Pergamon

Pattern Recognition, Vol. 29, No. 8, pp. 1233–1244, 1996
Elsevier Science Ltd

0031-3203/96 $15.00 + 00

0031-3203(95)00160-3

IMAGE RETRIEVAL USING COLOR AND SHAPE

ANIL K. JAIN and ADITYA VAILAYA
Department of Computer Science, Michigan State University, East Lansing, MI 48824-1027, U.S.A.

(*Received* 8 *August* 1995; *in revised form* 19 *October* 1995; *received for publication* 24 *November* 1995)

Abstract—This paper deals with efficient retrieval of images from large databases based on the color and shape content in images. With the increasing popularity of the use of large-volume image databases in various applications, it becomes imperative to build an automatic and efficient retrieval system to browse through the entire database. Techniques using textual attributes for annotations are limited in their applications. Our approach relies on image features that exploit visual cues such as color and shape. Unlike previous approaches which concentrate on extracting a single concise feature, our technique combines features that represent both the color and shape in images. Experimental results on a database of 400 trademark images show that an integrated color- and shape-based feature representation results in 99% of the images being retrieved within the top two positions. Additional results demonstrate that a combination of clustering and a branch and bound-based matching scheme aids in improving the speed of the retrievals. Copyright © 1996 Pattern Recognition Society. Published by Elsevier Science Ltd.

Image database Color Shape Retrieval Logos Trademarks

1. INTRODUCTION

Digital images are a convenient media for describing and storing spatial, temporal, spectral and physical components of information contained in a variety of domains (e.g. satellite images, biomedical images). Due to the low cost of scanners and storage devices, digital images are now playing an important role in depicting and disseminating pictorial information. As a result, large image databases are being created and used in a number of applications, including criminal identification, multimedia encyclopedia, geographic information systems, online applications for art and art history, medical image archives and trademarks database. These databases typically consist of thousands of images, taking up gigabytes of memory space. While advances in image compression algorithms have alleviated the storage requirements to some extent, the large volume of these images makes it difficult for a user to browse through the entire database. Therefore, an efficient and automatic procedure is required for indexing and retrieving images from databases.

Traditionally, textual features, such as filenames, caption and keywords have been used to annotate and retrieve images. However, there are several problems with these methods. First of all, human intervention is required to describe and tag the contents of the images in terms of a selected set of captions and keywords. In most of the images there are several objects that could be referenced, each having its own set of attributes. Further, we need to express the spatial relationships among the various objects in an image to understand its content. As the size of the image databases grow, the use of keywords becomes not only complex but also inadequate to represent the image content. The keywords are inherently subjective and not unique. Often, the preselected keywords in a given application are context-dependent and do not allow for any unanticipated search. For example, a query for all the images in the database with "people" in it will give good results if we annotate all the images containing people, but for the same annotations, a specific search for images with men or women in it will fail. If the image database is to be shared globally then the linguistic barriers will make the use of keywords ineffective. Another problem with this approach is the inadequancy of uniform textual descriptions of such attributes as color, shape, texture, layout and sketch.

It is generally agreed that image retrieval based on image content is more desirable in a number of applications. As a result, there is a need to automatically extract primitive visual features from the images and to retrieve images on the basis of these features. Humans use color, shape and texture to understand and recollect the contents of an image. Therefore, it is natural to use features based on these attributes for image retrieval. This paper demonstrates the effectiveness of using simple color and shape features for image retrieval.

We have developed a prototype image database system which consists of trademark images. Over 400 trademarks and logo types have been scanned and added to the database. Our goal is to build an image retrieval system which is insensitive to large variations in image scale, rotation and translation. In other words, even if the query image differs from its stored representation in the database in its orientation, position or size, the image retrieval system should be able to correctly match the query image with its prototype in the database.

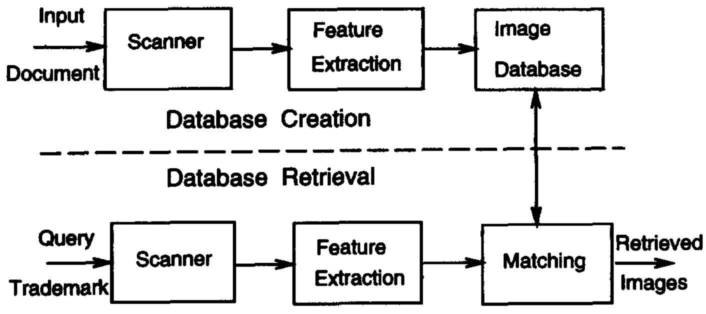

Fig. 1. An image retrieval model.

Figure 1 shows a block diagram of the proposed image retrieval scheme. The input images are preprocessed to extract the features which are then stored along with the images in the database. When a query image is presented, it is similarly preprocessed to extract its features which are then matched with the feature vectors present in the database. A ranked set of images with high matching scores are presented at the output.

2. BACKGROUND

Much of the past research in image retrieval has concentrated on the feature extraction stage. The methods proposed in the literature can be broadly classified into two categories on the basis of the approach used for extracting the image attributes. The *spatial information preserving* methods derive features that preserve the spatial information in the image and it is possible to reconstruct the images on the basis of their feature sets. Representative techniques include polygonal approximation of the object of interest, physics-based modeling and principal component analysis (PCA). The *nonspatial information preserving* methods extract statistical features that are used to discriminate among objects of interest. These include various feature-vector-based approaches, such as histograms and invariant moments. Both these categories extract features based on the cues, such as color and shape that may be present in images.

Examples of recent work with color include color indexing using histogram intersection,[1,2] which considers intersection of color histograms and retrieval using distance and reference color methods,[3] which employ metric distance-based comparison of color histograms. These schemes do not preserve the spatial information in the image, whereas the PCA on color used in reference (4) maintains the spatial adjacencies. Recent work with shapebased features include recognition on the basis of the shape[2] (polygonal approximation of the shape); image representation on the basis of strings,[5,6] which represent the shape of objects as strings and consider string matching techniques for retrieval; comparing images using the Hausdorff distance,[7] which measures the extent to which each point of the model lies near some point of the query and *vice versa*; experiments in point matching techniques,[8] which extract a set of distinctive local features from the query and the model, and then match the resulting point patterns. These shape-based approaches preserve the spatial information in the image.

Retrieval on the basis of color histograms has been shown to outperform retrieval on the basis of shape both in terms of efficiency and robustness. A 3-D histogram intersection technique[1] uses $16 \times 16 \times 16$ bins and matching is relatively fast. The reference color method[3] improves the performance further by using fewer reference color bins. Color histograms are generally invariant to translation and rotation of the images and normalizing the histograms leads to scale invariance. However, color histograms do not incorporate spatial adjacency of pixels in the image and may lead to inaccuracies in the retrieval.

The shape representation schemes proposed in the literature are not generally invariant to large variations of image size, position and orientation. In order to incorporate invariance to rigid motions (rotation and translation) these methods need to be applied for all possible rotations and translations, thereby reducing the speed of the retrievals. When we consider large image databases, this speed reduction can become considerable. We, therefore, need to identify shape features which are either invariant to rigid motions or which can be efficiently computed over a number of possible rigid motions.

Most of the recent work in image database retrieval has concentrated on developing a single concise feature like color, shape or texture. Although color seems to be a highly reliable attribute for image retrieval, situations where color information is not present in the images require the use of shape and/or texture attributes for image retrieval. Retrieval based on a single image attribute might lack sufficient discriminatory information and might not be able to accomodate large scale and orientation changes. Recently, a number of studies have been carried out,[4,9,10] which combine the various features for efficient and effective querying by image content. An attempt has been made in this paper to integrate the image representation on the basis of color and shape to improve the retrieval performance. In the next two sections we describe the features used for representing the color and the shape of the images in our database.

3. IMAGE DATABASE

The image database used in this study was created by scanning a large number of trademarks from several books.[11–13] Currently, the database consists of 150 color images and 250 gray scale images. The trademark pictures were scanned using a Hewlett Packard Scanjet IIcx flatbed scanner at a resolution of roughly 75 dpi. The images are approximately 200 × 200 pixels in size. These 400 trademarks were selected so that the images in our database have similar shape and color to make the retrieval problem more challenging. Most of the black and white trademarks are based on the alphabets in the English language. During preprocessing, the three 1-D (one-dimensional) color histograms and the shape histogram (described in Section 4) were computed from the images and stored in the database instead of the digitized images themselves. With this histogram-based representation each image requires 336 bytes of storage space. The preprocessing stage thus not only saves on expensive computation during matching, but also reduces the database size, since the entire image need not be stored.

4. IMAGE ATTRIBUTES

In order to retrieve images we must be able to efficiently compare two images to determine if they have a similar content. An efficient matching scheme further depends upon the expressiveness and the discriminatory information contained in the extracted features.

Let $\{\mathbf{F}(x,y); x,y=1,2,\cdots,N\}$ be a two-dimensional image pixel array. For color images, $\mathbf{F}(x,y)$ denotes the color value at pixel (x,y). Assuming that the color information is represented in terms of the three primary colors (red, green and blue), the image function can be written as $\mathbf{F}(x,y)=\{F_R(x,y),F_G(x,y),F_B(x,y)\}$. For black and white images, $\mathbf{F}(x,y)$ denotes the gray-scale intensity value at pixel (x,y). Let f represent a mapping from the image space onto the n-dimensional feature space, $X=\{x_1,x_2,\cdots,x_n\}$, i.e.

$$f:\mathbf{F}\rightarrow X,$$

where n is the number of features used to represent the image. The difference between two images, $\mathbf{F}_1$ and $\mathbf{F}_2$, can be expressed as the distance, D, between the respective feature vectors, X_1 and X_2. The problem of retrieval can then be posed as follows: Given a query image $\mathbf{P}$, retrieve an image $\mathbf{M}$ from the image database, $\mathscr{S}$, such that:

$$D(f(\mathbf{P}),f(\mathbf{M}))\leqslant D(f(\mathbf{P}),f(\mathbf{F})),\quad \forall\mathbf{F}\in\mathscr{S},\mathbf{F}\neq\mathbf{M}.$$

The performance of the retrieval system depends on the particular feature representation and the matching scheme employed. We have used features based on color and shape for the retrieval of trademark images. Color in an image is represented by using three 1-D color histograms, one each for the R, G and B components of the color image. A histogram intersection scheme as defined in reference (14) and a matching scheme based on Euclidean distance are both used for image retrieval based on color.

A concise and quantitative description of the object shape is a challenging problem. We try to represent the shape of an object from its edge image. The Canny edge operator[15] is used to retrieve the edge points and a histogram of the directions of the edge points is used to represent the shape. We note that this representation of shape is not rotation or scale invariant. Scale invariance can be achieved by normalizing the histograms.

We discuss later a method to incorporate rotation invariance in matching on the basis of shape. A histogram intersection technique is again used for shape-based retrieval.

4.1. *Color*

Color is an important attribute for image retrieval. Humans seem not to be as affected by small variations in color as by variations in gray-level values. A number of color space representation schemes (e.g. RGB and HSI) have been reported in the literature. The RGB space has been widely used due to the availability of images in the RGB format from image scanners. Regardless of the color space, color information in an image can be represented either by a single 3-D histogram or three separate 1-D histograms. These color representation schemes are essentially invariant under rotation and translation of the input image. A suitable normalization of the histograms(s) also provides scale invariance. Let $H(i)$ be a histogram of an image, where the index i represents a histogram bin. Then, the normalized histogram I is defined as follows:

$$I(i)=\frac{H(i)}{\Sigma_i H(i)}.$$

Techniques for matching on the basis of color histograms are faster compared to matching on the basis of shape and texture attributes. We have used three 1-D histograms in our experiments.

4.1.1. *Histogram intersection.* Let I_R, I_G and I_B be the normalized color histograms of an image in the database and let Q_R, Q_G and Q_B be the normalized color histograms of the query image. The similarity between the query image and a stored image in the database, $S_c^{HI}(I,Q)$, is given by the following equation:

$$S_c^{HI}(I,Q)=\frac{\Sigma_r\min(I_R(r),Q_R(r))+\Sigma_g\min(I_G(g),Q_G(g))+\Sigma_b\min(I_B(b),Q_B(b))}{\min(|I|,|Q|)3}. \quad (1)$$

Note that the value of $S_c^{HI}(I,Q)$ lies in the interval $[0,1]$. If the histograms I and Q are identical then $S_c^{HI}(I,Q)=1$. The main advantage of the use of color histograms is that they are rotation and translation invariant for a constant background. If either of the two images (query or prototype) is completely contained in the other, then $S_c^{HI}(I,Q)=1$.

4.1.2. *Distance method.* Euclidean distance is used as a metric to compute the distance between the feature vectors. Let I_R, I_G and I_B be the normalized color histograms of an image in the database and let Q_R, Q_G and Q_B be the normalized color histograms of the query image. The similarity between the query image and a stored image in the database, $S_c^{ED}(I,Q)$, is given by the following equation:

$$S_c^{ED}(I,Q) = 1.0 - \sqrt{\frac{\Sigma_r(I_R(r)-Q_R(r))^2 + \Sigma_g(I_G(g)-Q_G(g))^2 + \Sigma_b(I_B(b)-Q_B(b))^2}{2\times 3}}. \quad (2)$$

Note that the value of $S_c^{ED}(I,Q)$ lies in the interval $[0,1]$. If images I and Q are identical then $S_c^{ED}(I,Q)=1$.

Figure 2 shows three database images [(a) and (b) are similar in color but (c) is different] and their respective R, G and B histograms. The three 1-D histograms each containing 16 bins have been concatenated into a single 48-bin histogram in the figure [(*d*)–(*f*)]. S_c^{HI} and S_c^{ED} represent the similarity value based on the histogram intersection and the Euclidean distance method, respectively. Note that $S_c(d,e) > S_c(d,f)$ and $S_c(d,e) > S_c(e,f)$, where S_c represents the color-based similarity index (either S_c^{HI} or S_c^{ED}).

4.2. *Shape*

In the absence of color information or in the presence of images with similar colors it becomes imperative to use additional image attributes for an efficient retrieval. Incorporating rotation invariance in shape matching generally increases the computational requirements. We, therefore, have emphasized a faster, yet robust, scheme for retrieval on the basis of shape.

We describe the shape information contained in an image on the basis of its significant edges. A histogram of the edge directions is used to represent the shape attribute. The edge information contained in the database images is generated in the preprocessing stage using the Canny edge operator.[15] A histogram intersection technique [similar to Equation (1)] is used for shape-based retrieval.

There are many advantages and limitations in representing an image with its edge directions.

- Use of edge directions captures the general shape information.
- The histogram of the edge directions is invariant to translation in the image. Thus, the positions of the objects in the image have no effect on the edge directions. This may also turn out to be a limitation, as two totally different images may yield similar edge direction histograms.
- The use of edge directions is inherently not scale invariant. Two images identical in every respect except their size will yield different number of edge points and hence different histograms. In order to have invariance to scale, we normalize the histograms with respect to the number of edge points in the image. A drawback of normalized histograms is its inability to match parts of images. If an image Q is a part of an image I, then the histogram of Q is contained within the histogram of I. Normalizing the histograms does not satisfy this property.
- A histogram of the edge directions is also not invariant to rotation. A shift of the histogram bins during matching partially takes into account rotation of the images. A rotation of the image shifts each of the edge directions by the amount of rotation. However, due to the effect of quantization of the edge directions into bins, the effect of rotation is more than a simple shift in the bins. Rotation also affects the membership in the bins. For example, if a quantization of 45° (bin size = 45°) is used, then two edge points with directions of 30 and 40° will fall into the same bin. However, if the same image is rotated by 10° then these two points will fall into adjacent bins. We note that for a rotation of 45° there is no such effect. To reduce this effect of rotation we smooth the histograms. An histogram can be treated as a 1-D discrete signal. Smoothing can then be defined as:

$$I_s[i] = \frac{\Sigma_{j=i-k}^{i+k} I[j]}{2k+1},$$

where I_s is the smoothed histogram, I is the original histogram and the parameter k determines the degree of smoothing. In our experiments we used $k=1$.

- Edge directions are affected by the nature of the edge, i.e. black on white or white on black edge. These two types of edges generally differ in their directions by 180°. For this reason, an empty square will produce a different histogram than a partially filled square. The empty square will produce four significant bins, whereas the partially filled square will produce eight significant bins, four for the outer edges and four for the inner edges.
- The matching results depend on the bin size. By choosing a larger bin size, the matching speed is increased. However, this also reduces the accuracy in case of an arbitrary rotation of the image. A rotation not only causes a shift in the histogram bins but also affects the bin membership. Use of a very small bin size reduces the matching speed (since the number of bins increases). Use of a small bin size also requires that the

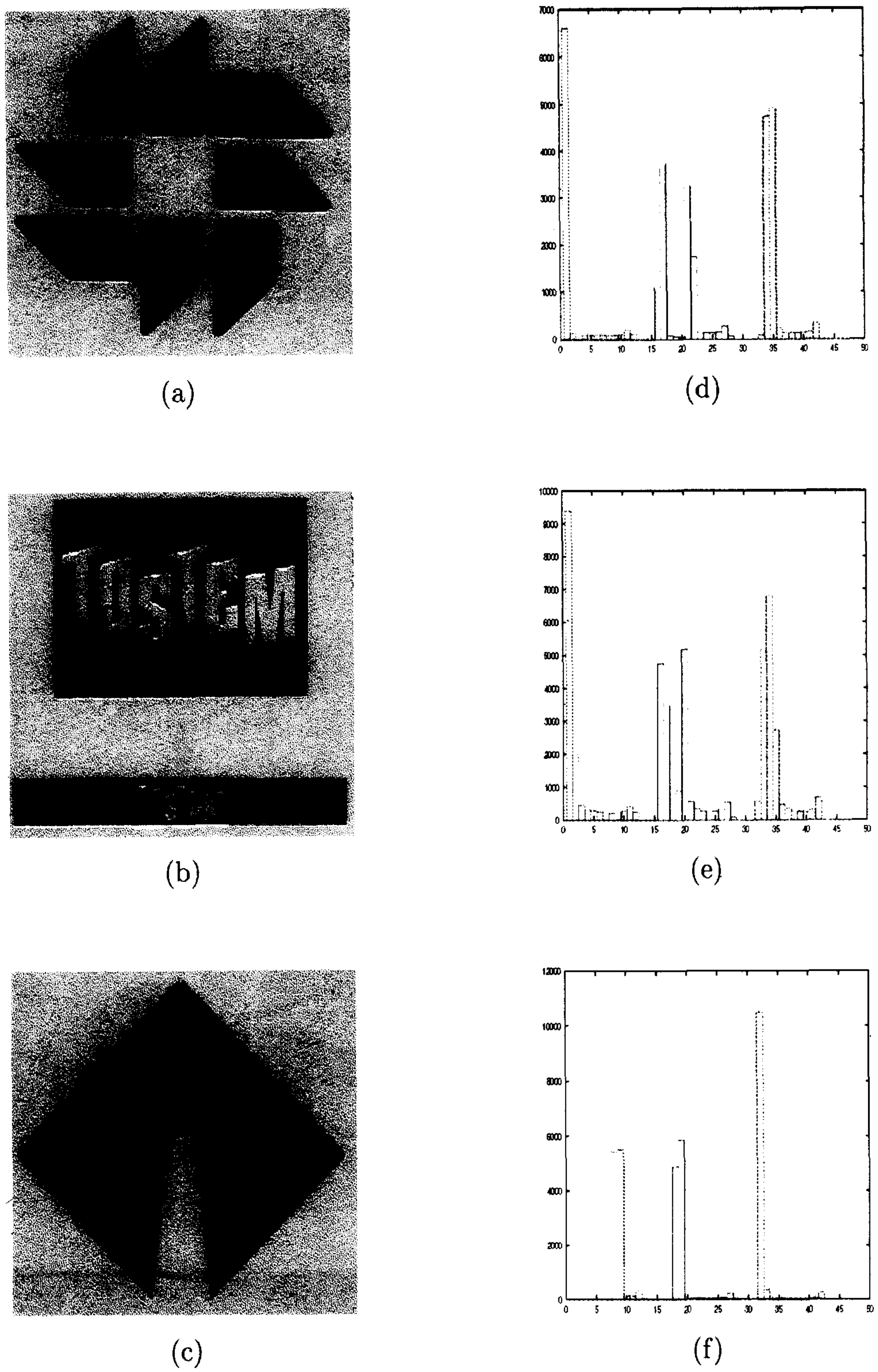

Fig. 2. Color histograms for three database images: (a), (b), and (c) show three color images in the database and (d), (e) and (f) show the corresponding histograms: $S_c^{HI}(d,e) = 0.65$, $S_c^{HI}(d,fe) = 0.07$, $S_c^{HI}(e,f) = 0.12$; $S_c^{ED}(d,e) = 0.85$, $S_c^{ED}(d,f) = 0.61$, $S_c^{ED}(e,f) = 0.65$.

edge directions be found to a very high degree of accuracy.

Figure 3 shows three database images [(a) and (b) are similar but (c) is different] and their respective edge angle histograms. S_s represents the shape-based similarity value of the images. Note that $S_s(g,h) > S_s(g,i)$ and $S_s(g,h) > S_s(h,i)$, where (g)–(i) represent the shape histograms for the three database images.

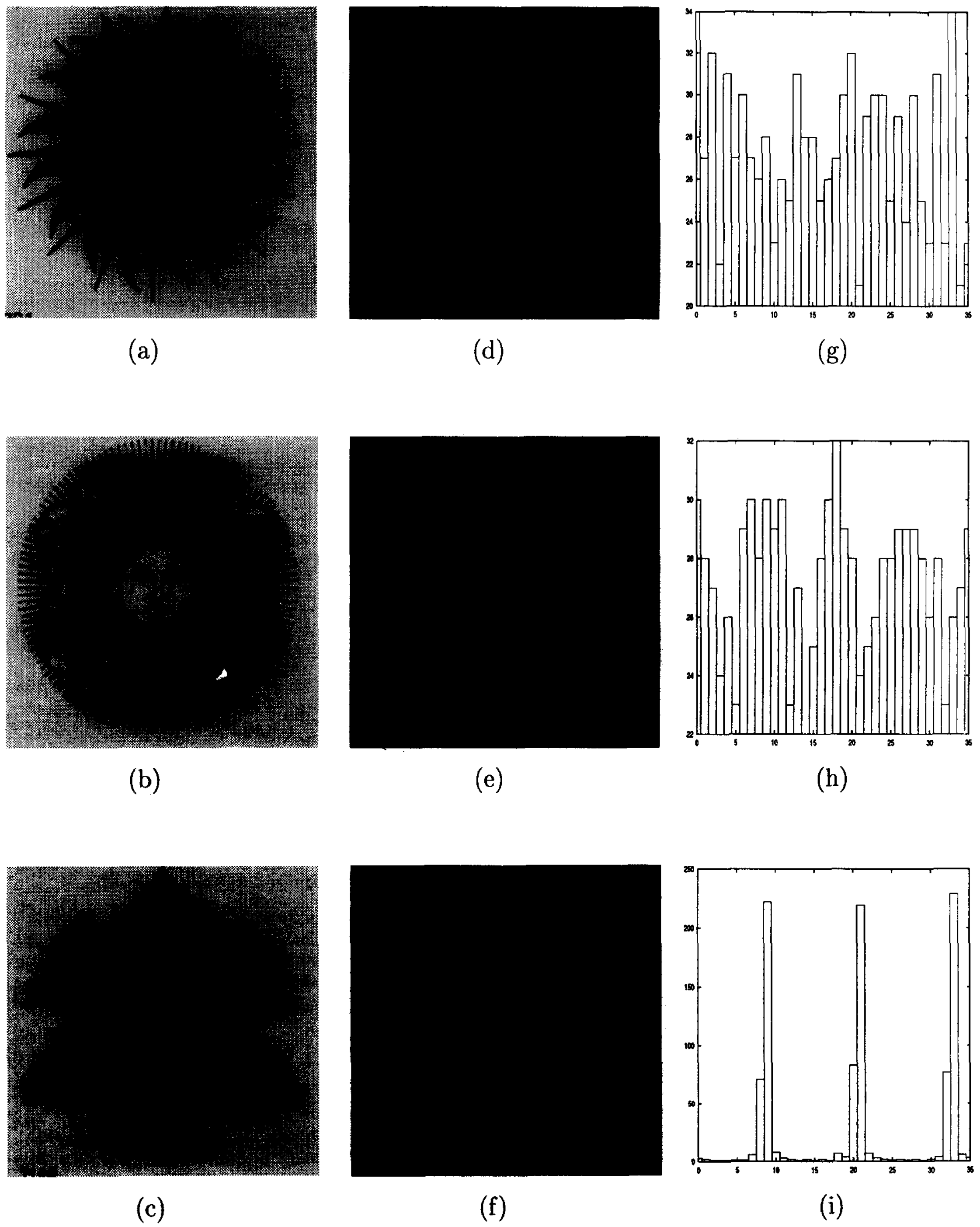

Fig. 3. Shape histograms for three database images; (a)–(c) show three database images, (d)–(f) show the corresponding edge images and (g)–(i) show the corresponding shape histograms: $S_s(g, h) = 0.93$, $S_s(g, i) = 0.25$, $S_s(h, i) = 0.26$.

4.3. *Integration of color and shape attributes*

The use of a single image attribute for retrieval may lack sufficient discriminatory information and might not be able to support large variations in image orientation and scale. In order to increase the accuracy of the retrievals we need to integrate the results obtained from the query based on individual features. The output of a query on the basis of either color or shape is a ranked set (based on the similarity value) of database images. An integrated rank of a retrieved image can be computed from either the ranks of the retrieved image in the individual queries (query on the basis of color, or query on the basis of shape) or the actual similarity value of the retrieved image in the individual queries. We integrated the results of the color- and shape-based queries on the rank of the retrieved images. This integrated scheme performed worse than each of the individual feature-based queries. Using the individual ranks may not be very effective since it does not use the similarity value between the query and the retrieved image. The rank of the retrieved images may give a false sense of similarity when actually the similarity value may be very low. Hence, we need to look into an integration scheme that takes into account the similarity values between the images.

We integrated the results of the shape-based retrieval and the color-based retrieval by combining the associated similarity values. Let Q be a query image and I be a database image. Let S_c be the similarity index between Q and I on the basis of color (either S_c^{HI} or S_c^{ED}) and S_s be the similarity index between Q and I on the basis of shape. We define an integrated similarity index S_t between Q and I as:

$$S_t = \frac{w_c S_c + w_s S_s}{w_c + w_s},$$

where w_c and w_s are the weights assigned to the color- and shape-based similarity, respectively. A set of top 10 images on the basis of the total similarity index S_t are presented as a result of the query. Note that S_t lies in the interval $[0, 1]$. We have used equal weights ($w_c = w_s = 1$) in our experiment. Another way of determining the weights is on the basis of the accuracy of the individual feature-based queries. In general, color-based queries are more accurate than shape-based queries and this fact can be used to assign a higher weight to the color-based similarity values.

5. EXPERIMENTAL RESULTS

Our aim is to develop an efficient and effective image retrieval scheme which is insensitive to large variations in the image orientation, position and size. We have conducted experiments to study the behavior of our image retrieval system in the presence of these variations. An efficient content-based retrieval scheme must have the following features:

- Accuracy and stability: The retrieval scheme must be accurate, i.e. the retrieved images must resemble the query image. We classify a retrieval as accurate if for a given query image the perceptually (to a human) most similar image in the database is retrieved by the system as the topmost retrieval. By robustness we refer to the fact that the performance should be stable for all types of queries, i.e. the system must not break down under specific cases. In order to test the stability and robustness of the system, we have tried to retrieve every image in the database under a variety of conditions.
- Speed: It is desirable to have an efficient retrieval scheme. Since image databases typically have thousands of images, the retrieval scheme must be "real-time". The total time taken for a retrieval on the entire database was measured in our experiments.

Each of the 1-D color histograms is represented by 16 bins, each spanning 16 intensity values. The shape histogram of the edge directions is represented by 36 bins, each spanning 10°. We next describe in detail the various experiments done to test the accuracy, robustness and efficiency of the system.

5.1. *Accuracy and stability*

The following experiments were conducted for retrieval on the basis of both shape and color. First each of the database images was presented as a query image. In order to study the affect of variations in orientation of the logo types, we study the performance of the system for arbitrary rotations of the database images. Each image in the database is arbitrarily rotated and presented as the query image. Next, the performance was studied under variations in size of the trademarks. Each of the database images was scaled and presented as the query. Finally, random noise was added to the database images and the performance of the system was studied under the presence of noise.

The four test cases are presented below:

- *Normal (N)*: Every image in the database itself was presented as the query image.
- *Rotated (R)*: Every image in the database was rotated arbitrarily and then presented as the query image.
- *Scaled (S)*: Every image in the database was scaled and presented as the query image.
- *Noisy (UN)*: A random noise of 5% as added to every image and presented as the query. The noise model is described in Section 5.1.1.

For future experiments we intend to scan rotated versions of the database images at various resolutions to incorporate natural variations in the size and orientation of the trademarks.

Figure 4 shows the rotated, scaled and noisy versions of six database images.

5.1.1. *Noise model.* To test the accuracy and stability of our system we added random noise to the database images and presented the noisy images as the query.

We have used an independent and identically distributed (*i.i.d.*) additive uniform noise as the noise model. In the case of color images a noise of $k\%$ was added by choosing $k\%$ of the image points at random and assigning each of the red, green and blue intensity values a random value within the range $(0, 255)$. In the case of retrieval by shape $k\%$ of the image points were chosen at random and their edge directions were assigned random values in the range $(0, 360)$. In our experiments, we used $k = 5$.

5.1.2. *Color.* For retrieval on the basis of color, the experiments were conducted on the 150 color images in our database. Tables 1 and 2 present the results of retrieval based on histogram intersection and Euclidean distance, respectively, where n refers to the position of the correct retrieval. The retrieval performance

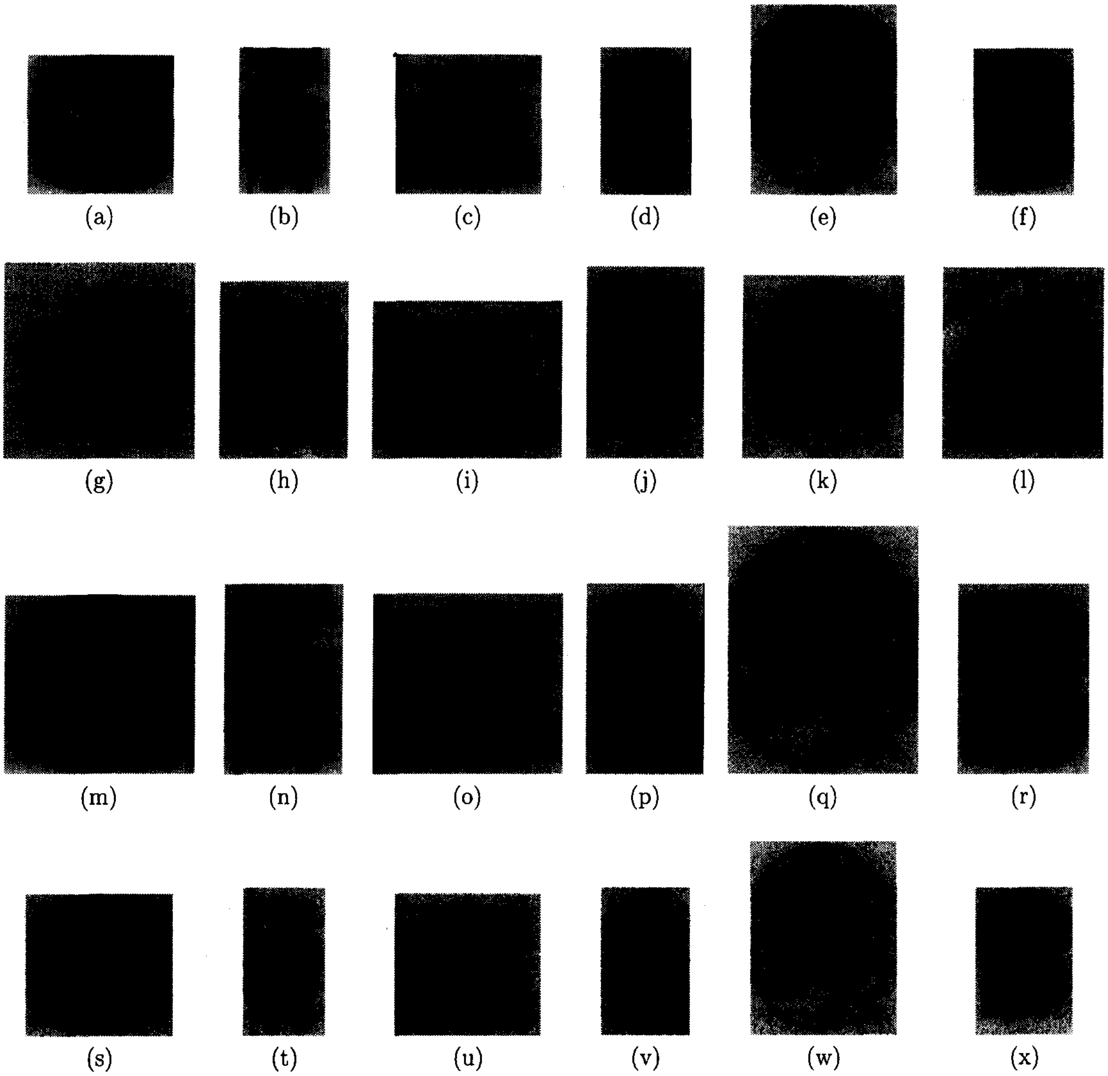

Fig. 4. A sample of test images; (a)–(f) show six database images, (g)–(l) show the corresponding rotated images, (m)–(r) show the corresponding scaled images and (s)–(x) show the corresponding noisy images.

Table 1. Color-based retrieval results using the Histogram Intersection scheme: n refers to the position of the correct, N: normal query, R: rotated query, S: scaled query, UN: query with a noisy image; the last column indicates percentage of time the query image was not retrieved in the top 20 matches

Query nature	$n=1$ (%)	$n\leqslant 2$ (%)	$n\leqslant 3$ (%)	$n\leqslant 4$ (%)	$n\leqslant 5$ (%)	$n\leqslant 20$ (%)	Not retrieved (%)
N	100	100	100	100	100	100	0
R	90.71	93.57	94.29	95	97.86	100	0
S	100	100	100	100	100	100	0
UN	92.14	95.71	95.71	97.14	98.57	100	0

was better in the presence of scale changes than in the presence of rotated or noisy images. We observe that the noise introduced in discretizing the analog rotation affects the accuracy. The worst-case retrieval accuracy of the system was 92%. The performance of the two schemes (Euclidean distance and histogram intersection) is comparable, but the number of false matches in the case of the Euclidean distance method is less, resulting in a correct match within the top five retrievals.

5.1.3. *Shape*. For retrieval on the basis of shape, the experiments were conducted on the entire database of 400 images.

Table 3 presents the results of retrieval on the basis of shape, where n corresponds to the position of the

Table 2. Color-based retrieval results using Euclidean distance: n refers to the position of the correct retrieval, N: normal query, R: rotated query, S: scaled query, UN: query with a noisy image; the last column indicates percentage of time the query image was not retrieved in the top 20 matches

Query nature	$n=1$ (%)	$n \leqslant 2$ (%)	$n \leqslant 3$ (%)	$n \leqslant 4$ (%)	$n \leqslant 5$ (%)	$n \leqslant 20$ (%)	Not retrieved (%)
N	100	100	100	100	100	100	0
R	92.14	93.57	95	98.57	99.29	100	0
S	100	100	100	100	100	100	0
UN	92.14	95	97.86	100	100	100	0

Table 3. Retrieval results on the basis of shape: n refers to the position of the correct retrieval, N: normal query, R: rotated query, S: scaled query, UN: query with a noisy image; the last column indicates percentage of time the query image was not retrieved in the top 20 matches

Query nature	$n=1$ (%)	$n \leqslant 2$ (%)	$n \leqslant 3$ (%)	$n \leqslant 4$ (%)	$n \leqslant 5$ (%)	$n \leqslant 20$ (%)	Not retrieved (%)
N	100	100	100	100	100	100	0
R	63	69.5	73	75.5	76.5	91	9
S	95.5	97	97.75	98.25	98.25	99.75	0.25
UN	94.5	97.75	97.75	99.25	99.25	100	0

correct retrieval. Experiments for the cases S and UN presented much better performance in terms of accuracy. We notice that our simple shape measure is not very effective in retrieving rotated images. The low accuracy for the case R is due to the fact that edge directions were quantized into bins. A rotation in an image not only shifts the bin but also changes the membership of the corresponding bins.

5.1.4. *Integration of color and shape attributes.* The results of the integrated query (integrating queries on the basis of both color and shape) are presented for the 150 color images. The choice of w_c and w_s, the weights assigned to the similarity indices of the results of the color and shape based queries, respectively, is determined empirically. We present the results with $w_c = w_s = 1$. Another possibility is to choose the weights on the basis of the accuracy of the individual feature-based queries. Since the query on the basis of color provides 90% accuracy in the worst case and the shape-based query provides an accuracy of 63% in the worst case, we can choose $w_c = 9$ and $w_s = 6$.

Table 4 presents the results of retrieval on the basis of integration of the color (histogram intersection scheme) and shape queries, where n corresponds to the position of the correct retrieval. With retrieving scaled and noisy images, the topmost retrieved image is the correct result for each of the presented query. In the case of rotated images the query matches the correct image in the topmost position in all but four cases and even then the correct match occurs within the top three positions.

We expect that each of the two feature-based schemes assigns a high similarity value to similar images, but they may also assign a high similarity value of perceptually different images (false matches). This may affect the rank order of the correct match and hence the accuracy in the individual feature case. Integrating the two schemes reduces the number of false retrievals as it is highly unlikely that a pair of perceptually different images is assigned a high similarity value in both the schemes.

5.2. *Computational requirements*

It is desirable to have a fast scheme for retrieving images from a database. The retrieval scheme described above sequentially computes the distance of a given query image with every image in the database. The CPU time (on a SUN Sparc 10) for a color-based query on a database containing 150 images was approximately 1 s, while the time for a shape-based query

Table 4. Integrated retrieval results: n refers to the position of the correct retrieval, N: normal query, R: rotated query, S: scaled query, UN: query with a noisy image; the last column indicates percentage of time the query image was not retrieved in the top 20 matches

Query nature	$n=1$ (%)	$n \leqslant 2$ (%)	$n \leqslant 3$ (%)	$n \leqslant 4$ (%)	$n \leqslant 5$ (%)	$n \leqslant 20$ (%)	Not retrieved (%)
N	100	100	100	100	100	100	0
R	97.14	99.29	100	100	100	100	0
S	100	100	100	100	100	100	0
UN	100	100	100	100	100	100	0

ranged between 1 and 2s on a database containing 400 images. These times do not include the preprocessing time for extracting the feature histograms (color and edge angle histograms).

In order to speed up the computationally expensive search over the entire database, we first clustered the database images. A branch and bound algorithm for computing the k-nearest neighbors[16] was then used on the clustered database to efficiently retrieve the images.

5.1.2. *Clustering color images.* In order to reduce the search space we have clustered the database images using CLUSTER,[17] a squared-error clustering algorithm. If the clustering or image partition is "good" then the query image should be compared only with the cluster centers in order to decide its cluster membership. Euclidean distance is used to compare the distance of the query image from the cluster centers. Once the cluster membership of a query image is established then it needs to be matched only with images in that cluster. The database images were clustered into five clusters based on the CLUSTER algorithm. It might not be efficient to cluster the 150 color database images using the high-dimensional (48 color features) feature vector. In order to have a more meaningful clustering, the database images were clustered using six extracted features. The means and variances of the three color histograms were used as the extracted color features. Figure 5 shows the database images closest to the five cluster centers and the Chernoff[18] of the cluster centers. Note that the "average" of a cluster does not correspond to a specific database image. The extracted features are visualized using Chernoff faces. The means of the R, G and B histograms are mapped onto the area of the face, shape of the face and length of the nose, respectively. The variances of the three color histograms are mapped to the location of the mouth, the curve of the smile and the width of the mouth.

Experiments were first conducted by presenting each of the color images as input and then by presenting noisy versions of the color images as the input. The results are summarized in Table 5. A major drawback of this scheme of choosing the closest cluster center is that if a wrong cluster is chosen then it is impossible to retrieve the correct image. We may thus need to consider an optimal search over the entire database ad retrieve a set of closely matched images. We next look into an optimal and efficient branch and bound algorithm for computing the k-nearest neighbors.[16]

5.2.2. *Optimal search using branch and bound method.* The basic approach of the branch and bound algorithm to compute the k-nearest neighbors is to divide the training data (here the database images) into a hierarchy of disjoint subsets and then apply the

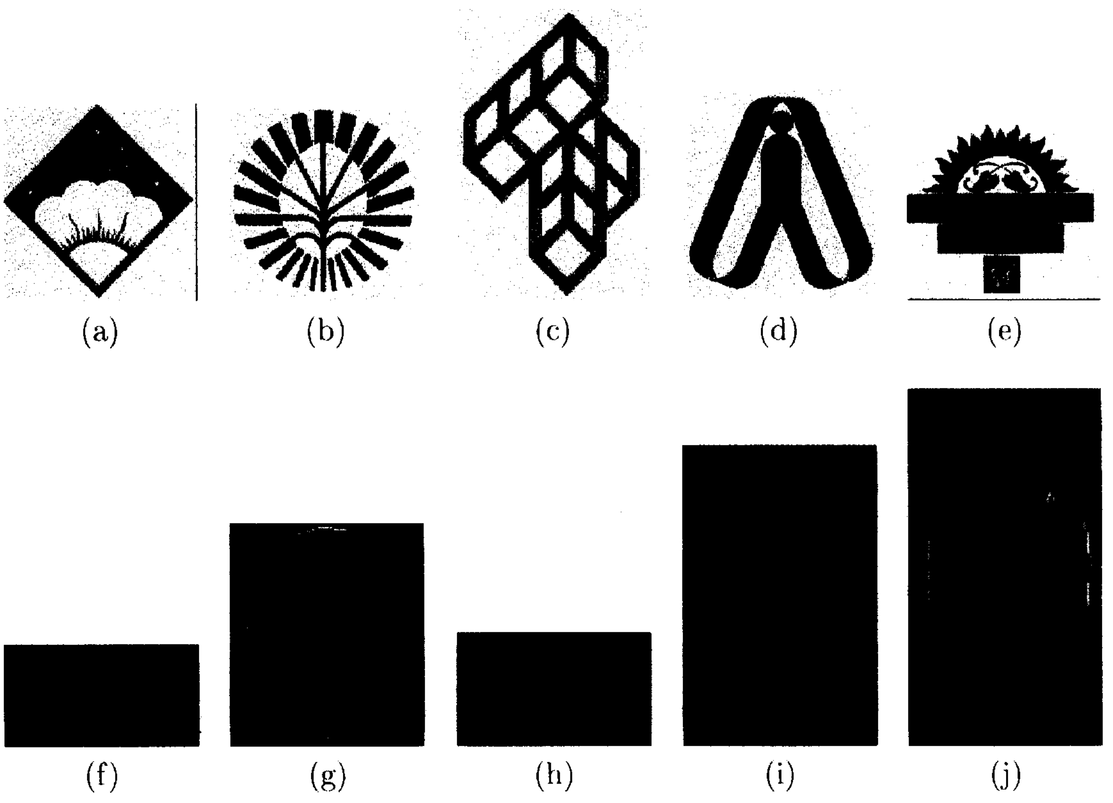

Fig. 5. Database images closest to the cluster centers: (a)–(e) show the database images closest to the five cluster centers; (f)–(j) show the Chernoff faces of the cluster centers.

Table 5. Retrieval results on the clustered images: n refers to the position of the correct retrieval, the last column indicates percentage of time the query image was not retrieved in the top 20 matches

Query nature	$n=1$ (%)	$n \leqslant 2$ (%)	$n \leqslant 3$ (%)	$n \leqslant 4$ (%)	$n \leqslant 5$ (%)	$n \leqslant 20$ (%)	Not retrieved (%)
Normal	100	100	100	100	100	100	0
Noisy	81	83	84	85	85	85	15

Table 6. Average number of distance computations for color images: K is the number of neighbors considered; C shows the case when cluster information was used; Column U shows the case when no cluster information was used; (%) improvement is against an exhaustive search on the entire database

K	C(avg)	(%) Improvement	U(av.)	(%) Improvement
1	61.77	56	94.62	32
2	109.62	22	131.31	6
3	118.96	15	135.54	3
4	123.99	11	137.45	2
5	126.85	9	138.70	1

branch and bound tree search to the resultant groups. We have implemented this algorithm which takes as input a general binary tree and the test pattern and presents as output the k-nearest neighbors. The cluster information is used to divide the training set (database images) into disjoint subsets and group them into a hierarchical binary tree according to the intercluster distances. We tested the efficiency of the branch and bound search strategy using this cluster information and also for an arbitrary clustering. Our goal was to study the effect of clustering the database images on the number of distance computations (measure of efficiency of the branch and bound algorithm).

Table 6 presents the result of branch and bound method. Note that the use of cluster information reduces the search space by 50% while finding the nearest neighbor of a query image, without affecting the optimality of the result. On an average, the branch and bound method finds the five-nearest neightbors with 10% fewer distance calculations compared to a sequential search.

6. SUMMARY

With trademark images, retrieval accuracy on the basis of color is better than that based on shape. Integrating the results of the color- and shape-based queries provides a better and more robust performance than either of the individual feature-based queries. Humans also use a combination of features (color, shape and texture) to recognize objects and do not rely on any one individual feature. We feel that a combination of simple features which can be easily extracted is more effective and efficient. The speed of retrievals can also be increased by using a branch and bound method to compute the nearest neighbors for the query image without affecting the robustness of the system. Future work deals with adding other shape features to our current system and increasing the database size.

Acknowledgements—We would like to thank Chitra Dorai for implementing the front end to the branch and bound algorithm. In addition, we acknowledge the contributions of Chitra Dorai and Sharath Pankanti to the final form of this manuscript.

REFERENCES

1. M. J. Swain and D. H. Ballard, Color indexing, *Intl. J. Comput. Vis.* **7**(1), 11–32 (1991).
2. R. Schettini, Multicolored object recognition and location, *Pattern Recognition Lett.* **15**, 1089–1097 (November 1994).
3. B. M. Mehtre, M. S. Kankanhalli, A. D. Narsimhalu and G. C. Man, Color matching for image retrieval, *Pattern Recognition Lett.* **16**, 325–331 (March 1995).
4. A. Pentland, R. W. Picard and S. Sclaroff, Photobook: Content-based manipulation of image databases, *SPIE Vol 2185: Storage and Retrieval for Image and Video Databases II* (February 1994).
5. G. Cortelazzo, G. A. Mian, G. Vezzi and P. Zamperoni, Trademark shapes description by string-matching techniques, *Pattern Recognition* **27**(8), 1005–1018 (1994).
6. P. W. Huang and Y. R. Jean, Using 2D C^{+}-strings as spatial knowledge representation for image database systems, *Pattern Recognition* **27**(9), 1249–1257 (1994).
7. D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, Comparing images using the Hausdorff distance, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**, 850–863 (September 1993).
8. D. J. Kahl, A. Rosenfeld and A. Danker, Some experiments in point pattern matching, *IEEE Trans. Syst. Man Cybernet.* **10**, 105–116 (February 1980).
9. C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic and W. Equitz, Efficient and effective querying by image content, *J. Intell. Inf. Syst.* **3**, 231–262 (1994).
10. B. Holt and L. Hartwick, Visual image retrieval for applications in art and art history in *SPIE Vol. 2185: Storage and Retrieval for Image and Video Databases II*, pp. 70–81 (February 1994).
11. T. Igarashi (Editor) *World Trademarks and Logotypes*. Graphic-sha, Tokyo (1983).
12. T. Igarashi, (Editor) *World Trademarks and Logotypes II: A Collection of International Symbols and their Applications*. Graphic-sha, Tokyo (1987).
13. *Collection of Trademarks and Logotypes in Japan*. Graphic-sha, Tokyo (1973).
14. M. P. Dubuisson and A. K. Jain, Fusing color and edge information for object matching, *Proc. First IEEE Intl. Conf. Image Process.* Austin (November 1994).

15. J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-8**, 679–698 (November 1986).
16. K. Fukunaga and P. M. Narendra, A branch and bound algorithm, for computing *k*-nearest neighbors, *IEEE Trans. Comput.* 750–753 (1975).
17. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, New Jersey (1988).
18. H. Chernoff, The use of faces to represent points in *k*-dimensional space graphically, *J. Am. Statist. Ass.* **68**, 361–368 (1973).

About the Author—ANIL JAIN is a University Distinguished Professor and Chairman of the Department of Computer Science at Michigan State University. His current research interests are computer vision, image processing, neural networks and pattern recognition. He has made significant contributions and published a large number of papers on the following topics: statistical pattern recognition, exploratory pattern analysis, neural networks, Markov random fields, texture analysis, interpretation of range images and 3D object recognition. Several of his papers have been reprinted in edited volumes on image processing and pattern recognition. He received the best paper awards in 1987 and 1991, and received certificates for outstanding contributions in 1976, 1979 and 1992 from the Pattern Recognition Society. Dr Jain served as the Editor-in-Chief of the IEEE Transactions on Pattern Analysis and Machine Intelligence (1991–1994) and currently serves on the editorial boards of Pattern Recognition journal, Pattern Recognition Letters, Journal of Mathematical Imaging, Journal of Applied Intelligence and IEEE Transactions on Neural Networks. He is the co-author of the book Algorithms for Clustering Data (Prentice-Hall, 1988), has edited the book Real-Time Object Measurement and Classification (Springer-Verlag, 1988) and has co-edited the books Analysis and Interpretation of Range Images (Springer-Verlag, 1989), Neural Networks and Statistical Pattern Recognition (North-Holland, 1991), Markov Random Fields: Theory and Applications (Academic Press, 1993) and 3D Object Recognition (Elsevier, 1993). Dr Jain is a Fellow of the IEEE. He was the Co-General Chairman of the 11th International Conference on Pattern Recognition, Hague (1992), General Chairman of the IEEE Workshop on Interpretation of 3D Scenes, Austin (1989), Director of the NATO Advanced Research Workshop on Real-time Object Measurement and Classification, Maratea (1987) and co-directed NSF supported Workshops on "Future Research Directions in Computer Vision", Maui (1991), "Theory and Applications of Markov Random Fields", San Diego (1989), and "Range Image Understanding", East Lansing (1988). Dr Jain was a member of the IEEE Publications Board (1988–1990) and served as the Distinguished Visitor of the IEEE Computer Society (1988–1990). He is currently a Distinguished Lecturer of the IEEE Computer Society's Asia-Pacific Lectureship Program.

About the Author—ADITYA VAILAYA received his B.Tech degree from Indian Institute of Technology, Delhi, in 1994. He is currently working towards his M. S. degree at Michigan State University, East Lansing. His research interests include image databases, image processing, pattern recognition, computer vision and robotics.