer science from Fudan University, Shanghai, China in 1991 and 1994, respectively. He received his Ph.D. degree in Computer Science from the University of California at Riverside in June 2003. His research interests include image processing, pattern recognition and machine learning.