Tan, K.H. and Ghanbari, M. (1995). Layered image coding using the dct pyramid. *IEEE Transactions on Image Processing*, **4**, 512–516.

Trussell, H.J. and Fogel, S. (1992). Identification and restoration of spatially variant motion blurs in sequential images. *IEEE Transactions on Image Processing*, **1**, 123–126.

Tuttle, G.T., Fallahi, S., and Abidi, A.A. (1993). An 8b CMOS vector A/D converter. *ISSCC Technical Digest*, **36**, 38–39.

Unser, M. (1993). On the optimality of ideal filters for pyramid and wavelet signal approximation. *IEEE Transactions on Signal Processing*, **41**, 3591–3596.

Waxman, A.M., Gove, A.N., Fay, D.A., Racamato, J.P., Carrick, J.E., Seibert, M.C., and Savoye, E.D. (1997). Color night vision: Opponent processing in the fusion of visible and IR imagery. *Neural Networks*, **10**(1), 1–6.

Waxman, A.M., Seibert, M., Bernardon, A.M., and Fay, D.A. (1993). Neural systems for automatic target learning and recognition. *The Lincoln Laboratory Journal*, **6**(1), 77-116.

Williamson, J.R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, **9**, 881–897.

Williamson, J.R. (1997). A constructive, incremental-learning network for mixture modeling and classification. *Neural Computation*, in press.

Interactive Target Recognition Using a Database-Retrieval Oriented Approach

G. Sudhir and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, California 92521
Email: bhanu@cris.ucr.edu
URL: http://www.cris.ucr.edu

Abstract

Recognition of objects when the number of model objects becomes large is a challenging problem which makes it increasingly difficult to view the object recognition problem as – "find *the* best match" problem. We present a database-retrieval oriented approach where the goal is to index, retrieve, rank and output a few top-ranked models, according to their similarity with an input query object. The approach consists of three stages: (1) feature-based representation of model objects and object-feature correspondence analysis; (2) clustering and indexing of the model objects in the factor space; and (3) ranking indexed models based on mutual information with query object. The approach is suitable for semi-automatic object recognition tasks which involve human interaction. Experimental results are presented using MSTAR data to demonstrate the merits of the approach.

1 Introduction

Model-based object recognition is a powerful approach which involves invariant feature domain representations of models of different objects in a model database, and matching these to the features of a new observation of an object, to select the best match. However, as the number of models increases, it becomes increasingly difficult to view the object recognition problem as – "find *the* best match" problem. There are two main reasons for this: (1) The discriminating power of a set of known features becomes increasingly insufficient for finding *the* correct matching model; and (2) The sensitivity of features to changes in object pose, image formation geometry, sensor parameters, etc. adds to the problem of correct recognition.

Supported in part by grant F49620-97-1-0184. The contents and information do not necessarily reflect the position or the policy of the U.S. Government.

In this context, the model-based object recognition problem shares similarity with the *content-based retrieval* problems where the insufficiency of features arises in capturing the notion of *content*. However, the database retrieval approach – wherein, the goal is index, retrieve, rank and output a few top-ranked models as the most probable matches – has proved to be useful for various practical tasks. Hence, for the case of large model databases, taking a database-retrieval stance to model-based recognition is a practically useful one for various interactive applications. One such case is that of model-based recognition of targets in Synthetic Aperture Radar (SAR) images. As has been observed and quantified in our earlier research efforts [1], one of the main characteristics of SAR images of objects is the sensitivity of features (scattering centers) to *azimuthal variation* (pose variation). That is, for a given depression angle, there is very low invariance for the scattering centers (features which are local maxima) when the azimuth of an object is changed even by a couple of degrees. This requires a set of 360 models for a single object, hence many thousands of models in the model-database. This paper presents a database-retrieval oriented approach for the problem using a feature-based representation, classification and ranking of the models.

2 Related Research and Contributions

Recently, Pun *et al.* [2] have used correspondence analysis (CA) for statistical structuring of pictorial databases for content based retrieval. They represent images using aggregate features based on multiple cues, and do CA on images and attributes. However, in the factor space generated by CA, they use the feature projections only to explain the dominant factors and not as seeds for clustering objects. They propose an Ascendant Hierarchical Classification method [3, 4] for structuring the projections of the images in the factor space. For large model databases of objects with sensitive and noisy features (like the SAR target database we have used), the projections of objects are spread out almost randomly in the factor space, and as a result, a tree based classification may not be appropriate. Triangle-inequality (TI) based indexing schemes take advantage of the triangle inequality of distance measures to reduce the number of direct comparisons in a threshold search[1]. In the context of content-based retrieval from image databases Barros *et al.* [5] have applied the idea to a real image database and Berman *et al.* [6] have reported the performance of different algorithms for the selection of *key* objects as well as handling multiple distance measures on a set of image features. However, application of the TI using the distance between objects in the feature space has the limitation of not dealing with any redundancies of feature set in capturing a description of the database objects. Berman *et al.* [6] have reported on ranking the retrieved model objects according to the computed lower bounds on distances, in the context of content-based retrieval of images. However, in the context of object recognition, since the feature based representation of model objects is usually insufficient to capture *all* the properties of objects, the ordering based on lower bounds may not be accurate. Viola *et al.* [7] use MI measure in their algorithm for alignment of a model object to a new observation, assuming that the new observation belongs to the same model object. In our context, we use the MI measure for comparing a *candidate* model object to a query object. For the empirical estimation of MI, we have used the technique reported in [8].

The main contribution of our paper is that it presents a new, systematic approach to model-based object recognition from the database-retrieval point of view. The advantages of our approach are, (*i*) it deals with feature redundancies using CA, (*ii*) it proposes feature-based clustering of model objects in factor space, followed by TI based indexing of objects for each cluster, as an useful alternative to standard hierarchical clustering and tree-based indexing methods, and (*iii*) advocates a robust approach for ranking the initial retrievals using mutual information between a query object and a candidate model object. Our approach compares the query and model objects by considering their information content first in the feature domain (in TI based indexing in factor space generated by CA), and then in the original data domain (in mutual information based ranking of initial retrievals). The former step helps in quickly pruning a large number of models from consideration for a query object and the latter step ranks the candidates in a more robust manner.

[1]Given an object database O, a query object Q, and a distance measure $d(\cdot)$, the threshold search for a given threshold T is to search for all objects $o \in O$ such that the distance $D(Q, o) \leq T$.

3 Technical Approach

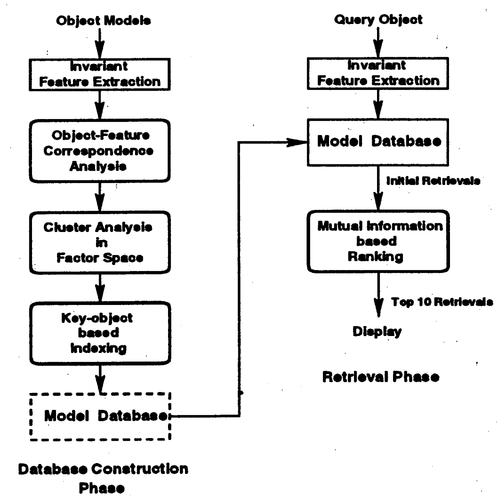


Figure 1: Overview of our database-retrieval oriented approach for object recognition.

Figure 1 illustrates our conceptual approach. There are two phases: (1) Database construction phase, and (2) Retrieval phase. In the database construction phase, the model objects are processed by an invariant feature extraction module. This represents the model objects in a feature space. It is processed by an object-feature CA module which projects both objects and features in a common reduced-dimension factor space. These projections are further analyzed and clusters of objects are formed in the factor space. For each cluster, a set of *key* objects are computed to enable TI based indexing of the objects in the factor space. Finally, the model information – the model objects, factors, clusters, key objects for each cluster, distances of model objects from key objects in each cluster – is assembled to generate the model database. In the retrieval phase, an unknown (query) object is processed by the invariant feature extraction module to represent it in the feature space. Then its features are input to the model database to index and retrieve a set of candidate models which are further ranked by the mutual information module to output only the top 10 ranked retrievals as the most probable matches to the input query object.

3.1 Database Construction Phase

The first step in this phase is the extraction of a set of invariant features for the model objects to represent them in the feature domain. In our case of SAR target databases, we have used translation invariant features [1]. We present details about these features in Sec. 4.

3.1.1 Object-feature Correspondence Analysis

Like any factor analysis method, CA provides a compact representation in a low dimension factor space of large sets of numerical data It shares the linear algebra of finding the *factor axes* with other factor analysis methods. However, in CA, the coordinates of the data points are defined so that the usual Euclidean metric in the factor space corresponds to the χ^2 distance between the points. Thus, the analysis is in terms of the *independence* of the data. Furthermore, unlike other factor analysis methods, CA assigns *symmetric* roles to rows (objects) and columns (features). This permits simultaneous representation of both objects and features in a common factor space which not only helps interpretation of the factor space but also makes clustering of objects easier (see Sec. 3.1.2). In the following, we describe CA as relevant to the analysis of objects and features.

Let there be M model objects and N features for each object. For large number of model objects, $N \ll M$. The model objects and their features are represented as the matrix $D = [d_{i,j}]$, $1 \leq i \leq M, 1 \leq N$, where, the rows identify the model objects and the columns identify the features. For CA, D is processed using the following sequence of steps: 1: Compute the normalized data matrix $K = [k_{i,j}]$, $1 \leq i \leq M$, $1 \leq j \leq N$ using $k_{i,j} = \frac{d_{i,j}}{\sqrt{d_I\, d_J}}$ where $d_I = \sum_{j=1}^{N} d_{i,j}$ and $d_J = \sum_{i=1}^{M} d_{i,j}$; 2: Compute[2] the χ^2 matrix $H = K^T K$ where K^T stands for *transpose* of a matrix K. Note that H is a $N \times N$ matrix and its elements in terms of original data matrix D are given by $h_{k,l} = \frac{1}{\sqrt{d_K\, d_L}} \sum_{i=1}^{M} \frac{d_{i,k}\, d_{i,l}}{d_I}$; where $d_I = \sum_{j=1}^{N} d_{i,j}$, $d_K = \sum_{i=1}^{M} d_{i,k}$, and $d_L = \sum_{i=1}^{M} d_{i,l}$; 3: Compute Singular Value Decomposition of H to find the eigenvalues $\lambda_i, 1 \leq i \leq N$ and eigenvectors $V_i, 1 \leq i \leq N$ of H. Note that λ_1 is always 1 because of normalization of the data and $\lambda_i \geq \lambda_j$ if $i \leq j$, $2 \leq i, j \leq N$. The eigenvalues λ_2 to λ_N determine the variance of the system and their corresponding eigenvectors (V_2 to V_N) determine the factor space and hence are called *factors*. Let $F_i = V_{i+1}, 1 \leq i \leq N-1$ be the factors; 4: Compute the ratio $r_i, 2 \leq i \leq N$ by using $r_i = \frac{\lambda i}{\sum_{j=2}^{N} \lambda j}$. These ratios indicate the percentage of the total variance of the system that each factor F_i explains. Note that only a few factors can explain up to 90% of the total variance of the system. Let P denote that number, where $P \leq N-1$; 5: Project the model objects $O_i, 1 \leq i \leq M$ along each factor axis $F_k, 1 \leq k \leq P$ using $O_{i,k} = \frac{\sqrt{Z}}{d_I} \sum_{j=1}^{N} \frac{d_{i,j}}{\sqrt{d_J}} f_{k,j}$ where $Z = \sum_{i=1}^{M} \sum_{j=1}^{N} d_{i,j}$, $d_I = \sum_{j=1}^{N} d_{ij}$, $d_J = \sum_{i=1}^{M} d_{ij}$ and $f_{k,j}$ is the j^{th} element of factor F_k. The quantity $O_{i,k}$ represents the scalar coordinate of the object O_i along factor F_k; 6: Project the features $A_j, 1 \leq j \leq N$ along each factor axis $F_k, 1 \leq k \leq P$ using $A_{j,k} = \frac{1}{d_J \sqrt{\lambda_k}} \sum_{i=1}^{M} d_{i,j}\, O_{i,k}$ where $d_J = \sum_{i=1}^{M} d_{ij}$. The quantity $A_{j,k}$ represents the scalar coordinate of the feature A_j along factor F_k. After these steps, the objects and features are both represented in a common, reduced dimension factor space. In all the experiments examples reported in this paper the first two factors determine more than 70% of the total variance Thus, only the first two *dominant* factors are considered in the clustering and indexing phase.

[2]This is actually a simplified, computationally less expensive version of the full χ^2 matrix; however, there is no difference as far the analysis is concerned since this matrix has the same eigen-structure as the full χ^2 matrix, except that the largest eigenvalue becomes trivial.

3.1.2 Cluster Analysis in Factor Space

We first identify different "feature-groups" in the factor space and then use their centroids as the *seeds* for generating clusters of model objects. There are two advantages of this: (1) the number of features are far less the number of model objects; thus, determining different groups of features in the factors space is not computationally intensive; (2) feature-groups, which act as the seeds around which objects get clustered, also explain automatically why the object cluster was formed.

3.1.3 Key-object based Indexing

Since the number of feature clusters is usually small (there are just 4 to 6 clusters of features in our experiments), there can be a large number of model objects inside each cluster. In order to efficiently index objects inside the clusters, we employ the TI based indexing scheme. The TI based indexing schemes rely on comparing a set of key objects to the database objects according to a distance metric, and storing the computed distances. The basic idea is to exploit the TI at the retrieval time to quickly compute the lower bounds on the distance of each database object from the query. The reason behind all these schemes is the fact that the distance between two objects cannot be less than the difference of their distances to any other object. Mathematically, if O is a database object, Q is a query object and K is some *key* object, the inequality $d(O,Q) \geq |d(O,K) - d(Q,K)|$ always holds. Thus, by comparing the database and query objects to a third *key* object, a lower bound on the distance between the model and query objects can be obtained. If a threshold T on the distance between a model and query object is known (or given), this lower bound can be compared to T to eliminate from any further consideration all those models whose lower bound is more than T. Note that, if the distances of all the model objects from the key object are stored, the only distance computation that needs to be done in order to know all the lower bounds is that between the query and key objects.

In our experiments, for simplicity, we have selected the two furthest apart objects in a cluster as the key objects. Thus the total number of keys used to index the model database in our approach is twice the number clusters.

3.1.4 Algorithm for Clustering and Indexing

In the following, we describe the sequential steps for clustering and indexing: 1: Consider the projections of M features along the first $\mathcal{F}$ *dominant* factors. Closely projected features explain the system similarly and are redundant. Group such closely projected features into a "feature-group" representing a single feature class. Let there be C different feature classes. Define the center of feature-group as the centroid of the feature projections of that group; 2: Generate C clusters of model objects using centers of each feature-group as the *seed* and using nearest-neighbor (NN) rule in the factor space; 3: For each cluster, select two model objects k_1 and k_2 which are furthest apart in the cluster. They form two key objects for the cluster; 4: For each cluster, compute the distances of model objects in the factor space with each of the key object; 5: For each cluster, store (*i*) indices of the model objects, (*ii*) indices of the key objects and (*iii*) distances of the model objects to key objects, in the model database.

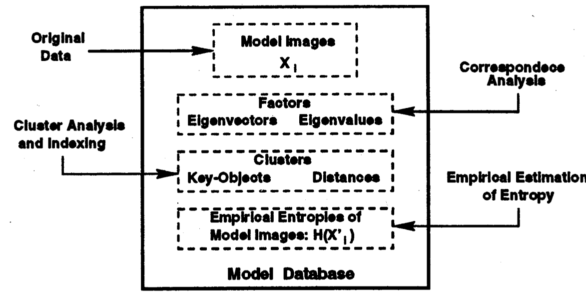


Figure 2: Contents of the Model Database.

3.1.5 Building model database

Figure 2 shows in detail the contents of the model database in our approach. The original model data is stored in the model database, for when the mutual information of the query object with each of the candidate model objects needs to be computed. Also, the *factors* computed by the object-feature analysis are stored to be used in the retrieval phase. The stored clusters contain two main items: key-objects and distances of all the model objects to all the key-objects along with indices to the model objects. Note that these distances are in the factor space. Finally, the empirical entropies of all the model objects are computed *a priori* and stored as part of the model database to speed-up the process of empirical estimation of mutual information.

3.2 Retrieval Phase

In this phase, a new observation is given in the form of a query object which is processed as follows: 1: Compute the same invariant features as done for each model object. Let $Q = [q_1, q_2, \cdots, q_N]$ be the query features; 2: Project the Q along each factor axis $F_k, 1 \leq k \leq P$ using $Q_k = \frac{\sqrt{Z}}{G} \sum_{j=1}^{N} \frac{q_j}{\sqrt{d_J}} f_{k,j}$ where Z, d_J, $f_{k,j}$ come from the stored factor details in the model database and $G = \sum_{j=1}^{N} q_j$. The quantity Q_k represents the scalar coordinate of the query Q along factor F_k; 3: Consider the query projections Q_K in the factor space spanned by the first few *dominant* factors used for clustering and indexing. Classify the query to one of the clusters based on nearest-neighbor classification. Let C_m be the cluster; 4: Consider the two key objects of the cluster C_m. Compute the distance (in factor space) of the query object from each of these two key objects. Using these distances and the stored distances of the other model objects from the two key objects, retrieve the model objects whose TI-

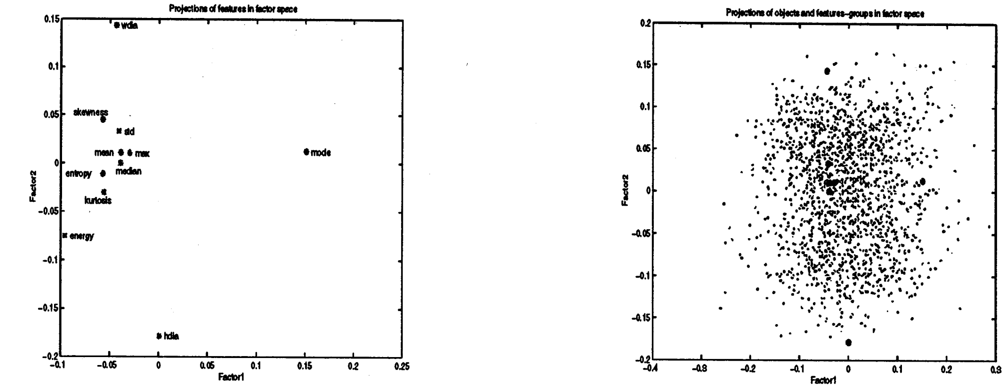


Figure 3: Left: Projection of 11 features (features 10, 11 and 12 are not used; see text for explanation) in factor space spanned by first two dominant factors for the entire *Database* 1 (see Table 2). Right: Projection of both objects and feature-groups in the same factor space for *Database* 1. The features are marked $*$ and the objects (targets) are marked by $+$ in the factor space. Note that the objects appear to be almost randomly spread out in the factor space. The four feature-groups are used as the seeds for forming four clusters of objects using nearest-neighborhood approach.

based lower bounds are less than a threshold Th^3. These initial retrievals form candidate matching objects to the query. The candidate models are further ranked using their mutual information (MI) with the query object [8].

4 Experimental Results

In this section, we present the experimental results on the MSTAR public real SAR image databases. Various *model databases* consist of SAR images of (*i*) objects at a particular depression angle, or (*ii*) objects of a particular configuration, or (*iii*) objects at a particular articulation. The corresponding *test data* consists of SAR images of (*i*) objects at a different depression angle, or (*ii*) objects of a different configuration, or (*iii*) objects at a different articulation. In each case, test data is an independently acquired one from which query objects are selected randomly.

The set of features we have used to represent each object in CA are listed in Table 1, where HRD is histogram of relative distances between scatterers, wdia is diameter of the object along width of the image, HI is histogram of intensity values and hdia is diameter of the object along height.

The first nine features are computed on the histogram of relative distance between scatterrers. Since the relative distances are used, the first nine

[3]This threshold can be *a priori* estimated as the average distance in factor space between pairs of similar objects, by considering a large set of similar object pairs.

Table 1: 14 features used to represent SAR targets in correspondence analysis.

Feature Number	Feature Description
1	mean of HRD
2	std of HRD
3	max of HRD
4	mode of HRD
5	median of HRD
6	skewness of HRD
7	kurtosis of HRD
8	energy of HRD
9	entropy of HRD
10	mean_{HI}
11	std_{HI}
12	max_{HI}
13	wdia
14	hdia

features are translation invariant. The next three features are computed on the gray level histogram of the SAR image pixels within ROI. While we compute the features 10 to 12 based on histogram of intensity values and use them in CA, we do not consider them in clustering and indexing. This is because they are computed based on gray level values of only the top scatterers which may be unreliable. The last two features are also translation invariant. In all our experiments, we consider only 11 of the 14 features (discounting features 10 to 12) from Table 1, for clustering and indexing.

Figure 3 (left) shows the projections of features in the factor space spanned by the first two dominant factors for *Database* 1 (see Table 2). Note that the *mean*, *std*, *max* and *median* fall closely in factor space and are redundant for describing the

Table 2: Results on various SAR databases

Model/Test Data Differences	Total no. of model objects	Total no. of test objects	% of total variance explained by first 2 factors	No. of random test queries	Avg. no. of retrievals after indexing	No. of cases with correct ID
Depr. angle differences (*Database* 1)	1621	1351	85.31	200	84 (5.18%)	185 (92.5%)
Configuration differences (*Database* 2)	694	1621	70.84	200	64 (9.22%)	170 (85%)
Articulation differences (*Database* 3)	606	239	78.29	200	96 (15.84%)	139 (69.5%)

target database, whereas, the *mode*, *wdia* and *hdia* fall distant to any other feature and hence form independent descriptors of the database. The other features look smeared in the factor space and so we do not use them in clustering. Thus, we form four independent feature-groups: the first consists of (*mean*, *std*, *max* and *median*), and the rest consist of just one feature each (*mode*, *wdia* and *hdia*, respectively). Figure 3 (right) shows the projections of both objects and the feature-groups in the factor space spanned by the first two dominant factors for the same databases. The centroids of the four feature-groups are used as the seeds for nearest-neighbor clustering of the objects. The results of using our approach are summarized in Table 2.

5 Conclusions

We have presented a database-retrieval oriented approach to model-based object recognition, for large model databases. We have presented detailed results on several real SAR target databases to demonstrate our approach. These are difficult databases since feature invariance may be small (20% to 50% [1]). The technique is general and has other interactive object recognition applications involving large model databases. It also helps interactive discovery and acquisition of new models to update database.

Currently, mutual information based ranking takes more than 99% of the time during recognition (retrieval) phase. The performance of our approach could be further improved by: (a) efficient computation of mutual information between a query and a candidate model object using a coarse-to-fine strategy, (b) overlapped boundaries in nearest-neighbor clustering method, and (c) efficient methods to adapt factors to dynamically changing databases. We are currently investigating along these directions.

References

[1] B. Bhanu, G. Jones, and J. Ahn. Recognizing articulated objects and object articulation in SAR images. In *SPIE Proceedings: Algorithms for Synthetic Aperture Radar Imagery V*, volume 3370, April, 1998.

[2] T. Pun and D. Squire. Statistical structuring of pictorial databases for content-based image retrieval systems. *Pattern Recognition Letters*, 17:1299–1310, 1996.

[3] J.-P. Benzécri. *Correspondence Analysis Handbook.* Marcel Dekker, New York, 1992.

[4] M. Jambu. *Exploratory & Multivariate Data Analysis.* Academic Press, New York, 1991.

[5] J. F. Barros, W. Martin, P. Kelley, and M. Cannon. Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval. In *Proc. of IS&T/SPIE – Storage and Retrieval for Still Image and Video Databases*, volume 4, pages 392–403, January, 1996.

[6] A. Berman and L. G. Shapiro. Selecting good keys for triangle-inequality-based pruning algorithms. In *Proc. of IEEE Intl. Workshop on Content-based Access of Image and Video Databases (CAIVD'98)*, pages 12–19, January, 1998.

[7] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. In *Proc. of 5th Intl. Conf. on Computer Vision (ICCV'95)*, pages 16–23, 1995.

[8] Y. L. Moon, B. Rajagopalan, and U. Lall. Estimation of mutual information using kernel density estimators. *Physical Review E: Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 52(3):2318–2321, 1995.

Bounding Fundamental Performance of Feature-Based Object Recognition*

Michael Boshra and Bir Bhanu
Center for Research in Intelligent Systems
University of California, Riverside, California 92521
{michael,bhanu}@cris.ucr.edu
http://www.cris.ucr.edu/

Abstract

Performance prediction is a crucial step for transforming the field of object recognition from an art to a science. In this paper, we address this problem in the context of a vote-based approach for object recognition using 2-D point features. A method is presented for predicting tight lower and upper bounds on fundamental performance of the selected recognition approach. Performance bounds are predicted by considering data-distortion factors, which are uncertainty, occlusion and clutter, in addition to model structural similarity. Given a statistical model of data uncertainty, the structural similarity between every pair of model objects is computed as a function of the relative transformation between them. Model-similarity information is then used along with statistical data-distortion models to predict bounds on the probability of correct recognition. Validity of the method is experimentally demonstrated using MSTAR public SAR data.

1 Introduction

The problem of object recognition is concerned with identifying and localizing model objects from scene data. It involves searching for a consistent correspondence between scene features, and those of a model object. Performance of such a process depends on a large number of factors, which are associated with either scene data (e.g., sensor noise, missing and spurious features), or model objects (e.g., size of model database, similarity of model objects, articulation of model parts). Predicting the performance of object recognition as a function of these factors is a challenging task.

In this paper, we address the problem of performance prediction in the context of an approach for object recognition using 2-D point features. Such an approach uses a vote-based matching criterion, which ranks object/pose hypotheses based on the number of model features (votes) that are consistent with scene features. We predict recognition performance of this approach, by considering the following factors: *1) Scene-Data Factors:* uncertainty (due to sensor noise and imperfections of the feature-extraction process), occlusion (missing features), and clutter (extraneous features), *2) Model-Object Factors:* similarity (degree of structural overlap between pairs of model objects), and number of model objects (this factor is implicitly considered in our handling of object similarity).

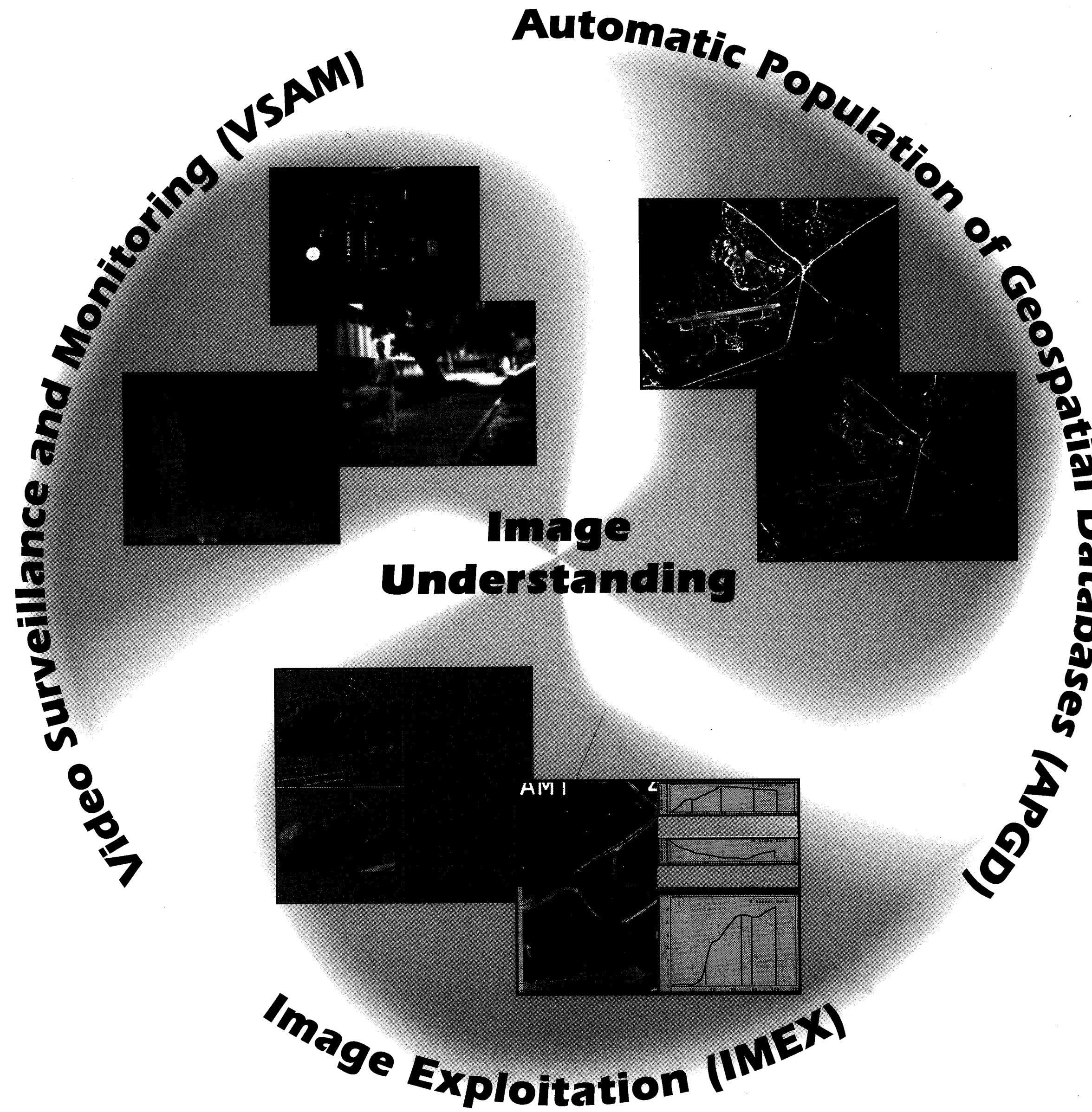
Problem Definition: Our performance-prediction problem can be defined as follows. We are given: *a)* a set of model objects, $\mathcal{M} = \{\mathcal{M}_i\}$, where each object $\mathcal{M}_i$ is represented by a set of 2-D point features, $\{F_{ik}\}$, that are discretized at some resolution, *b)* statistical models for data distortion (uncertainty, occlusion, and clutter), and *c)* a class of applicable transformations, $\mathcal{T}$ (e.g., translation, rigid, affine). Our objective is to predict tight lower and upper bounds on the probability-of-correct-recognition (PCR) plot, as a function of occlusion and clutter rates (assuming a fixed uncertainty model). The performance predicted by our method is fundamental, since it is obtained by analyzing the amount of information provided by both scene data and model objects, *independent* of the vote-based recognition algorithm used. Thus, it sets an upper bound on performance that is achievable by any recognition algorithm that uses the same matching criterion.

*This work was supported in part by DARPA/AFOSR grant F49620-97-1-0184; the contents and information do not reflect positions or policies of the U.S. Government.

Proceedings Volume II

1998 Image Understanding Workshop

Hyatt Regency Monterey
November 20–23
Monterey, California



Hosted by SRI International

Edited by
George E. Lukes

Sponsored by
Defense Advanced Research Projects Agency
Information Systems Office