2002 16th International Conference on Pattern Recognition

Vol. 3 (B)

Content-based Image Retrieval Using Fourier Descriptors on a Logo Database*

Andre Folkers
Institute for Signal Processing
Medical University of Lübeck
23569 Lübeck, Germany
E-mail: folkers@isip.mu-luebeck.de

Hanan Samet
Computer Science Department
University of Maryland at College Park
College Park, Maryland 20742
E-mail: hjs@umiacs.umd.edu

Abstract

A system that enables the pictorial specification of queries in an image database is described. The queries are comprised of rectangle, polygon, ellipse, and B-spline shapes. The queries specify which shapes should appear in the target image as well as spatial constraints on the distance between them and their relative position. The retrieval process makes use of an abstraction of the contour of the shape which is invariant against translation, scale, rotation, and starting point that is based on the use of Fourier descriptors. These abstractions are used in a system to locate logos in an image database. The utility of this approach is illustrated using some sample queries.

1. Introduction

A basic requirement of an image database is to perform content-based searches for images. In [9] we presented a pictorial query specification tool for objects represented by points. In this paper we extend this tool to permit query primitives that have a spatial extent such as ellipses, rectangles, polygons, and B-splines. Breaking down the image in terms of these primitives can be viewed as a classification approach. At times, it could be the case that the objects in the image cannot be decomposed into components that are comprised of these primitive shapes. In this case, we need an abstraction of the contour of the shape which ideally is invariant against translation, scale, rotation, and starting point, while still representing the essential form of the contour. We propose a method based on Fourier descriptors to achieve this effect. A comparison with other shapes representations, e.g. with invariant moments was made in [4]. In the rest of this paper we describe our query specification and shape description methods, and study their effectiveness by querying a database of logos obtained from the US patent office (e.g., Figure 2(c)). For related work with logos, see [1, 7, 12] where the goal is to locate a logo in the database based on a sketch of the logo, or with the introduction of noise, or variations in orientation or scale. In contrast, our goal is to find logos whose components consist of user-defined shapes, while also providing a way of specifying the extent of the similarity.

*This work was supported in part by the National Science Foundation under Grants EIA-99-00268, IIS-00-86162, and EIA-00-91474.

2. Image Processing

The shapes in a logo image are preprocessed by morphological closing and opening operations using structure elements of different sizes. This either disconnects components connected only by a few pixels (opening) or connects components which are only a few pixels apart from each other (closing). Furthermore, the closing or opening smooths the contours of the shapes.

Each image component is approximated with each of the following shape types: rectangle, ellipse, polygon, and B-spline. Next, the quality of these approximation shape types is evaluated, and the image component is classified as the shape type that approximates it best. In case of a tie, the approximation shape type with the smaller number of parameters is given priority, e.g., classification as an ellipse is preferred to classification as closed B-spline. Currently we can only handle the rectangle, ellipse, polygon, and B-spline shapes. Future work includes recognizing a straight line, arc, chord, pie slice, and freehand drawing shapes.

3. Logo database

The components of the logo images are stored using a prototype spatial database system called SAND (spatial and non-spatial data) developed at the University of Maryland [2]. In SAND, data is stored in tuples consisting of attributes for geometric entities such as points, lines, polygons, etc. (some of which are supported for arbitrary number of dimensions), in addition to traditional ones like integers, floating point numbers, and character strings. SAND can index both spatial – including high-dimensional points – and non-spatial data using different methods (e.g., PMR-trees, R-trees, kd-trees [8]). The central relation is the table of extracted image components. For each image component, we store the result of the classification which is one of the approximation shape types described above, a minimum

1051-4651/02 $17.00 © 2002 IEEE

bounding box as a rough approximation of the spatial extent of the component, and a reference to the image from which the component was extracted. The possible classifications are represented by the query primitives (e.g., a polygon, ellipse, etc.). In Section 5 we describe how pictorial queries are specified using these query primitives. All query primitives can have individual spatial representations, e.g., the representation of a polygon consists of a list of points, while an ellipse can be represented using a rectangle. Thus, we have different relations to store the spatial representation of image components.

In some cases, it may be impossible to classify the image components using one of the approximation shape types. In this case, we represent the component as a feature vector, termed its *abstraction* in contrast to its *classification*. The database contains both the classification and the abstraction for each image component, where in some cases the classification may be unknown. The advantage of storing both the classification and abstraction in the database for each component of a logo image is that this provides more flexibility for the query processing as the user may want to use the abstraction for a particular query image component instead of its classification (see [10] for more details).

4. Fourier Descriptors

Each image of the logo database is decomposed into its image components, i.e., its connected components. As an abstract representation of image components, we use their Fourier descriptors, which are made invariant against translation, scale, rotation, and their starting point. We retain the phase which contains essential information about the contour of the image component.

4.1. Definition and Properties

Consider the N contour points of an image component as a discrete function $\mathbf{x}(n) = (x_1(n), x_2(n))$. Using this function, we can define a discrete complex function $u(n)$ as

$$u(n) = x_1(n) + jx_2(n).$$

u(n) can be transformed into the frequency domain by the *Discrete Fourier Transformation* (DFT). The result can be transformed back into the spatial domain via the *Inverse Discrete Fourier Transformation* (IDFT) without any loss. DFT and IDFT are defined as $a(k)$ and $u(n)$, respectively:

$$a(k) = \frac{1}{N}\sum_{n=0}^{N-1} u(n)\, e^{-j2\pi kn/N} \quad k = -N/2, \ldots, N/2-1$$

$$u(n) = \sum_{k=0}^{N-1} a(k)\, e^{j2\pi kn/N} \qquad n = -N/2, \ldots, N/2-1$$

The coefficients $a(k)$ are also called Fourier descriptors [6]. They represent the discrete contour of a shape in the Fourier domain.

Certain geometric transformations of the contour function $u(n)$ can be related to simple operations in the Fourier domain. Translation by $u_0 \in \mathbb{C}$ affects only the first Fourier descriptor $a(0)$, while the other Fourier descriptors retain their values. Scaling of the contour with a factor α leads to scaling of the Fourier descriptors by α. Rotating the contour by an angle θ_0 yields a constant phase shift of θ_0 in the Fourier descriptors. Changing the starting point of the contour by n_0 positions results in a linear phase shift of $2\pi n_0 k/N$ in the Fourier descriptors [6].

4.2. Normalization

The contour functions are made invariant against translation by setting the first Fourier descriptor $a(0)$ to zero which moves the centroid of the contour onto 0. Since the contours are traced counterclockwise and describe a nonzero area, we can rely on the fact that the second Fourier descriptor $a(1) = r_1\, e^{j\varphi}$ is nonzero [5] (tracing it clockwise would imply that $a(-1)$ is nonzero for a contour with nonzero area). Therefore, we can divide all Fourier descriptors by the magnitude of the second Fourier descriptor to obtain a scale invariant vector: $a(k) = a(k)/|a(1)|$.

Rotation invariance could be achieved by simply taking the magnitude of each Fourier coefficient, but hereby an essential part of the information about the contour is lost. To achieve rotation invariant Fourier descriptors that still represent the shape of the original contour, we can use the orientation of the basic ellipse, which is defined by the Fourier descriptors $a(1) = r_1\, e^{j\varphi_1}$ and $a(-1) = r_{-1}\, e^{j\varphi_{-1}}$, as

$$u_e(n) = a(-1)\, e^{-j2\pi n/N} + a(1)\, e^{j2\pi n/N}.$$

With a few transformations, and with the abbreviation $\tilde{\varphi} = (\varphi_1 - \varphi_{-1})/2$, we get

$$u_e(n) = e^{j(\varphi_{-1}+\varphi_1)/2} \left[r_{-1}\, e^{-j(\tilde{\varphi}+2\pi n/N)} + r_1\, e^{j(\tilde{\varphi}+2\pi n/N)} \right]$$

which shows the rotation φ_e of the basic ellipse as:

$$\varphi_e = (\varphi_{-1} + \varphi_1)/2.$$

Using the orientation of the basic ellipse leads to an ambiguity of π radians. Therefore, the 'rotation invariant' Fourier descriptors are only rotation invariant modulo a rotation by π radians. Figure 1 illustrates this for a set of equal contours at different scales and rotations which are normalized by the above operations.

Now, only the position of the starting point remains to be normalized. This can be done by subtracting the phase of the second Fourier descriptor, weighted by k, from the phase of all Fourier descriptors, that is,

$$a(k) = a(k)\, e^{-j\varphi_1 k}.$$

After this normalization, the starting point is approximately at angle 0.

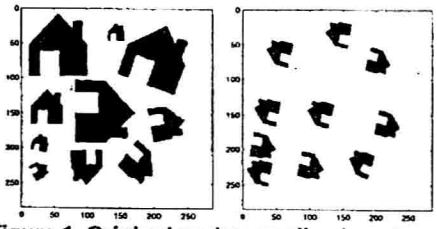


Figure 1. Original and normalized contours.

5. Pictorial Query Specification

We briefly review how individual pictorial queries are specified using our method. For more details and examples, see [3, 11]. The *matching similarity level* (*msl*) is a value between 0 and 1. Components in the database are considered to match a query component if they are similar with respect to a certain similarity measurement (see Section 6). The *extent similarity level* (*esl*) is an upper bound for the difference in spatial extent. The *contextual similarity level* (*csl*) specifies how well the content of a database image D matches that of a query image Q. The choices are:

1. All components, and no others.
2. All components, and maybe others.
3. Any of the components, but no others.
4. Any of the components, and maybe others.

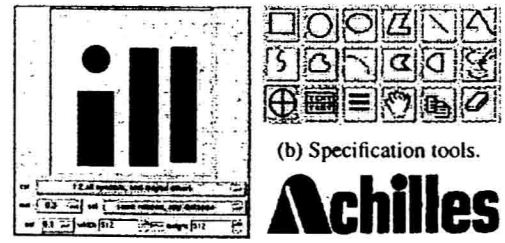
The *spatial similarity level* (*ssl*) specifies how close the database image D and the query image Q are with respect to distance and directional relation between the symbols in the query. We distinguish between five different levels which are defined as

1. exact same location
2. same relation, bounded distance
3. same relation, any distance
4. any relation, bounded distance
5. any relation, any distance.

A pictorial query consists of query primitives. The user selects the desired query primitive using check buttons (see Figure 2(b)). Next, an instance of the query primitive can be drawn onto the canvas (see Figure 2(a)). Currently, the implementation supports the rectangle, circle, ellipse, polygon, and closed B-Spline query primitives. Note, that circles are included into the interface in order to help the user to draw real circles. During the processing phase they are handled as ellipses. Query primitives can be positioned arbitrarily. The matching, contextual, spatial, similarity, and extent level for the query is defined using combo boxes (see the bottom of Figure 2(a)).

6. Pictorial Query Processing

A query can be processed in either classification or abstraction mode. In the classification mode, the database is searched for the specified query primitives. In order to check the extent similarity of an image component and a query primitive we compute the factors α_1 and α_2 that scale the respective query primitive and image component so that it fits into a rectangle of width 1. Then the difference is determined as $|\alpha_1 - \alpha_2|$ which is required to be less than *esl*. Similarity between two ellipses or two rectangles is measured using the aspect ratio of their minimum bounding boxes. Similarity of two polygons is measured using the ratio between the area of the intersection polygon and the maximum area of both polygons. This ratio becomes 1 if the polygons are equal, and is zero if they do not intersect. Similarity of two closed B-Splines is measured using the same method, whereby the control points are interpreted as defining polygons. This makes sense as a B-spline is uniquely determined by its control polygon.



(a) Specification canvas. (b) Specification tools. (c) Result.

Figure 2. Query Specification and result.

Due to the ambiguity of π radians in the 'rotation invariant' Fourier descriptors that was described in Section 4, we maintain two Fourier descriptors in the abstraction mode for each component in the query image after which the starting point of both vectors is normalized. Next, the database is searched for both vectors. For this, we use a spatial index provided by SAND to perform a fast search for vectors in the database which are at a small Euclidean distance from the vectors of the query component. The maximum distance for the nearest neighbor search is determined by the value of *msl*. Currently we use 16 Fourier descriptors ($k = -8, \ldots, -1, 2, \ldots, 9$), which result in 32-dimensional vectors (as we have both real and imaginary parts, and we ignore $a(0)$ and $a(1)$ since they are 0 and 1, respectively, due to normalization). The factors α_1 and α_2 are given by the first Fourier descriptor of the respective component as $1/|a(1)|$, and the extent similarity check is done as described above.

After a set of matching images has been found, the contextual similarity and spatial similarity check is performed. This has been described in detail in [3].

7. Sample Queries and Results

Figure 2(a) is an example of a pictorial query consisting of four shapes: a circle and three rectangles. These shapes are required to have the same spatial relation as ssl=3. The contextual similarity level (i.e., 2) stipulates that all compo-

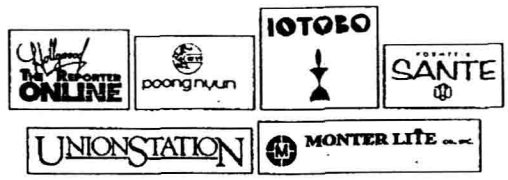
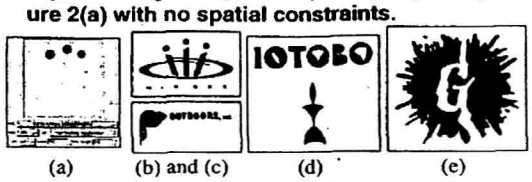


Figure 3. Logos retrieved by the query in Figure 2(a) with no spatial constraints.



(a) (b) and (c) (d) (e)

Figure 4. (a) Sample query with three circles, and (b)–(e) the logos that satisfy the contextual constraint of containing three circles.

nents in the query image must occur in the database image, while the database image may contain other components. The extent similarity level is set to 0.1 which allows quite different sizes for the matching components. Figure 2(c) shows the result of this query.

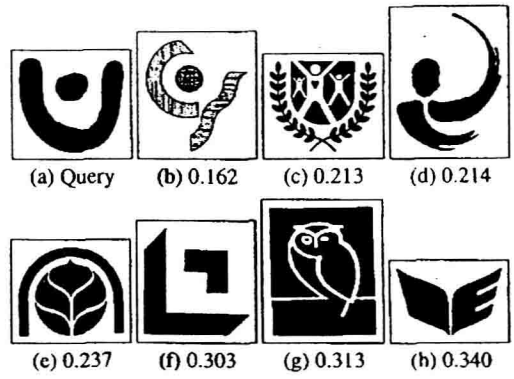
Figure 3 shows six logos that satisfy the contextual constraint of containing at least three rectangles and one circle for the query in Figure 2(a). The logo in Figure 2(c) also fulfills it. This logo is the only one in the database that also satisfies the query's spatial constraint.

Figure 4(a) is another query. It contains three circles. csl=2 and ssl=3 (i.e. the components may have any distance but the directional relation must be similar). Figure 4(b)–(e) shows some logos that only satisfy the contextual constraints. Closer scrutiny of Figure 4(e) reveals the absence of any circles. The shapes are obviously misclassified. The other two result logos (Figures 4(d) and 4(c)) actually contain three circles. The best result of the query is shown in Figure 4(b). Here we find the circles in the desired spatial configuration.

Finally, Figure 5 shows an example query processed in abstraction mode and the retrieved logos. It contains a u-bend and a circle. csl=2 and there are no spatial constraints. The sizes are supposed to be similar. The result images are ordered by their matching rank, i.e., Figure 5(b) has first rank and is the best match. All result logos actually contain a u-bend and a circle, except for the one in Figure 5(h). Note, that the u-bend occurs at different rotations.

8. Concluding Remarks and Future Work

Both classification and abstraction queries produced acceptable results. The main advantages of the abstraction approach compared with the classification approach is that we have a greater flexibility in choosing an arbitrary shape of the components of the query image. Future work involves



(a) Query (b) 0.162 (c) 0.213 (d) 0.214

(e) 0.237 (f) 0.303 (g) 0.313 (h) 0.340

Figure 5. (b)–(h)Logos that are retrieved based on Fourier descriptors using query image (a), msl=0.6, csl=2, esl=0.01, and ssl=5.

extending the pictorial query specification tool to incorporate other shapes as well as freehand shapes.

References

[1] D. Doermann, E. Rivlin, and I. Weiss. Applying algebraic and differential invariants for logo recognition. *Machine Vision and Applications*, 9(2):73–86, 1996.

[2] C. Esperança and H. Samet. An overview of the SAND spatial database system. Submitted for publication.

[3] A. Folkers, A. Soffer, and H. Samet. Processing pictorial queries with multiple instances using isomorphic subgraphs. In *Proc. ICPR'00*, vol. 4, pages 51–54, Barcelona, Spain, Sept. 2000.

[4] D. Heesch and S. Rüger. Combining features for content-based sketch retrieval – A comparative evaluation of retrieval performance. In *Proc. of the 24th BCS-IRSG European Colloquium on IR Research*, LNCS 2291, pages 41–52, Berlin, 2002. Springer.

[5] B. Jähne. *Digital Image Processing*. Springer, Berlin, 4 edition, 1997.

[6] A. K. Jain. *Fundamentals of Digital Image Processing*. Information and Systems Science Series. Prentice Hall, 1989.

[7] M. Jaisimha. Wavelet features for similarity based retrieval of logo images. In *Proc. of SPIE Document Recognition III*, vol. 2660, pages 89–100, San Jose, CA, Jan. 1996.

[8] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.

[9] A. Soffer and H. Samet. Pictorial query specification for browsing through spatially referenced images databases. *Jour. of Vis. Lang. and Comp.*, 9(6):567–596, Dec. 1998.

[10] A. Soffer and H. Samet. Two approaches for integrating symbolic images into a multimedia database system: a comparative study. *VLDB Journal*, 7(4):253–274, Dec. 1998.

[11] A. Soffer, H. Samet, and D. Zotkin. Pictorial query trees for query specification in image databases. In *Proc. ICPR'98*, vol. 1, pages 919–921, Brisbane, Australia, Aug. 1998.

[12] P. Suda, C. Bridoux, B. Kämmerer, and G. Maderlechner. Logo and word matching using a general approach to signal registration. In *Proc. ICDAR '97*, pages 61–65, Ulm, Germany, Aug. 1997.

Background Learning for Robust Face Recognition

R. K. Singh and A. N. Rajagopalan
Department of Electrical Engineering
Indian Institute of Technology Madras, India
{randhir, raju}@ee.iitm.ernet.in

Abstract

In this paper, we propose a robust face recognition technique based on the principle of eigenfaces. The traditional eigenface recognition (EFR) method works quite well when the input test patterns are cropped faces. However, when confronted with recognizing faces embedded in arbitrary backgrounds, the EFR method fails to discriminate effectively between faces and background patterns, giving rise to many false alarms. In order to improve robustness in the presence of background, we argue in favor of learning the distribution of background patterns. A background space is constructed from the background patterns and this space together with the face space is used for recognizing faces. The proposed method outperforms the traditional EFR technique and gives very good results even on complicated scenes.

Keywords: Face recognition, eigenfaces, face detection, background learning

1. Introduction

In the literature, several works have appeared on the face recognition problem [1, 2, 3, 4, 5]. One of the very successful and well-known face recognition methods is based on the Karhunen-Loeve (KL) expansion [3]. In 1986, Sirovich and Kirby [3] studied the problem of KL representation of faces. They showed that if the eigenvectors corresponding to a set of training face images are obtained, any image in that database can be optimally reconstructed using a weighted combination of these eigenvectors. The paper explored the representation of human faces in a lower dimensional subspace. In 1991, Turk and Pentland [5] used these eigenvectors (or eigenfaces as they are called) for face detection and identification.

Methods such as EFR work quite well provided the input test pattern is a face i.e, the face image has already been cropped and plucked out of a scene. The more general and difficult problem of recognizing faces in a cluttered background has also received some attention in [1, 5]. The authors in [1, 5] propose the use of distance from face space (DFFS) and distance in face space (DIFS) to detect and eliminate non-faces. We show with examples that DFFS and DIFS by themselves (in the absence of any information about the background) are not sufficient to discriminate against arbitrary background patterns. The traditional EFR technique either ends up missing faces or throws up many false alarms, depending on the threshold value. In this paper, we extend the EFR technique to solve the more general problem of robustly recognizing multiple faces in a given scene with background clutter. We explore the possibility of constructing a "background space" which will represent the background images corresponding to a given test image. If the background space is learnt well, it is our claim that patterns belonging to clutter will be closer to the background space than to the face space. This provides a basis for eliminating false alarms which would otherwise have crept in.

2. Effect of Background

The problem involving non-face test images is a difficult one and some attempts have been made to tackle it [1, 5]. In [5], the authors advocate the use of distance from face space to reject non-face images. If $\hat{T}_i$ is the projection of the mean subtracted image pattern T_i in the face space, then $\hat{T}_i$ can be expressed as

$$\hat{T}_i = \sum_{n=1}^{L'} w_n^i u_n$$

where w_n^i is the weight corresponding to eigenface u_n and L' is the number of eigenfaces used. The distance from face space (DFFS) is then defined as

$$\|\hat{T}_i - T_i\|^2 \qquad (1)$$

1051-4651/02 $17.00 © 2002 IEEE

It has been pointed out in [5] that a threshold θ_{DFFS} could be chosen such that it defines the maximum allowable distance from the face space. A test pattern is treated to be a face provided its DFFS value is less than θ_{DFFS}. In order to perform recognition, the difference error between the weight vector and the weight vector corresponding to every person in the training set is computed. This error is also called the distance in the face space (DIFS). The face class in the training set for which the DIFS is minimum is declared as the recognized face provided the difference error is less than an appropriately chosen threshold θ_{DIFS}.

However, it is difficult to conceive that by learning just the face class we can segregate any arbitrary background pattern against which the face patterns may appear. As we will show, it may not always be possible to come up with threshold values that will result in no false alarms and yet detect all faces. What would truly be desirable is to have a way of setting the threshold high, so that very few face images are rejected as unknown, while at the same time all incorrect classifications are detected. This is exactly what we attempt to do in this paper. We believe that some properties of the background scene local to a given image must be extracted and utilized for robust face recognition.

3. The Background Space

We argue in favor of learning the distribution of background images specific to a given scene. It is to be expected that background distribution will favor background images while the distribution of faces would favor the face patterns. In any given image, the number of background patterns usually far outnumbers the faces. To learn the distribution of the background, we need to generate *sufficient* number of observation samples from the given test image. We use simple thresholding to separate background patterns using the a priori statistical knowledge base of faces or the face space. Let $\underline{m}_1, \underline{m}_2, \ldots \underline{m}_q$ be the mean values of the weights corresponding to each face class in the training set. Here q is the number of face classes or people in the training set. In the face space, let the weight vector of the test subimage $\underline{x}$ be given by $\underline{W}$. Then, the pattern $\underline{x}$ is treated as a background image if the Euclidean distance of its weight vector from each of the class mean weights is greater than a predefined threshold θ_b i.e.,

$$\text{If } \|\underline{W} - \underline{m}_i\| > \theta_b \ \forall i, \ i = 1, \ldots, q \qquad (2)$$

then the image pattern is considered to be a non-face image. For high confidence, this threshold is chosen to be large enough. Sufficient number of background patterns can be obtained from the given test image in this manner. These patterns would represent a reasonable sampling of the background scene. The mean and covariance estimated from the samples obtained via (2) allow us to effectively extrapolate to other background patterns as well. A background image reconstructed with the eigenbackground images can be expected to have smaller error as compared to the case when it is reconstructed using eigenfaces.

We group the background patterns into K different clusters by the classical K-means algorithm where each cluster contains one pattern center. Each pattern center is treated to be representative of all the samples within its cluster. Thus, we can significantly reduce the number of background images that we have to deal with.

The pattern centers returned by the K-means algorithm are used as training images for learning the background space. Although the pattern centers belong to different clusters, they are not totally uncorrelated and further dimensionality reduction is possible. The procedure that we follow is similar to that used to create the face space. We first find the principal components (KL expansion) of the background pattern centers or the eigenvectors of the covariance matrix C_b of the set of background pattern centers. The space spanned by the eigenvectors corresponding to the largest K' eigenvalues of the covariance matrix C_b is called the background space. The significant eigenvectors of the matrix C_b, which we call 'eigenbackground images', form a basis for the background image patterns.

4. Robust Face Recognition

In this section, we propose a robust face recognition scheme that finds faces by searching a given test image for patches of image patterns of faces embedded in a cluttered background and finally classification. Training data samples of image patterns of faces are first used to create the face space. Given a test image, the background is then learnt 'on the fly' and the background space corresponding to that test image is derived. Finally, the system classifies a subimage as being either a known face or as a background pattern by using the knowledge of both the face space and the background space.

Once face space and the background space are learnt, the test image is examined again, but now for the presence of faces at all points in the image. Let the subimage pattern under consideration in the test image be denoted as $\underline{x}$. The vector $\underline{x}$ is projected onto the face space as well as the background space to yield estimates of $\underline{x}$ as $\hat{\underline{x}}_f$ and $\hat{\underline{x}}_b$, respectively. The test pattern $\underline{x}$ is classified as belonging to the 'face class' if

$$\|\underline{x} - \hat{\underline{x}}_f\|^2 < \|\underline{x} - \hat{\underline{x}}_b\|^2$$
$$\text{and} \quad \|\underline{x} - \hat{\underline{x}}_f\|^2 < \theta_{DFFS} \qquad (3)$$

where θ_{DFFS} is an appropriately chosen threshold. Recognition of $\underline{x}$ is then carried out as follows. The weight vector $\underline{W}$ corresponding to pattern $\underline{x}$ in the face space is compared

with the pre-stored mean weights of each of the face classes. The pattern $\underline{x}$ is recognized as belonging to the i^{th} person if

$$i = \min_j \|\underline{W} - \underline{m}_j\|^2, \quad j = 1, \ldots, q$$
$$\text{and} \quad \|\underline{W} - \underline{m}_i\|^2 < \theta_{DIFS} \tag{4}$$

where q is the number of face classes or people in the database and θ_{DIFS} is a suitably chosen threshold.

Since a background pattern will be better approximated by the eigenbackground images than by the eigenface images, it is to be expected that $\|\underline{x} - \hat{\underline{x}}_b\|^2$ would be less than $\|\underline{x} - \hat{\underline{x}}_f\|^2$ for a background pattern $\underline{x}$. On the other hand, if $\underline{x}$ is a face pattern, then it will be better represented by the face space than the background space. Thus learning the background space helps to reduce the false alarms considerably and imparts robustness to the EFR technique.

5. Experimental Results

In this section, we demonstrate the performance of the proposed scheme on two different datasets i) the standard Yale face database and ii) face database generated in our laboratory. The Yale database consists of 165 gray scale frontal images of 15 subjects. These are taken under different lighting conditions and facial expressions, and our intention is to test the proposed method under different conditions. For our experiments, we selected 15 individuals and 10 training images for each individual. The images were cropped to 33 × 33 pixel arrays. The face space was constructed from this training set offline. After some experimentation, the number of significant eigenvectors was found to be 40 for satisfactory performance. The database created in our laboratory consists of images of 8 subjects with 10 images per subject. The face images were cropped to 21 × 21 pixel arrays for training. The number of significant eigenfaces used to create the eigenface space for this database was chosen to be 20.

The system was first tested by artificially embedding images of some of the subjects from the Yale database at random locations in different test images of size 128 × 128 pixels against a background scene that included trees, roads and building structures. The test image was scanned for the presence of faces at all points in the image. If a face pattern is found at any location in the test image, a white box is drawn at that location. For the second set of experiments, test images were captured in our laboratory and the subjects appear naturally in these real images. The background consisted of computers, furnitures etc. These images serve to represent real face recognition situations. A black box is drawn at the location where the system finds a face.

For the proposed method, the eigenbackground space was learnt 'on the fly' for each test image using the methodology discussed in Section 3. Thresholds θ_{DFFS} and θ_{DIFS} were chosen to be the maximum of all the DFFS and DIFS values, respectively, among the correctly recognized faces in the training set. The number of background pattern centers was chosen to be 600 while the number of eigenbackground images were chosen to be 150. The number of eigenbackground images was arrived at based on the accuracy of reconstruction of the background patterns.

Results corresponding to Yale database for the two methods are shown in Fig. 1. The figures are quite self-explanatory. The traditional EFR incurs many alarms when it attempts to recognize all the faces in the image. On the other hand, the proposed method detects all the faces without false alarms. Results obtained on real images captured in the laboratory are given in Figs. 2 - 3. Our method utilizes the background information quite effectively in order to discard non-face patterns, whereas the traditonal EFR throws up false alarms.

6. Conclusion

In the literature, the eigenface technique has been demonstrated to be very useful for face recognition. However, when the scheme is directly extended to recognize faces embedded in background clutter, its performance degrades as it cannot satisfactorily discriminate against non-face patterns. In this paper, we have presented a robust scheme for recognizing multiple faces in still images of natural scenes against a cluttered background. We argue in favor of constructing a background space from the background images of a given scene. With moderate computational complexity, the scheme outperforms the traditional EFR technique and gives accurate recognition results on real images with almost no false alarms even on fairly complicated scenes.

References

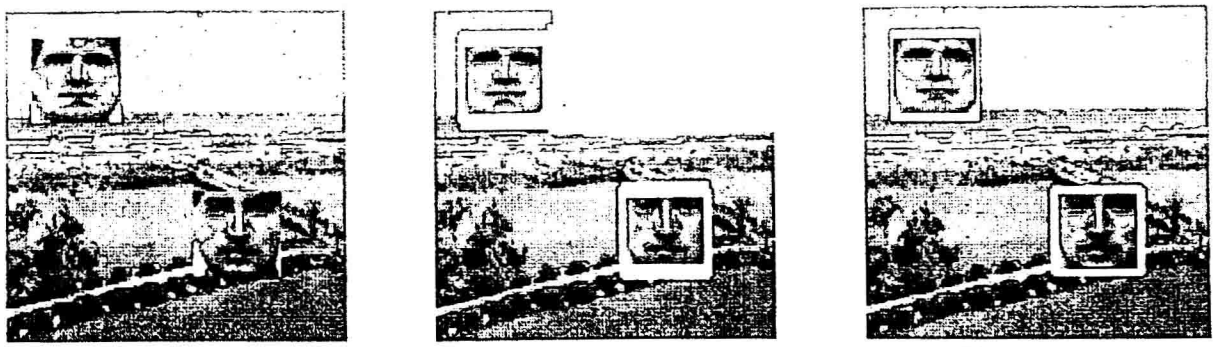
[1] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Analysis and Machine Intell.*, 19:696–710, 1997.

[2] A. N. Rajagopalan, K. S. Kumar, J. Karlekar, R. Manivasakan, M. M. Patil, U. B. Desai, P. G. Poonacha, and S. Chaudhuri. Locating human faces in a cluttered scene. *Graphical Models in Image Processing*, 62:323–342, 2000.

[3] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J. Opt. Soc. Am. A*, 4:519–524, 1987.

[4] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 20:39–51, 1998.

[5] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neurosciences*, 3:71–86, 1991.

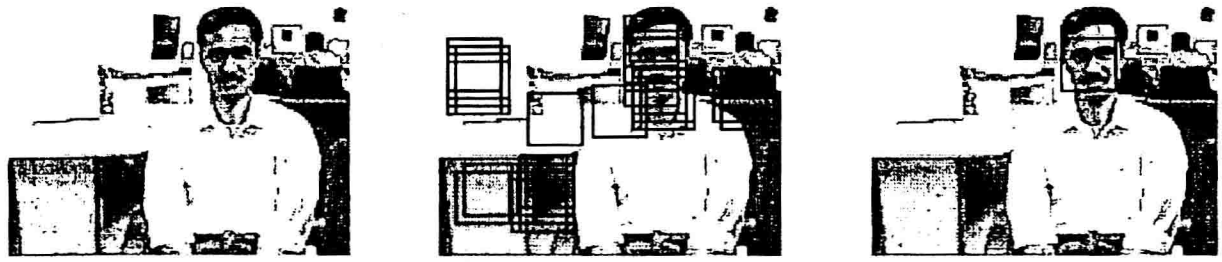


(a) (b) (c)

Figure 1. (a) A test image with faces embedded in it. (b) Recognition results corresponding to traditional EFR using both DFFS and DIFS. Even though the faces are correctly recognized, there are a lot of false alarms in the upper right corner. (c) Output results for the proposed EFR method. There are no false alarms and both the faces are correctly recognized.

(a) (b) (c)

Figure 2. (a) A real test image where a person appears naturally against a cluttered scene. (b) Face recognition results for the traditional EFR technique using both DFFS and DIFS. (c) Recognition results with proposed method.



(a) (b) (c)

Figure 3. (a) Test image consisting of desks and computers as background clutter. Recognition results for (b) traditional EFR, and (c) proposed method. Note that traditional EFR throws up many false alarms.

Two-Hand Gesture Recognition using Coupled Switching Linear Model

Mun Ho Jeong,
Dept. Computer-Controlled
Mechanical Systems,
Osaka University.

Yoshinori Kuno,
Dept. Information and
Computer Sciences,
Saitama University.

Nobutaka Shimada,
Yoshiaki Shirai,
Dept.Computer-Controlled
Mech. Sys., Osaka Univ.

Abstract

We present a method coupling multiple switching linear models. The coupled switching linear model is an interactive process of two switching linear models. Coupling is given through causal influence between their hidden discrete states. The parameters of this model are learned via EM algorithm. Tracking is performed through the coupled-forward algorithm based on Kalman filtering and a collapsing method. A model with maximum likelihood is selected out of a few learned models during tracking. We demonstrate the application of the proposed model to tracking and recognizing two-hand gestures.

1. Introduction

Gesture recognition plays an important role in a host of man-machine interaction applications. Although some gestures are expressed by one hand, many of them are done by two hands. To model these two-hand gestures, we have to consider interactions between the two hands. We assume that a two-hand gesture is an interacting process of the two hands whose shapes and motions are described by the switching linear dynamics [2,3,6], and propose a coupled switching linear dynamic model to capture interactions between the two hands.

CHMM (coupled hidden Markov model) [4] has been proposed to deal with interacting processes. However, since CHMM inherits from HMM, it has a limitation in treating time series having dependencies like shape-changing hand gestures.

Reynard [5] has introduced a coupling concept to track complex motions, however, that means just a coupling of two kinds of continuous state variables in a single process, and is essentially different from interaction considered here.

We demonstrate an application of the coupled switching linear model to tracking and recognizing two hands whose shapes change during their motion. The presented coupling scheme enables tracking both hands even when one of them is not observed well in images by occlusions or complex backgrounds. And it also gives probabilistic explanation to recognition of gestures by combination of two hands.

2. Coupled switching linear model

2.1 Model specification

To represent a variety of shapes of a hand, it may be an efficient way that outlines of the hand are parameterized by active contour model using B-spline, which was well established in [1]. A curve is parameterized into a control vector composed of B-spline control points. A control vector is transformed to a low-dimensional shape vector on a specific shape space formed with some key control vectors. Then the shape vector, s_t, is considered as a state vector in switching linear dynamics:

$$x_t = \begin{pmatrix} s_t \\ \dot{s}_t \end{pmatrix}$$

Switching linear model can be seen as a hybrid model of the linear state-space model and HMM. It is described using the following set of state-space equations:

$$\begin{aligned} & x_t = F_{m_t} x_{t-1} + D_{m_t} + u_t,\ u_t \sim N(0, Q_{m_t}) \\ & \Phi_{m_{t+1}, m_t} = p(m_{t+1} \mid m_t) \\ & \pi_{m_1} = p(m_1) \end{aligned} \tag{1}$$

In the above equations, x_t is a hidden continuous state vector. u_t is independently distributed on the Gaussian distribution with zero-mean and covariance Q_{m_t}, π_{m_1}, F_{m_t} and D_{m_t}, which are typical parameters of linear dynamic model, denote the prior probability of a discrete state, the continuous state transition matrix, and the offset, respectively. The parameters with the subscript m_t are dependent on the discrete state variable m_t indexing a linear dynamic model. And the switching process between discrete states obeys the first Markov process and is defined with the discrete state transition matrix Φ.

Coupled switching linear model is an interactive process of two switching linear models. Coupling is given through causal influence between their hidden discrete states. The complex state space representation is equivalently depicted by dependency graph in Figure 1.

To accommodate another interacting process, it seems

1051-4651/02 $17.00 © 2002 IEEE

good enough to consider a single lumped system with dimension-increased state variables. However, there exist a few problems. Due to increased number of discrete states, the computational cost is prohibitive, and sufficient data can rarely be obtained for estimation of parameters, usually resulting in under-sampling and numerical underflow errors [4]. Consequently, the suggested coupling scheme, as shown in Figure 1, offers conceptual advantages of parsimony and clarity with computational benefits in efficiency and accuracy. This is revealed in the following sections.

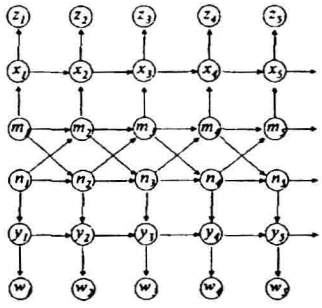


Figure 1. Coupled switching linear model. m_t and n_t denote discrete state variables. x_t and y_t denote continuous state variables. z_t and w_t denote observation vectors.

In the coupled switching linear model, since transition between discrete states is Markov process, it follows that

$$p(m_t, n_t \mid m_1, \cdots, m_{t-1}, n_1, \cdots, n_{t-1}) = p(m_t, n_t \mid m_{t-1}, n_{t-1})$$

Assuming

$$p(m_t, n_t \mid m_{t-1}, n_{t-1}) \propto p(m_t \mid m_{t-1}) \cdot p(m_t \mid n_{t-1}) p(n_t \mid n_{t-1}) p(n_t \mid m_{t-1}) \quad (2)$$

referred to in [4], transition probability of joint discrete states can be parameterized as

$$p(m_t, n_t \mid m_{t-1}, n_{t-1}) = k_c \Phi_{m_{t-1} m_t} \Gamma_{n_{t-1} m_t} \hat{\Phi}_{n_{t-1} n_t} \hat{\Gamma}_{m_{t-1} n_t} \quad (3)$$

where k_c is a normalizing constant, Γ is the state transition matrix representing causal influences between two switching linear system, and superscript ^ denotes the lower switching linear system in Figure 1.

2.2 Coupled-forward algorithm

Following [3], given the known parameters of switching linear dynamics, the predicted joint-continuous state variable and the corresponding covariance are defined dependently on $m_{t-1} = i$ and $m_t = j$:

$$\begin{aligned} x_{t|t-1}^{(i,j)} &= F_j x_{t-1|t-1}^{(i)} + D_j \\ P_{t|t-1}^{(i,j)} &= F_j P_{t-1|t-1}^{(i)} F_j' + Q_j \end{aligned} \quad (4)$$

where $x_{t-1|t-1}^{(i)}$ and $P_{t-1|t-1}^{(i)}$ are the filtered continuous states and its covariance at time $t-1$ based on information up to time $t-1$. Now the filtered joint-continuous state $x_{t|t}^{(i,j)}$ and its covariance $P_{t|t}^{(i,j)}$ are estimated by the conventional Kalman updating algorithm. In particular, we follow Kalman filtering application of [1] to active contour model.

From the above fact, as noted by [3], switching linear dynamic model requires computing a Gaussian mixture with M^t components at time t for M switching states. If coupled with a N-switching linear system as shown in Figure 1, typically $M^t + N^t$ computations are required, which is clearly intractable for moderate sequence length. It is necessary to introduce some approximations to solve the intractable computation problem.

We collapse $M^2 + N^2$ jointed continuous state variables into $M + N$ state variables at each time, and can avoid prohibitive increase of computational cost. The following collapsing method is given: Expediently only in terms of the upper system in Figure 1.

$$x_{t|t}^{(j)} = \frac{\sum_{i=1}^{M} \left(\sum_{ii=1, jj=1}^{N} p\begin{pmatrix} m_{t-1} = i, n_{t-1} = ii, \\ m_t = j, n_t = jj \mid O_t \end{pmatrix} \cdot x_{t|t}^{(i,j)} \right)}{p(m_t = j \mid O_t)} \quad (5)$$

$$P_{t|t}^{(j)} = \frac{\sum_{i=1}^{M} \left(\sum_{ii=1, jj=1}^{N} p(m_{t-1} = i, n_{t-1} = ii, m_t = j, n_t = jj \mid O_t) \cdot \left(P_{t|t}^{(i,j)} + (x_{t|t}^{(j)} - x_{t|t}^{(i,j)})(x_{t|t}^{(j)} - x_{t|t}^{(i,j)})' \right) \right)}{p(m_t = j \mid O_t)}$$

where O_t is a sequence $(o_1, o_2, \cdots, o_t)$ and o_t is an observation vector (z_t, w_t).

The filtered coupled-joint distribution of discrete states is defined by

$$\begin{aligned} & p(m_{t-1}, n_{t-1}, m_t, n_t \mid O_t) \\ &= k_t p(z_t \mid x_{t|t-1}^{(m_{t-1}, m_t)}) p(w_t \mid y_{t|t-1}^{(n_{t-1}, n_t)}) p(m_{t-1}, n_{t-1}, m_t, n_t \mid O_{t-1}) \end{aligned} \quad (6)$$

where k_t is a normalizing constant. From (2) and (3) the prediction step given sequence up to time t gives

$$\begin{aligned} p(m_t, n_t, m_{t+1}, n_{t+1} \mid O_t) &= k_p \Phi_{m_t m_{t+1}} \Gamma_{n_t m_{t+1}} \\ &\cdot \hat{\Phi}_{n_t n_{t+1}} \hat{\Gamma}_{m_t n_{t+1}} \sum_{m_{t-1} n_{t-1}} p(m_{t-1}, n_{t-1}, m_t, n_t \mid O_t) \end{aligned} \quad (7)$$

$$p(m_{t+1}, n_{t+1} \mid O_t) = \sum_{m_t, n_t} p(m_t, n_t, m_{t+1}, n_{t+1} \mid O_t) \quad (8)$$

$$p(m_t \mid O_t) = \sum_{m_{t-1}, n_t, n_{t-1}} p(m_t, n_t, m_{t-1}, n_{t-1} \mid O_t)$$

$$x_{t|t} = \sum_{m_t=1}^{M} p(m_t \mid O_t) x_{t|t}^{(m_t)}$$

where k_p is a normalizing constant. Now (6) and (7) are iterated during filtering process.

2.3 Coupled -backward algorithm

While the coupled-forward algorithm is a filtering process given sequence up to current time, the coupled-backward algorithm is a smoothing process given sequence of full length. Like the conventional Kalman smoothing method, joint-continuous state variable and its covariance are smoothed [3]. And the collapsing is similarly performed using the following probability of the smoothed coupled-joint discrete states:

$$p(m_t,n_t,m_{t+1},n_{t+1}\mid O_T)=p(m_t,n_t,m_{t+1},n_{t+1}\mid O_t)\cdot\frac{p(m_{t+1},n_{t+1}\mid O_T)}{p(m_{t+1},n_{t+1}\mid O_t)} \quad (9)$$

From (9) the followings are obtained as

$$p(m_t,n_t\mid O_T)=\sum_{m_{t+1},n_{t+1}} p(m_t,n_t,m_{t+1},n_{t+1}\mid O_T) \quad (10)$$

$$p(m_t\mid O_T)=\sum_{n_t} p(m_t,n_t\mid O_T) \quad (11)$$

3. EM learning

EM algorithm is a general iterative technique for finding maximum likelihood parameter estimates in problems where some variables are unobserved [7]. It is natural to use EM algorithm for our problem, in which unobserved variables are continuous state variables x_t, y_t and discrete state variables m_t, n_t.

Assuming that the probability density for an observation sequence is parameterized using λ, which consists of $\{F,Q,\pi,\Phi,\Gamma\}$ and $\{\hat{F},\hat{Q},\hat{\pi},\hat{\Phi},\hat{\Gamma}\}$, its auxiliary log-likelihood is given by

$$L=\sum_{M_T,N_T}\int_{X_T,Y_T}\bar{p}\log p(M_T,N_T,X_T,Y_T,O_T\mid\lambda)dX_TdY_T$$
$$=E_{\bar{p}}[\log p(M_T,N_T,X_T,Y_T,O_T\mid\lambda)] \quad (12)$$

where (M_T,N_T) and (X_T,Y_T), are sequences(of length T) of discrete states and continuous states, respectively, $\bar{\lambda}$ is the parameter set estimated previously, and $\bar{p}=p(M_T,N_T,X_T,Y_T\mid O_T,\bar{\lambda})$. EM algorithm starts with some initial guess and proceeds by applying the following two steps repeatedly until the likelihood converges:

E-step On the condition given the observation sequence of full length O_T and the previous parameter set $\bar{\lambda}$, we estimate the hidden continuous states and discrete states through the backward process following the forward process described in sections 2.2 and 2.3.

M-step If L is expressed by λ and the estimations from E-step, then we estimate λ maximizing L.

4. Recognition

Recognition of hand gestures can be considered as the problem to determine which model tracks a hand gesture well. Therefore, a given sequence of hand gestures can be recognized by means of the likelihood values of candidate models.

In order to track and recognize hand gestures simultaneously, we have to compute the likelihood of each model while tracking is being performed with the coupled-forward algorithm.

If the coupled switching linear model is represented by the parameter set λ, log-likelihood $\tilde{L}_\tau$ of λ at time τ is obtained by

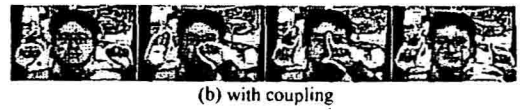
$$\tilde{L}_\tau=\sum_{t=1}^{\tau}\log(\frac{1}{k_t}) \quad (13)$$

where k_t has been computed in (6).

5. Experimental Result

An observation, o_t, is shown as edges detected by line searching along the normal direction at sample points on a hand contour [1]. Generally the use of a skin color model gives robustness to the edge detection. However, the edge detection using color models fails well when a hand moves in front of a face, which is frequent in hand gestures considered. Figure 2-(a) shows two saparate trackers of hands using conventional switching linear model [6]. The left hand tracker was not able to catch a finger's moves due to failure in the detection of edges. However, in Figure 2-(b) where the presented coupling method was applied, since the right hand tracker forces the left one to operate its own switching dynamics, the left tracker could track successfully regardless of failures in edge detection. We can confirm it in Figure 3 which shows transition between discrete states in the left hand tracker. The left hand model has been trained to have three states: State 1 corresponds to moving one's fist, state 2 describes shape changes from stone to scissors, and state 3 corresponds to moving the scissors to the origin. If coupling is applied, switching discrete states is well performed as shown in Figure 3-(b).

(a) without coupling

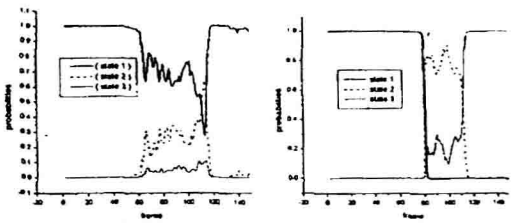


(b) with coupling

Figure 2. Tracking two hands

For the purpose of recognition during tracking, tracking is performed through the coupled-forward

algorithm with respect to all models. At the same time, likelihood values for all models are computed by (13). Accordingly, an observed sequence is recognized as the model with the maximum likelihood.



(a) Without coupling (b) With coupling

Figure 3. Transition between discrete states.

We have prepared two gesture models in order to test whether the proposed scheme is effective to express combinational property of two hand gestures. Two gesture models were designed to have three discrete states for each hand. In gesture model A two hands change their moving patterns at the same time. Gesture model B has the similar moving patterns to gesture model A except different changes of moving patterns on the way as shown in Figure 4.

The different changes of moving patterns can be explained by differences in causal influences between both hands. The causal influences were parameterized as $\Gamma, \hat{\Gamma}$ through the EM learning. When the two models are applied to the image sequence of Figure 4-(a), the two models can track the given sequences since they have similar motions. However, Figure 5 shows that the difference between likelihood values of the models increased severely from about 70^{th} frame, at which the different change of moving patterns happpened. This fact confirms that with the causal influences denoted by $\Gamma, \hat{\Gamma}$, the proposed coupled switching linear model well explains the interaction between two hands.

7. Conclusion

We have proposed a coupled switching linear model, which is an interacting process between two switching linear models and presented its EM learning method using a collapsing method. We have applied the proposed scheme to recognizing two-hand gestures. The presented method showed the effectiveness in tracking shape-changing hands under failures in feature detection. And it also showed satisfactory results that two-hand gestures are recognized and tracked simultaneously.

References

[1]Blake,A. and Isard,M., *Active contour*,Springer-Verlag, 1998.

[2]Ghahramani,Z. and Hinton,G. E., "Variational Learning for Switching State-Space Models", *CRG-TR-96-3 of Toronto Univ.*, 1996.

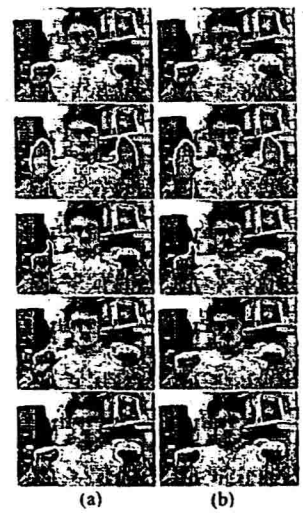
[3]Kim,C-J., "Dynamic Linear Models with Markov-Switching", *Journal of Econometrics*, Vol. 60, pp. 1-22, 1994.

[4]Brand,M., Oliver,N. and Pentland,A.,1997, "Couple Hidden Markov Models for Complex Action Recognition", *Proc.,IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'97.

[5] Reynard David, Andrew Wildenberg, Andrew Blake and John Marchant, "Learning Dynamics of Complex Motions from Image Sequences", *Proc. European Conf. on Computer Vision*, vol. 1, pp. 357-368, Cambridge UK, 1996.

[6] Jeong,M.H., Kuno,Y., Shimada,N., and Shirai,Y., "Recognition of Shape-Changing Hand Gestures Based on Switching Linear Model", ICIAP2001, 2001.

[7]Dempster, A., Laird, M. and Rubin, D., "Maximum Likelihood from Incomplete Data via the EM Algorithm", *J. of the Royal Statistical Society*, B(39), pp.1-38, 1977.



(a) (b)

Figure 4. Recognition of two-hand gestures during tracking.

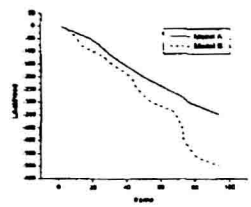


Figure 5. Likelihood vs. frame. Given the sequence in Figure 4-(a), likelihood values are plotted with respect to model A and B.

Improving Retrieval Performance by Long-term Relevance Information

Peng-Yeng Yin[1], Bir Bhanu[2], Kuang-Cheng Chang[1] and Anlei Dong[2]
[1]Department of Computer Science, Ming Chuan University, Taoyuan 333, Taiwan
[2]Center for Research in Intelligent Systems, University of California, Riverside, California 92521, USA
{pyyin, kcchang}@mcu.edu.tw, {bhanu, adong}@vislab.ee.ucr.edu

Abstract

Relevance feedback (RF) is an iterative process which improves the retrieval performance by utilizing the user's feedback on retrieved results. Traditional RF techniques use solely the short-term experience and are short of knowledge of cross-session agreement. In this paper, we propose a novel RF framework which facilitates the combination of short-term and long-term experiences by integrating the traditional methods and a new technique called the virtual feature. The feedback history of all the users is digested by the system and is represented as a virtual feature of the images. As such, the dissimilarity measure can be adapted dynamically depending on the estimate of the relevance probability derived from the virtual features. The results manifest that the proposed framework outperforms the one that adopts a single traditional RF technique.

1. Introduction

Since the users in general do not know the make-up of the image database and the techniques used for indexing, the query formulation process should be treated as a series of tentative trials until the target images are found. Relevance feedback (RF) is an automatic process which fulfills the requirements of the query formulation.

Let a user initialize a query session by submitting an image represented by $Q = (q_1, q_2, ..., q_t)$ where t is the number of selected features and q_i is the calculated value of the ith feature. The retrieval system compares the query image with each of the database images, say $D = (d_1, d_2, ..., d_t)$, by deriving the dissimilarity measure $Dist(Q, D)$. The top k database images that have the smallest dissimilarity score are then returned to the user. If the user is not satisfied with the retrieved result, he or she can activate an RF process by identifying which retrieved images are relevant to the query and which are not. The system will adapt its internal parameters to involve as many desirable images as possible in the next retrieved result. The process is repeated until the user is satisfied with the retrievals. The general system flow chart of the RF process is depicted in Fig. 1. In the following, the main RF techniques for image retrieval are presented.

The query vector modification (QVM) approach [1] repeatedly reformulates the query vector as the mean difference vector between relevant images and nonrelevant ones, in the attempt to redirect the query vector toward a more desired area. The feature relevance estimation (FRE) approach [2] assumes, for a given query, some specific features may be more important than others when computing the similarities between images and the query. The most natural way of estimating the individual feature relevance is to verify the retrieval ability using each feature alone. Finally the feature relevance is used as a weight incorporated into the dissimilarity metric. The Bayesian inference-based (BI) approach [3] estimates the posterior probability that a database image is relevant to the query given the prior feedback history. The probability distribution over all database images is updated after each feedback iteration, the system is therefore able to improve the future retrieval performance.

These methods suffer their respective shortcomings. First, the QVM put equal emphasis on every relevant image by averaging their feature vectors, however, not every relevant image has the same magnitude of relevance. Second, the success of both QVM and FRE is based on the assumption that the distributions of the feature vectors of the relevant images form an intrinsic cluster. Whereas, no matter how sophisticated features are selected, they are insufficient to fully represent the image semantics, and the relevant images will usually do not form a single cluster. Third, without storing the

1051-4651/02 $17.00 © 2002 IEEE

relevance information directly, some information is lost such as the relevance significance of each individual image. The BI approach is theoretically the most flexible one since it does not rely on the nearest neighbor criterion. However, the BI approach needs more feedback iterations to accurately approximate the posterior probability distribution. So it is less efficient than the other RF techniques.

Moreover, all the three kinds of RF approaches improve the retrieval performance based on the feedback history within one query session. Hence, the previous approaches maintain a form of short-term memory that captures the user's intention for only this specific query. There is no consideration being taken for the cross-session feedback history, which is a form of long-term memory that captures the common agreement among various query sessions. The long-term memory is useful in leading the feedback process to converge at an earlier iteration.

2. The Proposed Approach

To digest the relevance information accumulated from within- or cross-session query experiences, we add a virtual feature (VF) to the feature vector of each of the database images. The VF is determined by the set of relevant images and is used to assist the original pictorial features to evaluate the similarity degree between images in accordance with the human subject. The details of the proposed approach are presented as follows.

2.1 Virtual Feature Computation

Given a query $Q=(q_1,q_2,...,q_t)$, the retrieval system firstly searches the top k nearest images using the dissimilarity metric of the adopted short-term RF technique. If the user is not satisfied with the result, he would activate an RF process by identifying relevant and nonrelevant images. We denote by R the set of identified relevant images. Initially, the VF of each database image is empty. Each relevant image $D=(d_1,d_2,...,d_t)$ in R will derive its VF by requesting a number from a system counter. The system counter starts counting from 1 and is increased by 1 after every time it is requested. Therefore, all the images in R will be assigned the same value as their VFs to mark that these images deliver the same concept possessed by the query.

As the feedback process repeated, one case may arise that some of the images in R have been already assigned the VFs. The relevant images that have not determined their VFs yet will be firstly given a number from the system counter and this number is then concatenated by the VFs of the other relevant images and converted into a canonical form. We define the canonical form $\Im$ with the set of positive integers Z^+ and a concatenation operator $\otimes$ as follows. The concatenation operator $\otimes$ is defined by

$$c_i^{e_i}\otimes c_j^{e_j}=\begin{cases} c_i^{e_i}\otimes c_j^{e_j} & \text{if } c_i<c_j,\\ c_j^{e_j}\otimes c_i^{e_i} & \text{if } c_i>c_j,\\ c_i^{e_i+e_j} & \text{if } c_i=c_j,\end{cases}$$

where c_i, c_j, e_i and e_j are in Z^+. An expression f is in canonical form $\Im$ if $f=c_1^{e_1}\otimes c_2^{e_2}\otimes\cdots\otimes c_m^{e_m}$, where $c_i<c_j$ if $i<j$. Apparently, canonical form $\Im$ holds a closure property. Formally, $f_1, f_2\in\Im\rightarrow f_1\otimes f_2\in\Im$. The closure property guarantees that the VF yielded by concatenating several VFs is still in the canonical form $\Im$.

In this way, each value in the VFs represents a relevance concept impressed by a certain user, and the system can digest multiple concepts of image relevance in the VFs. To estimate the relevance between the query and database images, the VF of the query is computed as the concatenation of the VFs of all images in R which are specified in the previous feedback iteration, i.e., $VF(Q)=VF(D_1)\otimes VF(D_2)\otimes\cdots\otimes VF(D_{|R|})$, $D_i\in R$, where $VF(\cdot)$ denotes the VF of the corresponding image. The VFs of the query and the database images are used to define the dynamic dissimilarity measure which will be discussed in the next subsection.

2.2 Probabilistic dissimilarity measure

Let the VF of an image D be $c_1^{e_1}\otimes c_2^{e_2}\otimes\cdots\otimes c_m^{e_m}$, we firstly define the concept set of image D as $C(D)=\{c_1,c_2,...,c_m\}$, each concept c_i is associated with a support value e_i. The larger the cardinality of the concept set, the more general the overall concept delivered by the image. Also, the larger the support value, the more important to the image the corresponding concept. We define the probability that D is semantically recognized as concept c_i, or the confidence that D is delivering concept c_i, given $VF(D)$ as

$$p(D\quad c_i|VF(D))=\frac{e_i}{\sum_{j=1}^{m}e_j}.$$

Assume the two events that the concepts delivered by the query and by the database image are independent given their VFs. The probability, denoted by $p_{Q\equiv D|VF_Q}$, that the query Q and the database image D are

delivering the same concept given their VFs is calculated by

$$p_{Q\approx D|VFs} = \sum_{c_i \in C(Q)\cap C(D)} p(Q \quad c_i \text{ and } D \quad c_i | VF(Q), VF(D))$$

$$= \sum_{c_i \in C(Q)\cap C(D)} p(Q \quad c_i | VF(Q)) \, p(D \quad c_i | VF(D)).$$

Based on the probability estimate, we define a dynamic dissimilarity measure as $Dist_{VF}(Q,D) = p_{Q\approx D|VFs}(Dist(Q,D)-\Delta) + (1-p_{Q\approx D|VFs})(Dist(Q,D)+\Delta)$, if both $VF(Q)$ and $VF(D)$ are known, and $Dist_{VF}(Q,D) = Dist(Q,D)$ otherwise, where Δ is the quantity of the maximal distance adjustment, and $Dist(Q,D)$ is the distance metric defined by the short-term RF technique incorporated into our approach. The first equality can be rewritten as $Dist_{VF}(Q,D) = (1-2p_{Q\approx D|VFs})\Delta + Dist(Q,D)$. It is observed that $Dist_{VF}(Q,D) < Dist(Q,D)$ if $p_{Q\approx D|VFs} > 0.5$, $Dist_{VF}(Q,D) > Dist(Q,D)$ if $p_{Q\approx D|VFs} < 0.5$, and $Dist_{VF}(Q,D) = Dist(Q,D)$ if $p_{Q\approx D|VFs} = 0.5$. Therefore, the proposed method dynamically adjusts the distance between the query and the database images based on the estimate of $p_{Q\approx D|VFs}$ which is derived from the long-term feedback history.

Compared with the existing RF techniques, the proposed method has the following features.

- We assume neither the shape of the nearest neighborhood of the query nor the presence of one cluster containing all relevant images.
- The relevance information of the original users' intention is stored directly in the VFs. This mechanism enables us to define a flexible dissimilarity measure.
- The proposed method combines the short-term and long-term RF techniques to establish an effective retrieval system.

3. Experimental Results

We have implemented the QVM approach [1] and the proposed VF technique. The UCR database is chosen for the experiments. The database is obtained from the UCR Visualization and Intelligent Systems Lab (VISLab) [4]. There are 10038 images covering a variety of outdoor scenes such as castles, cars, humans, animals, etc. Some sample images are shown in Fig. 2. Since the number of images in the database is tremendous, it is laborious to classify these images manually. As such, we employ the c-means clustering algorithm [5] to automatically classify these images into 70 classes for performance evaluation purpose. Each image is represented by a 16-dimensional feature vector using the Gabor filters [6].

Also, in all the experiments, the performance is measured using the precision rate defined as

$$\text{Precision Rate} = \frac{\text{Relevance Retrievals}}{\text{Total Retrievals}} \times 100\%.$$

To simulate the practical situation of online users, the sequence of query images is generated randomly until each database image has been chosen at least once. Each query session is allowed to refine its retrievals by executing the RF process for two iterations. The average precision rates obtained at three different stages, namely the one without any relevance feedback (PR0), the one after the first feedback iteration (PR1), and the one after the second feedback iteration (PR2), are computed, respectively.

To understand the influence on the growing of precision rates by using the proposed VF technique, the accumulated precision rates that are averaged over the number of processed queries are plotted in Fig. 3. There is a fluctuating period in the beginning of the plotted curves depending on which images are firstly selected as query images. After this period, the accumulated precision rates climb up rapidly due to the contribution of the use of the active nearest neighborhood learned by the VFs. Looking at the curve of PR0, it reveals that the precision rate obtained even before performing the feedback iterations can be as high as 95% because the relevance information of the previously processed queries provides a valuable clue. Also, the improving ratio from PR0 to PR1 is higher than that from PR1 to PR2. This is a desired property since the users can not stand too many feedback iterations and they expect a greatly improved result after the first feedback. On the other hand, if we use solely the QVM method, there is no gain on the retrieval precision along the number of processed queries. As a result, the accumulated precision rates hold themselves to a relatively fixed value as shown in Fig. 4.

Next, we analyze the scalability of the proposed approach on the storage requirement of the VFs. First, we construct nine subdatabases from the UCR database. Each subdatabase consists of the images that are a certain amount of percentages of the original database volume (from 10% to 90%) and includes at least one image from every labeled class. Fig. 5 shows the storage requirement for the average length of the VFs. It is observed that the memory needs of the VFs grow less than three times when the test subdatabase size varies from 10% to 100% of the original database volume. Thus the proposed method is scalable against the variations of database size.

4. Conclusions

In this paper, we have presented a new RF approach for content-based image retrieval. The traditional RF techniques use only within-session query experience to improve the retrieval precision. We devise a new technique called the virtual feature which digests the cross-session query experiences to give the retrieval results that are more satisfactory. Experimental results show that the proposed retrieval system which uses a combination of short-term and long-term relevance information performs better than that adopting the short-term RF technique only.

5. References

[1] G. Salton and C. Buckley, Improving retrieval performance by relevance feedback, Journal of Am. Soc. Information Sci. 41 (1990) 288-297.

[2] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Trans. Circuits and Systems for Video Technology 8 (1998) 644-655.

[3] I. Cox, M. Miller, T. Minka, T. Papathomas and P. Yianilos, The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments, IEEE Trans. Image Processing 9 (2000) 20-37.

[4] Vision and Intelligent System Lab (VISLab), University of California, Riverside. Http://www.vislab.ucr.edu.

[5] L. Bobrowski and J. C. Bezdek, c-Means clustering with the l_1 and l_∞ norms, IEEE Trans. Syst. Man Cybernet. 21 (1991) 545-554.

[6] J. Peng, B. Bhanu, and S. Qing, Probabilistic feature relevance learning for content-based image retrieval, Computer Vision and Image Understanding 75 (1999) 150-164.

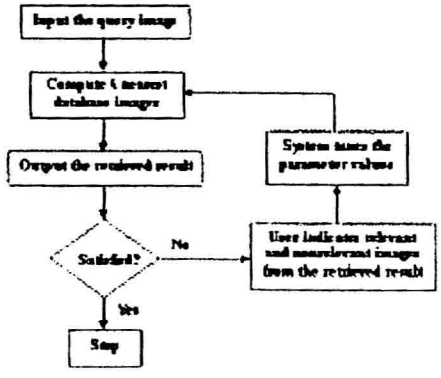


Fig. 1 System flow chart of the RF process.

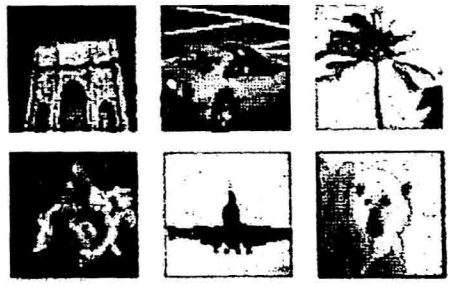


Fig. 2 Sample images from UCR database.

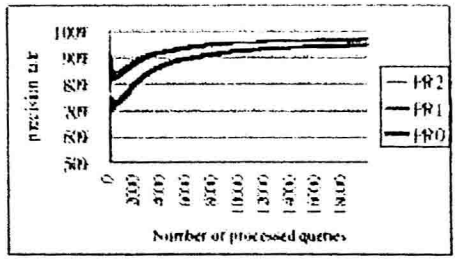


Fig. 3 Performance of the proposed method.

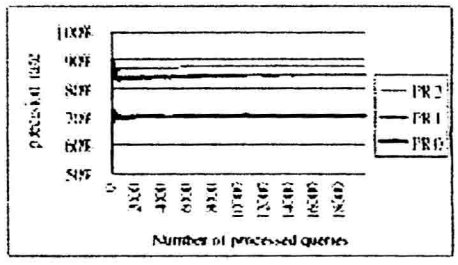


Fig. 4 Performance of the QVM approach.

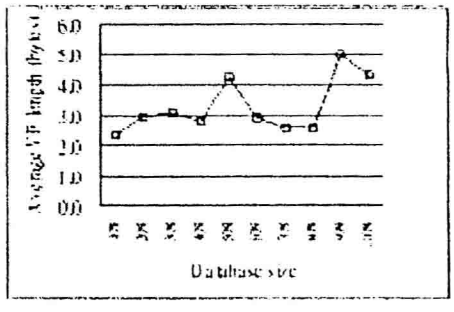


Fig. 5 The average VF length v.s. database size.