1-1-2008

Association-based image retrieval for automatic target recognition.

Arun D. Kulkarni
University of Texas at Tyler, akulkarni@uttyler.edu

H. Gunturu

S. Dalta

Association-Based Image Retrieval for Automatic Target Recognition

ARUN KULKARNI, HARIKRISHA GUNTURU, AND SRIKANTH DATLA
Computer Science Department
The University of Texas at Tyler, Tyler, TX 75799

Abstract: Model-based automatic target recognition (ATR) systems deal with recognizing three dimensional objects from two dimensional images. In order to recognize and identify objects the ATR system must have one or more stored models. Multiple two dimensional views of each three dimensional object that may appear in the universe it deals with are stored in the database. During recognition, two dimensional view of a target object is used a query image and the search is carried out to identify the corresponding three dimensional object. Stages of a model-based ATR system include preprocessing, segmentation, feature extraction, and searching the database. One of the most important problems in a model-based ATR system is to access the most likely candidate model rapidly from a large database. In this paper we propose new architecture for a model-based ATR system that is based on association-based image retrieval. We try to mimic human memory. The human brain retrieves images by association. We use generalized bi-directional associative memories to retrieve associated images from the database. We use the ATR system to identify military vehicles from their two dimensional views.

Key-Words: Automatic Target Recognition, Model Representation, Association-Based Image Retrieval, Bi-directional Associative Memories.

1. Introduction

Model-based automatic target recognition (ATR) systems deal with recognizing three dimensional objects from two dimensional images. In order to recognize and identify objects, the ATR system must have one or more stored models that may appear in the universe it deals with. The stages of a model-based ATR system include preprocessing, detection, segmentation, and searching a database. One of the important problems in a model-based ATR system is to access the most likely candidate model rapidly from a large database. Content-based image retrieval (CBIR) systems also deal with the problem of searching images that are similar to the query image in a large database. Most CBIR systems use low-level features such as color, texture, and shape features for image indexing. Shape features based on Fourier descriptors, moment invariants have been used in conjunction with color and texture features. Smelders et al. [8] consider the description of image content in two steps. The first step deals with image processing operations that transpose the image data array into another spatial data array that can include methods over local color, local texture, or local geometry. The second step is to extract invariant features. Rui et al. [7] proposed a relevance feedback based interactive retrieval approach. In their approach during the retrieval process, the user's high-level query and subjectivity are captured by dynamically updated weights based on the user's feedback. In this paper, we suggest a new approach for searching images in a database that is based on association-based image retrieval. We try to mimic the human brain. Association is one of the fundamental characteristics of the human brain. The human memory operates in associative manner; that is, a portion of recollection can produce an associated stream of data from the memory. The human memory can retrieve a full image from a partial or noisy version of the image as the query image. Furthermore, given a query image as the input, the human brain can recall associated images that have been stored in the past. The human memory can respond to abstract queries. For example, if we see an image of a person, we can recall images of his house, spouse, and car. The associative storage and retrieval mechanism is not explored in the present ATR systems. Stages in a typical CBIR system include annotation, preprocessing, and feature extraction. In order to store an image it is first annotated. The preprocessing stage deals with geometric and radiometric corrections, mapping the image from red, blue, green (RBG) color space to the hue, saturation, and intensity (HIS) color space. In the feature extraction stage, features based on attributes such as the color, texture, and/or shape are extracted. Most CBIR systems use color histograms to compare the query image with images that are stored in the database. Often color histograms alone are not sufficient to retrieve desired images from the database, because a single color histogram may represent multiple images in the database. There are many CBIR systems that use texture and shape features in addition to color features [2, 5, 8, 9]. The most commonly used similarity function is the Minkowski distance that is given by Equation (1).

$$d\left(\mathbf{f}^q,\mathbf{f}^d\right) = \left[\sum_{j=1}^{n}\left|f_j^q - f_j^d\right|^2\right]^{\frac{1}{2}} \qquad (1)$$

Where $\mathbf{f}^q$ and $\mathbf{f}^d$ represent the features vectors corresponding to the query image and the image in the data base, and n is the number of features. The query image is compared with images in the database and the images in the database are ranked based on the similarity measure. Images that are similar to the query image are retrieved and displayed. While it is feasible to retrieve a desired image from a small collection by exhaustive search, techniques that are more effective are needed with a larger database. The well-known indexing technique is used for efficient retrieval. Other features of a typical CBIR system include defining query feature space and displaying query results. In this paper, we propose a new architecture for a model-based ATR system. Our architecture is based on a generalized associative memory (GBAM).

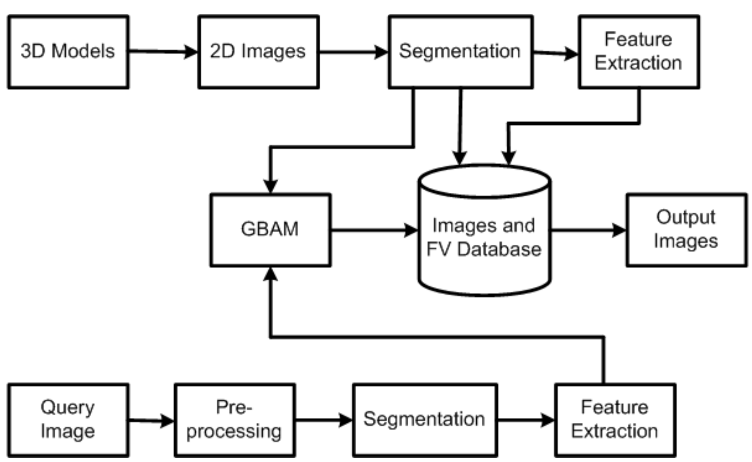

Figure 1. Architecture for a model-based ATR system

2. Methodology

The architecture for the proposed model-based ATR system is shown in Figure 1. It can be seen from Figure 1 that there are two data paths. The first path deals with storing multiple views of an object model. For a given three-dimensional object, two-dimensional images are obtained by viewing the model from different angles. The two-dimensional views pass through the segmentation stage that separates the objects from the background. The feature extraction stage generates invariant features representing the shape. The feature vectors and two-dimensional images are then indexed and stored in the database. It may be noted that for each object. The stimulus and response feature vectors represent correspond to the query image and associated images in the database. Bidirectional associative memories (BAMs) have studied by Kosko [4]. A two-layer network that simulates a BAM is shown in Figure 2. The second path deals with retrieving associated two-dimensional views from the database. The query image represents a two-dimensional view of an object. The query image goes through the pre-processing stage that deals with radiometric and geometric corrections. The pre-processed image is then segmented and a feature vector is obtained from the segmented image. The extracted feature vector is fed as the stimulus vector to the GBAM. The output vectors of the GBAM are used to recall associated two-dimensional views from the database. The most important feature of the proposed models-based ATR system is that we use GBAM to store and retrieve images from the database. The basic functions of the GBAM are to store associative pairs through self-organizing process and to produce appropriate feature vectors that represent the strengths of the GBAM. The network is designed to map stimulus vectors

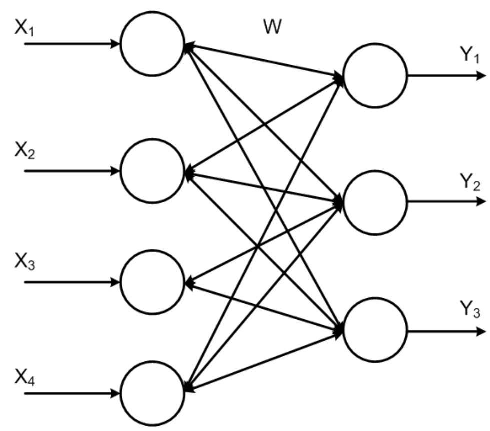

Figure 2. Bi-directional associative memory.

$\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$ to response vectors $\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_n$. In an auto-associative network, the response vector $\mathbf{y}_i$ and the corresponding stimulus vector $\mathbf{x}_i$ are the same. In a hetero-associative memory, the stimulus and response vectors are not identical. Associative memories are often able to produce correct response patterns even though stimulus patterns are distorted or incomplete. Conventional BAMs are used to store are retrieve pairs of stimulus and response vectors. However, if the number of associated inputs and/or outputs is more than two; that is, instead of pairs of vectors, if we want to store triplets or quadruplets of vectors, then we need to use generalized BAMs. Humpert [1] has suggested generalization of the BAM that can store associated multiple input/output patterns. We use the GBAM with a tree topology for the proposed ATR system. Pairs of vectors $\mathbf{x}_i, \mathbf{y}_i$ can be stored with the BAM by summing bipolar correlation matrices. If the input vectors are orthonormal then the recall is perfect. If the input vectors are not orthonormal, then the output vector may contain cross talk. If we assume a nonlinear transfer function for neurons in the BAM, then the recalled output is a nonlinear function of a transformed input vector and is given by

$$\mathbf{y}_i = F\ \mathbf{W}\mathbf{x}_i \qquad (2)$$

With the feedback the input vector $\mathbf{x}_i$ can be estimated as

$$\mathbf{x}_i = F\ \mathbf{W}^T\mathbf{y}_i \qquad (3)$$

The simplest transfer function for the BAM is a step function. The stable reverberation corresponds to the system energy local minimum. When the BAM neurons are activated, the network quickly evolves to a stable state of two-pattern reverberation or a non-adaptive resonance. A number of associations can be stored by adding corresponding correlation matrices.

$$\mathbf{W} = \sum_{i=1}^{N} \mathbf{y}_\mathbf{i}\mathbf{x}_\mathbf{i}^\mathbf{T} \qquad (4)$$

In order to recall vector $\mathbf{y}_i$, we can use the stimulus vector $\mathbf{x}_i$ as the input vector. If input pattern vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_n$ are orthonormal i.e.

$$\mathbf{x}_i\mathbf{x}_j = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ for } i \neq j \end{cases} \qquad (5)$$

then the recall is perfect. If the input vectors are not orthonormal, then the output vector may contain cross talk. In a dual BAM, feedback is achieved with $\mathbf{W}^\mathbf{T}$ and is given by

$$\mathbf{W}^\mathbf{T} = \sum_{i=1}^{N} \mathbf{y}_\mathbf{i}\mathbf{x}_\mathbf{i}^\mathbf{T}\ ^\mathbf{T} = \sum_{i=1}^{N} \mathbf{x}_\mathbf{i}\mathbf{y}_\mathbf{i}^\mathbf{T} \qquad (6)$$

The simplest transfer function for the BAM is a step function. The stable reverberation corresponds to the system energy local minimum. When the BAM neurons are activated, the network quickly evolves to a stable state of two-pattern reverberation or a non-adaptive resonance. In order to improve recall accuracy, the output vector $\mathbf{y}_\mathrm{i}$ can be synchronously fed back. The back-and-forth flow of distributed information quickly resonates on a fixed data pair. Humpert [1] has suggested generalization of the BAM that can store associated multiple input/output patterns. The generalized BAM with tree topology is shown in Figure 3.

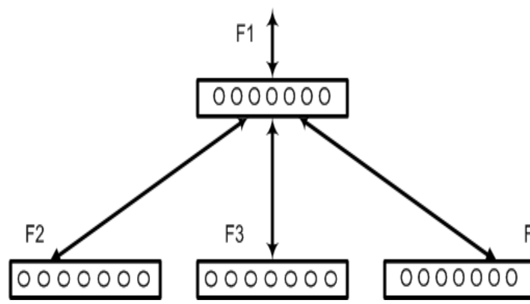

Figure 3. Generalized BAM- Tree Structure

3. Results and Discussions

We have developed software to store and retrieve feature vectors. In our simulation, we have considered three sets of test images of characters {1, 2, 3}, {A, B, C}, and {α, β, γ}. Each character was represented by a 12x12 matrix. Each image was represented by a feature vector of 144 elements. These feature vectors are stored in the generalized BAM. During retrieval, any one image (partial or noisy) from any set was used as the query image, and the corresponding images from the other sets were retrieved. The results are shown in Figures 4 and 5. The first row shows the query images and subsequent rows show corresponding retrieved images. The generalized BAM stores and recalls feature vectors. In present simulation, we used a tree topology for storing three sets of images. The first set contained images of numbers, the second set contained images of characters, and the third set contained images of Greek characters [6].

In the second example, we have used the system to store and retrieve multiple views of military vehicles. The system can be used for automatic target recognition. In this example, we have considered images of military vehicles such as jeeps, tanks, and HUMVEES as shown in Figure 5. We have used the system to store images of four views of each vehicle such as side, rear, front, and at an angle. The system retrieves images similar to the human brain. For example, if we see a front view of a vehicle, our mind can retrieve side and rear views of the same vehicle. We have used the GBAM with a tree structure to store and retrieve these associations. The units in the root node represent the reference vectors, and the units in leaf nodes F_1, F_2, and F_3 represent front, side and rear views, respectively. Figure 6 shows the images stored in the database, and Figure 7 shows the query and output images. In a feature extraction stage, we have used histograms of red, blue, and green components of each image to generate a binary feature vector of 192 bits. In order to extract a feature vector, we divide each histogram in sixteen bins, and use four bits to represent the number of pixels in each bin. We can also use feature vectors that may represent color, texture and/or shape of objects in the image.

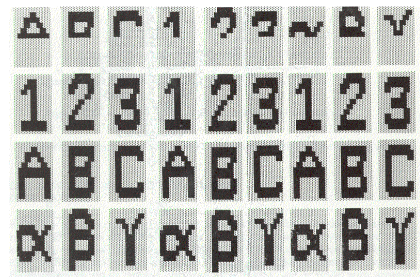

Figure 4. Partial input and recalled patterns

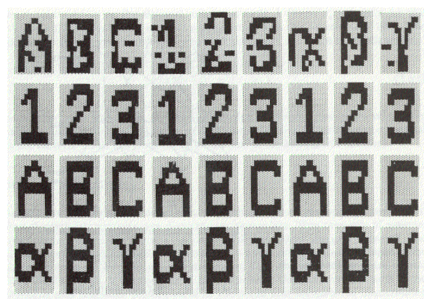

Figure 5. Noisy input and recalled patterns

Figure 6. Image retrieval-Example 1

Figure 7. Retrieved images-Example 2

4. Conclusions

In this paper, we have proposed architecture for a model-based ATR system that is based on the ABIR. The system has been used successfully to recognize military vehicles. In illustrative examples, we have used a GBAM with a tree topology. However, a GBAM with other topologies such as the star, bus, or ring can be used. The bus topology is more suitable for retrieving temporal images. The generalization of a BAM to several vector fields raises questions regarding the updating process. In a BAM, all units in a field are synchronously updated. By contrast, the sequence of updating weights in a GBAM is not obvious. The generalization of a BAM to several fields also raises the question of interconnections. In addition, one needs to consider the capacity of the generalized BAM. The number of images that can be stored and retrieved depends on the capacity of the GBAM and the size of the feature vector. The GBAMs are useful in association-based image storage and retrieval for multimedia applications.

Acknowledgements

The authors are thankful to anonymous reviewers for their valuable comments.

References

[1] B. Humpert, Bi-directional associative memory with several patterns, *Proceedings of International Joint Conference on Neural Networks, San Diego*, vol. 1, 1990, pp. 741-750.

[2] A. K. Jain and A. Vailaya, Shape-based retrieval: A case study with trademark image databases, *Pattern Recognition,* vol. 31, no. 9, 1998, pp 1369-1390.

[3] T. Kohonen, *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1988.

[4] B. Kosko, Bi-directional associative memories, *IEEE Transactions on Systems, Man and Cybernetics* vol. 18, 1988, pp 49-60.

[5] R. Krishnapuram, S. Medasani, S. Jung, Y. Choi, and R. Balasubramanian, Content-based image retrieval based on a fuzzy approach, *IEEE Transactions on Knowledge Engineering,* vol. 16, no. 10, 2004, pp 1185-1199.

[6] A. D. Kulkarni, *Computer Vision and Fuzzy-Neural Systems*. Prentice Hall, Upper Saddle River, NJ, 2001.

[7] Y. Rui, T. S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, vol 8, no. 3, 1998, pp 644-655.

[8] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain , Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Recognition and Machine Learning,* vol. 22, no. 12, 2000, pp 1349-1380.

[9] X. Wan and C. C. J. Kuo, A new approach to image retrieval with hierarchical color clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, vol 8, no. 3, 1998, pp 628-64.