Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computer Vision and Image Understanding 94 (2004) 193–233

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Region-based image retrieval using integrated color, shape, and location index

B.G. Prasad,*,[1] K.K. Biswas, and S.K. Gupta*

Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi 110 016, India

Received 1 December 2002; accepted 29 October 2003

Abstract

A technique to retrieve images by region matching using a combined feature index based on color, shape, and location is presented within the framework of MPEG-7. Dominant regions within each image are indexed using integrated color, shape, and location features. Various combinations of regions are also indexed. The resulting indices and related metadata are stored in a Hash structure, where similar images tend to form clusters. The retrieval process is non-cascading and images can be retrieved based on color, shape or location and also based on a combined color–shape–location index. Results obtained show that retrieval effectiveness increases in non-cascaded region-based querying by combined index.

Keywords: Indexing techniques; Image database; Content-based image retrieval; Color-based feature selection; Color image segmentation; Color image indexing; Combining color and shape information; Integrated color–shape–location index

1. Introduction

There has been a lot of interest in content-based image retrieval (CBIR) using visual features over the last decade. An overview of work in this area can be found

* Corresponding authors. Fax: +91-11-2658-1060 (S.K. Gupta).
E-mail addresses: bgprasad@cse.iitd.ernet.in (B.G. Prasad), kkb@cse.iitd.ernet.in (K.K. Biswas), skg@cse.iitd.ernet.in (S.K. Gupta).
[1] On leave from Department of Computer Science and Engineering, P.E.S. College of Engineering, Mandya 571 401, Karnataka State, India.

1077-3142/$ - see front matter
doi:10.1016/j.cviu.2003.10.016

in [1–7]. CBIR is like an information filter process and is expected to provide a high percentage of relevant images in response to user query. It should conform to human perception of visual semantics. In general, the image features tend to capture only some of the aspects of image similarity and hence it becomes difficult to clearly specify what or how a user should initiate queries. The presence of large volumes of digital repositories leads to many schemes of indexing and retrieval of such data (e.g., QBIC [8,9], Netra [10], VisualSeek [11], Chabot [12], Blobworld [13], etc). In all these cases, the user is interested in seeking the most similar images to his query.

Color is one of the most important image indexing features employed in CBIR. Schettini et al. [14] and Del Bimbo [4] provide a comprehensive survey of various methods employed for color image indexing and retrieval in image databases. Some of the popular methods to characterize color information in images are color histograms [15,16], color moments [17], and color correlograms [18]. Though all these methods provide good characterization of color, they have the problem of high-dimensionality. This leads to more computational time, inefficient indexing, and low performance. To overcome these problems, use of SVD [16], dominant color regions approach [19,20], and color clustering [21,22] have been proposed. The work in [23] presents a scheme for indexing and retrieval of color image data. In [24], a signature bit string representation for color feature is used. A grid is superimposed on the entire image to obtain partition-based image retrieval, but is not rotation invariant. No access structure is used to support the signature-based representation of color. The proposal of using variable number of color histograms for color representation, depending on number of colors in the image is discussed in [25]. It is shown to be better than using global histograms. In [26], a flexible sub-block image retrieval algorithm robust to translation, lighting change, and object appearance is proposed. A reduced color space is used to overcome the problems associated with global color histograms. The global color-based methods suffer from problems of non-invariance and large storage requirements. Hence other features like shape, texture, and spatial location are added to the feature space to enhance the retrieval efficiency and effectiveness.

Shape is an important feature for perceptual object recognition and classification of images. For a survey of various shape methods used for content-based image retrieval refer to [2,4,27]. Many techniques such as chain code, polygonal approximations, curvature, Fourier descriptors, radii method, and moment descriptors have been proposed and used in various applications [28]. Recently, techniques using shape measure as an important feature have been used for CBIR. Features such as moment invariants and area of region have been used in [8,29], but do not give perceptual shape similarity. Cortelazzo et al. [30] used chain codes for trademark image shape description and string matching technique. The chain codes are not normalized and string matching is not invariant to shape scale. Jain and Vailaya [31] proposed a shape representation based on the use of a histogram of edge directions. But these are not normalized to scale and computationally expensive in similarity measures. Mehrotra and Gary [32] used coordinates of significant points on the boundary as shape representation. It is not a compact representation and the similarity measure is computationally expensive. Jagadish [33] proposed shape decomposition into a number of rectangles and two pairs of coordinates for each rectangle are used to represent the shape, but it is not rotation invariant. A region-based shape representation and indexing scheme that is translation, rotation, and scale invariant is proposed in [34]. Compared to Fourier method, it is shown to give better retrieval performance but works only on binary images and has been applied on only 2D planar images. Moreover the shapes with similar eccentricity but different shapes are also retrieved as matching images. In [35] it is shown that applying shape-based indexing of color images using a similar grid-based representation as in [34], but different indexing and similarity measures, better retrieval results are obtained for color image regions. Further, along with color, shape is added as an additional feature to index the extracted regions within the images. A retrieval system using combined color and shape indexing where query is cascaded by color and then shape has been developed [36]. This combines the earlier grid-based shape representation with the dominant color-based index to provide better retrieval efficiency and effectiveness.

Spatial location of objects or regions as well as inter relationships between regions within an image provide vital clues for searching image databases. There have been a few spatial query methods in literature including [37–39]. These methods mostly work on the concept of representing image regions using string representations and matching is based on 2D subsequence matching and exact matching. They also assume that the regions have been segmented out of the images reliably. But in most cases this is not a reasonable factor, since reliable segmentation of regions poses a problem. So most of the recent approaches like [40,41] advocate the method of dividing the image space into sub-blocks and use the features extracted out of each sub-block to index the whole image on those features. Since our application database contains a mixture of both natural objects and simulated and man-made objects we have used this approach for representation of location feature.

Recently, the ISO MPEG Group has initiated the "MPEG-7 Multimedia Description Language" work item. It defines standardized descriptions and description systems that allow users to search, identify, filter, and browse visual content [42]. Images are retrieved by defining objects, including color patches or textures and the color and shape of the object, object size, etc. The description schemes are used to retrieve images based on a combination of these descriptors. Color descriptors used are Scalable color descriptor (SCD—based on color histogram encoded by Harr transform), Dominant color descriptors (based on clustering color regions), Color layout descriptors, CSD, and GOF/GOP color descriptors. Shape descriptors supported in MPEG-7 are Shape spectrum, Region-based descriptor (moment invariant based), Contour-based shape descriptor (curvature scale space based), and 2D/3D shape descriptors. To allow for maximum flexibility for design of applications, MPEG-7 specifies only minor parts of the descriptor extraction methods and not how similarity between content should be measured [42].

In the context of MPEG-7 framework, we have employed the dominant color clustering approach for color description. Shape is described using a region-based shape descriptor invariant to translation, rotation, and scaling. It uses a grid-based approach for extracting region information of the shape regions.

In the current effort, we have implemented a system that indexes and retrieves similar images based on a composite color–shape–location index. The color segmentation is made more robust by applying a pre-processing step to smoothen out the homogeneous color regions using anisotropic diffusion to improve segmentation of color regions. The shape features used are invariant to translation, rotation, and scaling. The effectiveness of the algorithm has been illustrated with number of examples. Our database is limited to collection of 200 basic images of flags and 140 basic images of fruits, flowers, vegetables, and simulated objects. Most of them are then transformed by scaling and rotating to generate totally about 900 images in all. Our proposed method is targeted at applications which are amenable to environments where homogeneous regions can be segmented out of the images like in the case of Logo/Trademark images of companies, identifying and inspecting articles against uniform background, etc.

It is desirable for any system to be tested on a larger database of images with well established ground truth. In this direction, we have conducted studies on the performance of our indexing and retrieval method on the image database testbed used in the MPEG-7 core experiments [48]. The total number of images in the database is 1400: 70 classes of various shapes, each class with 20 images. We have used about 46 of these classes for testing our method for robustness to scaling, rotation, and noise.

The paper is organized as follows. In Section 2 we explain the segmentation of dominant regions within the image using color categorization and region selection method. Section 3 describes the shape representation using grid-based approach. Identifying the location of regions within the image space is dealt in Section 4. Indexing of images by content feature like color, shape, and location indices and combined indices is covered in Section 5. The hash data structure for storing the indices and the shape similarity measure used for matching is depicted in Section 6. Section 7 summarizes the experimental results and performance evaluation. Conclusions about our integrated image indexing and retrieval scheme are discussed in Section 8.

2. Color-based dominant region segmentation

Efficient database indexing involves extracting visual features off-line and storing them as metadata in the image database. The variety and dimension of visual features are very high and this poses the greatest challenge in building efficient index structures. The semantics of images is much richer or complex and thus requires more levels of interpretation. We are interested in understanding certain objects in the image and relationships between them.

We need to have the ability to look at image attributes to judge their relevance with respect to what the user has in mind. Also, search engines now organize indices based on general taxonomy. So issues like organization of data, techniques used to access data, etc. are quite important for construction of image databases and the retrieval aspect.

The strong motivation for using color to perform selection comes from the fact that it provides region information and that, when specified appropriately, it can be relatively insensitive to variations in normal illumination conditions and appearance/viewpoint of objects. It is also robust to scaling, orientation, perspective, translation, rotation, and occlusion of images. Dominant color region in an image can be represented as a connected fragment of homogeneous color pixels which is perceived by human vision [43]. Our technique to index images is based on this concept of dominant color regions present in the image. The segmented out dominant regions along with their features are used as an aid in the retrieval of similar images from the image database.

2.1. Feature extraction and color clustering

The two most common issues to be considered are the selection of a proper color space and a good color quantization scheme to reduce its dimension. Usually there is a transformation from the traditional RGB color space to other color spaces like HSV, YIQ, Lab, etc. But it has been shown [44] experimentally that for indexing purposes there is not much of an improvement by using any particular transformed color space.

Hence, our indexing approach employs color clustering that is carried out in RGB-space to extract up to three dominant regions in the images based on human perceptual colors [20]. It has been shown that the cardinality of images by color is four on the average [23]. We resort to a weak segmentation, since image retrieval does not entail solving the general image recognition problem. It is sufficient that a retrieval system present similar images in some user-defined sense.

2.2. Anisotropic diffusion

In order to smoothen out small variations in color and texture, while preserving the overall boundary of objects we propose to utilize anisotropic diffusion averaging process [45,46] which is sensitive to structures in the image. In image segmentation, a number of visual features can be used to specify the boundary of the segments. One such strategy is to use the local homogeneity of the local image structure where stable differential measurements are made via a variable-conductance diffusion process. This process is called anisotropic diffusion. It is shown to preserve important image structure while reducing unwanted noise. We propose to use it as a pre-processing step before the actual image segmentation by dominant color regions approach.

Dividing an image into meaningful parts serves a number of purposes. Primarily, it consolidates the vast array of information present in the visual field. It can help distinguish areas of importance from those of less importance, and it thereby provides means of focussing attention. Segmentation also gives a level of structure for subsequent analysis that reflects not only basic spatial relationships but the deeper structure of the image. Our main issue is to obtain a high degree of distinguishing boundaries of color regions within a given image before going in for the actual segmentation.

The main step involved in anisotropic diffusion as proposed by Perona–Malik [46] makes use of the gradient descent as shown below:

$$I_{\mathrm{s}}^{t+1} = I_{\mathrm{s}}^{t} + \frac{\lambda}{|\eta_{\mathrm{s}}|} \sum_{p \in |\eta_{\mathrm{s}}|} \psi\left(I_{\mathrm{p}} - I_{\mathrm{s}}^{t}, K\right), \tag{1}$$

where t denotes discrete time steps, K is the "scale" parameter, ψ is the robust error norm, I_s^t is the discretely sampled image, s denotes the pixel position in a discrete image with p being the neighborhood pixels belonging to η_s and λ is a scalar that determines the rate of diffusion. η_s represents the spatial neighborhood of the pixel s, and is the eight neighborhood of s, except at the boundary. Three error norms for use in anisotropic diffusion equation as the influence function have been proposed by Black et al. [45] for the edge-stopping function $\psi(r, K)$ and are shown below.

2.2.1. Lorentzian error norm

The edge-stopping function according to Lorentzian error norm is given by

$$\psi(r, K) = \frac{2r}{2 + (r^2/K^2)}. \tag{2}$$

Here, as the value of K increases, the diffusion also increases. Its effect on the size depends on the nature of the image.

2.2.2. Huber's minmax norm

According to the Huber's minmax norm, we get the following:

$$\begin{aligned} \psi(r, K) \quad &= r/K \quad &&\text{if } |r| \leqslant K \\ &= \operatorname{sign}(r) \quad &&\text{if } |r| > K. \end{aligned} \tag{3}$$

Here, as the value of K increases, the diffusion decreases as a higher value of K implies that the diffusion could be less for the same gradient.

2.2.3. Tukey's biweight norm

Tukey's biweight norm is given by equation below

$$\begin{aligned} \psi(r/K) \quad &= r[1 - (r/K)^2]^2 \quad &&\text{if } |r| \leqslant K \\ &= 0 &&\text{otherwise.} \end{aligned} \tag{4}$$

Here, also it is the same as in Lorentzian error norm. An increase in K results in an increase in the diffusion.

We have experimented the effect of these norms on the images in our database to find the most optimal one for our domain. We have tried to look into the visual effects produced by the error norms on the original images for obtaining the best diffusion using each of the error norms. It is seen that the Tukey's biweight is most suited since it produces sharper boundaries than diffusing with the other two norms. Fig. 1 depicts the sample flower image to be segmented. Applying Lorentzian error norm gives good diffusion but the edges are not sharp as seen in Fig. 2. Huber's minmax norm method as shown in Fig. 3 provides very poor results. It is observed that Tukey's biweight norm as seen in Fig. 4 gives the best diffused image for the sample image of all the three norm factors used.

We have applied anisotropic diffusion as a pre-processing step before segmentation to smoothen out the color regions in the images. Fig. 5 shows the effect of anisotropic diffusion on our segmentation procedure on two representative database images. It is seen clearly that the color regions are more smoothened and hence lead to better identification of the homogeneous color regions within the image.

Fig. 1. Input flower image.

Fig. 2. Lorentzian norm applied.

Fig. 3. Huber's minmax norm.

2.3. Color space categorization

Segmentation is very important for image retrieval since it greatly enhances the retrieval based on shape and layout features. To obtain objects at high-level, we need some human assistance in the segmentation process. Since our proposed method of indexing involves only regions that are dominant in a given image, we use the automatic segmentation approach. Also different applications require varying levels of accuracy in segmentation. For our intended domain, a weak segmentation suffices

Fig. 4. Tukey's biweight norm.

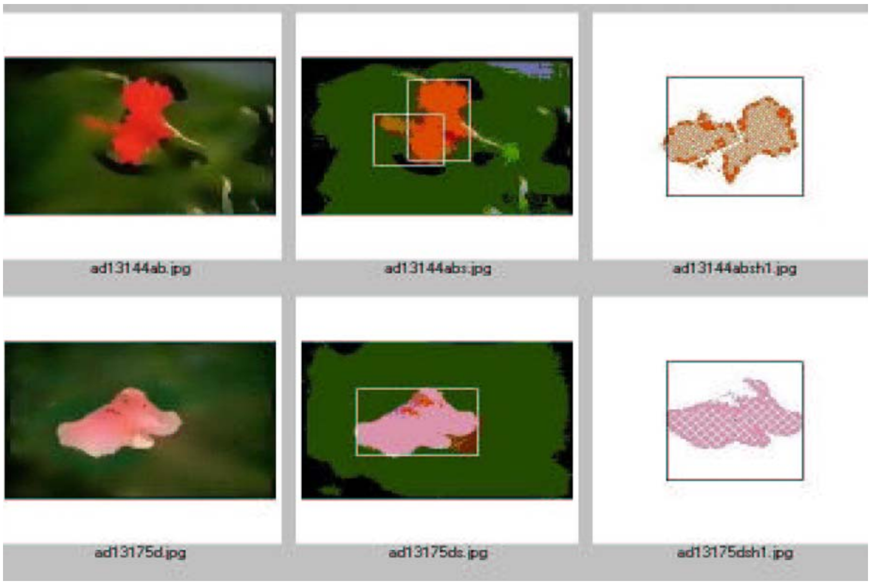


Fig. 5. Effect of anisotropic diffusion on segmentation.

to yield us with clearly segmented regions since we assume that the image is captured using a clear background and objects in the scene are non-occluding in nature.

Color quantization is applied when the color information of an image is to be reduced. In our work, we are interested only in similarity match rather than exact matching of color regions. Hence, it suffices that we use a reduced color space, which also serves as a pre-segmentation process. Also, the number of dominant colors in images are limited to a small number, usually four or five on the average. From the human recognition point of view also, the number of colors any user can recollect is limited. Another reason for color reduction is that the storage requirements for the index structure has to be traded for amount of colors used to represent colors in the image. A spatial reduction in color space makes it possible for the perceived color image to look as similar to the perceived quantized image as possible. The assignment of pixels to colors and the grouping of optimal colors in the image also facilitates this. The main objective is to make nearby colors map into a common bin so that they are all treated as perceptually similar for color matching and retrieval. The loss in detail of the color information will not affect the matching process drastically. The number of colors may be increased if finer matching is required depending on the application domain.

Many approaches exist for color quantization that include vector quantization, clustering [21,47] and neural networks [9]. The entire RGB color space can be represented using a smaller set of color categories that are perceptual to humans. A possible theoretical justification for this is provided by Corridoni et al. [47], who present a system supporting image retrieval by high-level chromatic contents, which are meant as sensations that color accordances generate on the observer. Images are archived by describing the spatial arrangement of regions with homogeneous chromatic attributes. They choose a color space which replaces the original 3D color space by a discrete set of representative colors. A final set of 180 reference colors is used in their work.

Cluster-based approach has the advantage that, if we apply it to all or at least some representative images in the database, the clustering process will take into account the color distribution of images over the entire database. This minimizes the likelihood of color bins in which none or very few pixels fall, thus resulting in a very efficient color quantization. A color-look-up table can be used to store the reduced space and then colors of original images may be mapped into this reduced color space thereby speeding up the overall process. Such an approach has been followed by Kankanahalli et al. [21], wherein they use a set of 27 discrete colors to map the entire color space, such that all the colors in the application are covered perceptually. An unsupervised learning algorithm based on clustering is used to deduce from the representative sample of images the table of colors.

Syeda-Mahmood [43] proposes a color region-based retrieval system in which perceptual color categories are used. A color-look-up table using 220 different colors is made use of to coarsely describe the colors of regions and later used to index the images. This perceived color of a region specifies the color category corresponding to the dominant color of the region.

According to the MPEG-7 visual feature descriptor language, users can perform tasks on images to define objects, including color patches or textures, and get, in return, example images from which one can select regions of interest. Colors in a region are clustered into small number of representative colors and along with their percentages, spatial coherency, and variance form the descriptor [42].

Following [21,43,47], and as per the recent MPEG-7 standardization directives for visual descriptors [42], we too use a color quantization in RGB space using 25 perceptual color categories to segment images based on dominant colors. This is justified since these 25 colors are sufficient to clearly distinguish all the colored regions in our chosen domain of image database. From the segmented image we find the enclosing minimum bounding rectangle (MBR) of the region, its location, number of regions in the image, etc., and all these are stored in a metafile for further use in the construction of an image index using a hash data structure.

The selection of the number of colors in the reduced space is a trade off between performance and speed for a particular application. For efficient indexing, a small number of cells is advisable which also reduces the computation load. A smaller set is more useful since it gives a coarser description of the color of a region to accommodate slight variations in imaging conditions. We have taken a table of 25 perceptual colors chosen from the standard RGB color palette as summarized in Table 1. Color space is divided uniformly and to compute similarity between two regions we use the standard euclidean distance.

2.4. Color matching and region selection

The method depends on determination of boundaries where perceptual color changes occur before any cluster in color space may be interpreted as corresponding to a region in image space. The RGB color space is partitioned into sub-spaces called color categories. The perceptual color of a pixel can be specified by the color category into which it maps. A flowchart of segmentation and the dominant region selection process is shown in Fig. 6.

Table 1
Color-look-up table

Color	R	G	B
Black	0	0	0
Sea green	0	182	0
Light green	0	255	170
Olive green	36	73	0
Aqua	36	146	170
Bright green	36	255	0
Blue	73	36	170
Green	73	146	0
Turquoise	73	219	170
Brown	109	36	0
Blue gray	109	109	170
Lime	109	219	0
Lavender	146	0	170
Plum	146	109	0
Teal	146	182	170
Dark red	182	0	0
Magenta	182	73	170
Yellow green	182	182	0
Flouro green	182	255	170
Red	219	73	0
Rose	219	146	170
Yellow	219	255	0
Pink	255	36	170
Orange	255	146	0
White	255	255	255

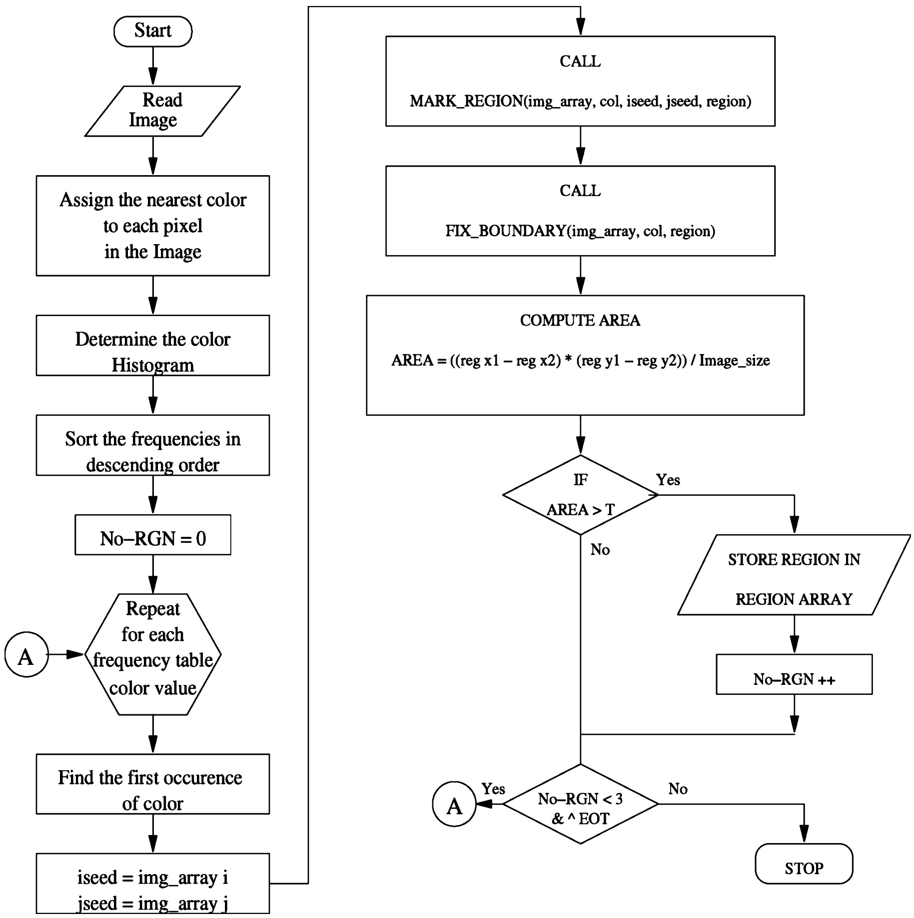


Fig. 6. Segmentation and region selection process flowchart.

The procedure to segment the image into regions according to their perceived color involves mapping all pixels to their categories in color space, and grouping pixels belonging to same category. A color will be selected from 25 pre-defined colors which is very near to image pixel color and it will be stored as new color pixel in the image. Using p_r, p_g, p_b as red, green, and blue components of intensity values of the pixel and C_{i_R}, C_{i_G}, C_{i_B} the corresponding values of the color entry in the color table, color distance C_d is calculated using Euclidean distance formula, as specified below

$$C_d = \min\left(\sqrt{(p_r - C_{i_R})^2 + (p_g - C_{i_G})^2 + (p_b - C_{i_B})^2}\right), \quad i = 1, \ldots, 25. \tag{5}$$

Region marking is done on the updated image. A bounding rectangle is drawn for each dominant region selected. The area of boundary rectangle is used in determining normalized area of dominant region. Number of regions present, each regions' information like color, normalized area, etc. are stored in a meta-file for further processing. This file information is used for constructing image index tree. When the search engine is initiated, the index tree is constructed.

Steps involved in segmentation and boundary detection

(1) Read in the image and create an image array that contains the RGB components of each pixel in the image.
(2) For each pixel in the image do
   (a) Search color-look-up table for the nearest color by finding the distance between the pixel color I represented as (p_r, p_g, p_b) and the color in the color-look-up table C_i represented as (C_{i_R}, C_{i_G}, C_{i_B}) using the distance formula C_d given in Eq. (5).
   (b) Assign to the pixel the RGB entry in color-look-up table for which C_d is minimum.
(3) Create a frequency table for each assigned color.
(4) Sort the frequency table in descending order to give the dominant color regions.
(5) Repeat steps 6–10 till three dominant regions are found or frequency table is exhausted.
(6) Going in scan order fashion, determine the first occurrence of a pixel that has the same RGB value as in the sorted frequency table.
(7) Assign the pixel location to horizontal and vertical seeds viz: *iseed*, *jseed*.
(8) Following *iseed* and *jseed*, mark the entire region using 8-connected neighboring region growing method.
(9) Obtain (x, y) coordinates of boundary of marked region R, and draw the bounding rectangle.
(10) Determine normalized size $s(R)$ of bounding rectangle using: $s(R) = (|x_1 - x_2| * |y_1 - y_2|)/N$ where x_1, y_1 and x_2, y_2 are coordinates of the opposite vertices of the bounding rectangle and N is the size of the image.

The illustration of segmentation and boundary detection are shown in Figs. 7 and 8, respectively.

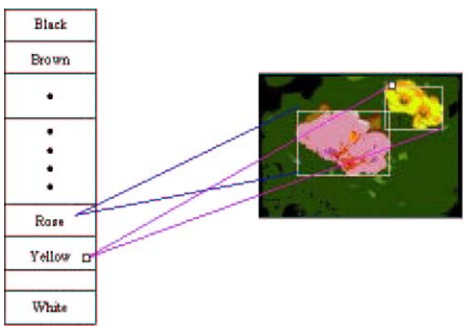


Fig. 7. Illustration of assign-color.

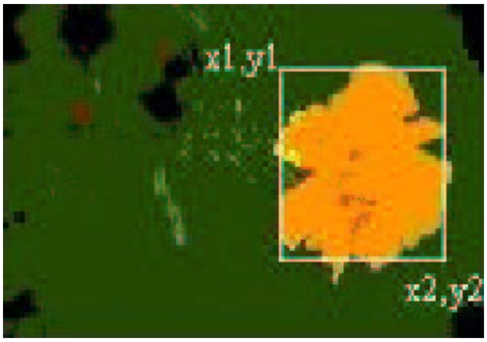


Fig. 8. Image with boundaries marked.

3. Shape representation

Another important feature used for content representation and CBIR is the shape of objects or regions of interest. In contrast with low-level color and texture features, shape features require good segmentation to detect objects or regions. Basically, shape characterization is of two types: boundary-based and region-based. Boundary-based shape features include rectilinear shapes, polygonal approximation, finite element model, and Fourier-based shape descriptors. Region-based features include statistical moments and grid-based approach. Hence methods employing shape features are suited for applications where objects or regions are readily available. Also shape similarity poses a crucial problem since it needs a representation that is invariant to translation, rotation, and scaling. This is required since human visual perception tends to be robust to such variations in object recognition. In our work, dominant regions are segmented out from the images based on color features. These act as regions of interest for the shape representation. We adopt a region-based approach similar to that in [34] to represent shape using a grid-based approach that is invariant to translation, rotation, and scaling. In [35] we have proposed and implemented a shape indexed based image retrieval system that uses a similar shape representation but different similarity measure and indexing. The retrieval results are also shown to be better than [34].

3.1. Terminology

The terminology used for representing shape features using a grid-based approach [34] are enumerated below:

- *Major axis.* It is the straight line segment joining the two points on the boundary farthest away from each other (in case of more than one, select any one).
- *Minor axis.* It is perpendicular to the major axis and of such length that a rectangle with sides parallel to major and minor axes that just encloses the boundary can be formed using the lengths of the major and minor axes.

- *Basic rectangle.* The above rectangle formed with major and minor axes as its two sides is called basic rectangle.
- *Eccentricity.* The ratio of the major to the minor axis is called eccentricity of the region.
- *Centroid or center of gravity.* A single point of an object/region towards which other objects/regions are gravitationally attracted. For 2D shapes, the coordinates (X_c, Y_c) of the centroid are given by

$$X_c = \sum_x \sum_y f(x,y)x \Big/ \sum_x \sum_y f(x,y),$$

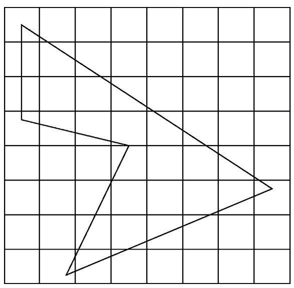
$$Y_c = \sum_x \sum_y f(x,y)y \Big/ \sum_x \sum_y f(x,y),$$

where (x,y) are pixel coordinates and $f(x,y)$ is set to 1 for points within the boundary and set to 0 elsewhere.

3.2. Basic scheme

Given a shape region, a grid consisting of fixed-size square cells is placed over it, so as to cover the entire shape region as shown in Fig. 9. We assign a "1" to a cell with at least 25% pixel coverage by the region and "0" to other cells. A scan order traversal results in a binary sequence of 1s and 0s as representation of the shape feature. For example, the shape in Fig. 9 can be represented by a binary sequence 00000000 11000000 11110000 01111000 00011110 00011110 00111000 00100000.

Smaller the grid size, more accurate is the shape representation but at the cost of more storage and computation requirements. The representation is compact, easy to obtain, and translation invariant. Hence, a scale and rotation normalization is carried out to make it invariant to scale and rotation.



0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0
0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0
0	0	0	1	1	1	1	0
0	0	1	1	1	0	0	0
0	0	1	0	0	0	0	0

Fig. 9. Original shape region.

3.3. Rotation normalization

The purpose of rotation normalization is to place shape regions along a unique standard orientation. This is done by rotating the shape region such that its major axis is parallel to the x-axis.

There are still two possibilities as shown in Figs. 10 and 11, caused by 180° rotation.

Further, two more orientations are possible due to the horizontal and vertical flips of the original region as shown in Figs. 12 and 13, respectively. Two binary sequences are needed for representing these two orientations. But only one sequence needs to be stored and at the time of retrieval we can suitably take care for these two sequences.

3.4. Scale normalization

To achieve scale normalization, we proportionally scale all the shape regions so that their major axes have the same length of 96 pixels. This provides us with size invariance of the color regions in the image.

A sample output of the normalization process of the shape regions is shown in Fig. 14. The input image shown on the left of Fig. 14 has four color regions, of which the first three dominant ones are segmented using the dominant color region

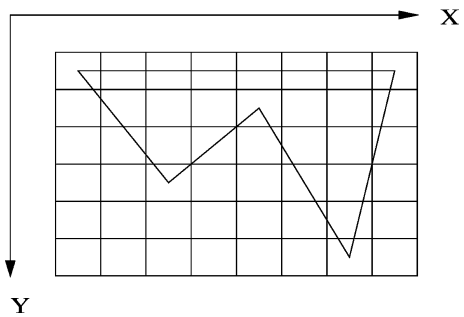


Fig. 10. Major axis aligned to x-axis.

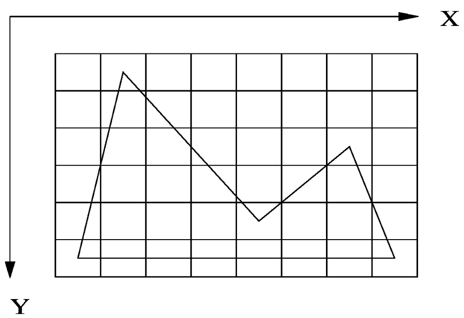


Fig. 11. Aligned region rotated by 180°.

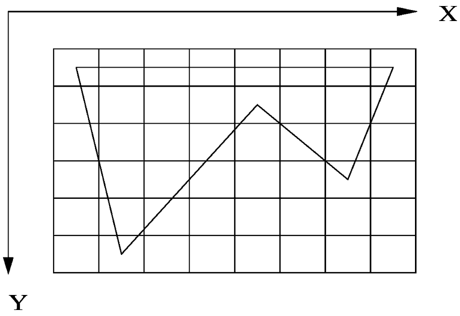


Fig. 12. Horizontal flip of aligned region.

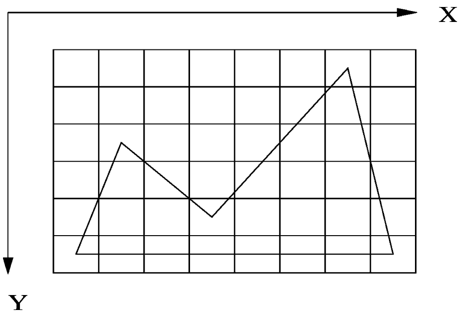


Fig. 13. Vertical flip of aligned region.

Fig. 14. Normalized shape regions.

segmentation procedure (central part of Fig. 14). Then shape normalization procedure is applied on the individual regions to make them invariant to translation, rotation and scaling. All shape regions are placed in fixed sized 96×96 pixel grids (right side of Fig. 14) in the order of region dominancies. The GUI also depicts the reference colors assigned to each region from the color table as well as the percentage of each color region in the image.

4. Finding location of the region

Our approach is based on the sub-block technique where the image space is divided into nine sub-locations as shown in Fig. 15. The approximate position of the region is determined with respect to the relative location of the region within the image. The classification is according to a location map containing nine regions of the image space.

A program segment highlighting the find location procedure is given following this paragraph. We have considered image size of 192×128 pixels in our work. The four intersecting corners of the nine sub-locations depending on image size are initialized at the beginning of the program.

```
// program segment to find location of a region
public void assignPosition( ... ) {
  int x1 = 64, y1 = 43, x2 = 128, y2 = 43, x3 = 64, y3 = 86, x4 = 128,
  y4 = 86;
  int pos = 0;
  if((r.x1 < x1) && (r.x2 < x1)) {
    // Fully left
    if ((r.y1 < y1) && (r.y2 < y1)) {
     // Top Left
     pos = 1;
    }
    else if((r.y1 < y1) && (r.y2 > y1)) {
     // Left - bet middle and top
     if(Math.abs(y1-r.y1) >= Math.abs(y1-r.y2)) pos = 1;
     else pos = 4;
    }
    else if((r.y1>=y1) && (r.y2 < y3)) {
     // Center Left
     pos = 4;
    }
```

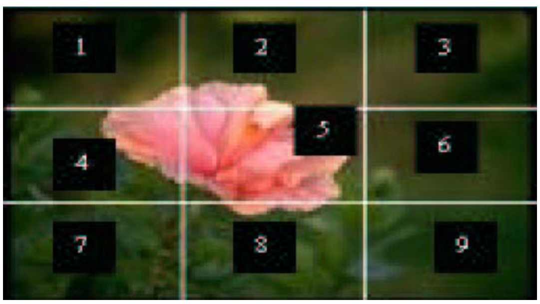


Fig. 15. Illustration of find location.

```
      else if((r.y1 < y3) && (r.y2 > y3)) {
       // Left - bet middle and bottom
       if(Math.abs(y3-r.y1) >= Math.abs(y3-r.y2)) pos = 4;
       else pos = 7;
      }
      else if (r.y1>=y3) {
       // Bottom Left
       pos = 7;
      }
    } //end of fully left
    else if(r.x1 >= x2) {
      // Fully right
      ...
      ...
    } // end of Fully right
    else if((r.x1 < x1) && (r.x2 > x1)) {
      // Between left and center
      ...
      ...
    } // end of Between left and center
    else if((r.x1 < x2) && (r.x2 > x2)) {
      // Between right and center
      ...
      ...
    } // end of Between right and center
  } //end of assignPosition procedure
```

The extracted dominant region features viz., color, area, and location are stored as a feature vector in a sequential file. Image database is constructed by processing all images off-line as this saves query processing time. A hash data structure is made use of to create an index structure to store the index. Each entry of the hash table points to similar images having unique indices. When a query is made based on an example image, only the example image is processed for index value and matching is based on the extracted region features of the query image with that of the images in the database.

5. Indexing by image content

5.1. Color indexing

The color space is grouped into 25 perceptual color categories indexed as $1, 2, 3, \ldots, 25$. The image is segmented and n dominant regions are selected. The overall color index CI is formed as shown below

$$\mathrm{CI} = \sum_{i=1}^{n} 25^{n-i} C_i, \tag{6}$$

where C_i is the index of the ith region. Images with similar CIs are stored in the same hash entry of hash index structure. Along with it, other metadata about the region are also stored. For our study we have considered only the first three dominant regions.

5.2. Shape indexing

Initially the segmented homogeneous color regions are considered. The major and minor axes are extracted and the object is rotated such that the major axis is parallel to the x-axis. The object shape is normalized to have a constant scale factor by making the major axis of fixed length of 96 pixels to achieve scale normalization. The object is centered in the image plane with its center of gravity at the center of the image plane. A fixed grid of size 96×96 pixels is placed over the object. It is divided into 64 cells, each of size 12×12. Hence each shape region is described completely by 8×8 cells. The images are categorized into different groups based on the vertical coverage of the shape regions. This is obtained by rounding off the value of the ratio of the normalized minor axis length to the cell size (12 in our case). The shape categories S_i range from 1 to 8 based on this computation. The shape index SI for n dominant regions is then formed as follows:

$$\mathrm{SI} = \sum_{i=1}^{n} 8^{n-i} S_i, \tag{7}$$

where $S_1, S_2, S_3, \ldots$ are the shape indices of the dominant regions in decreasing order of size (S_1 is the most dominant region).

A detailed procedure for shape indexing follows:

(1) Compute the major and minor axes of each dominant color region.
(2) Rotate the shape region to align the major axis with the x-axis to achieve rotation normalization and scale it such that major axis is of standard fixed length (96 pixels).
(3) Place a 8×8 grid over the normalized color region.
(4) Each cell C_{ij} $(i, j = 1, 2, \ldots, 8)$ is assigned "1" if the region covers at least 25% of cell area, otherwise it is assigned "0."
(5) Compute the row coverage total (RC) and column coverage total (CC):

$$\mathrm{RC}_i = \sum_{j=1}^{8} C_{ij},$$

$$\mathrm{CC}_j = \sum_{i=1}^{8} C_{ij}.$$

(6) Generate a binary sequence by traversing the image in scan-order fashion.
(7) Compute shape index for the region: $\mathrm{SI} = \text{round}$ (normalized minor axis length/12.)

5.3. Location indexing

Next we propose to use region-wise location information so as to enable image retrieval based on spatial location of regions. It also allows for matching images based on relative location positions with respect to regions within an image. To compute location index, we divide the image space into 3×3 grid cells and number them 1–9. Depending on the cell that is maximally covered by the region under consideration, it is allocated an index value equal to the cell number. This is then combined with the color and shape indices to provide an integrated color–shape–location index allowing for a non-cascaded matching on these combined features. The composite indexing method is simple, computationally less intensive, and shown to provide better matching results.

For location indexing, the image space is divided into 3×3 grid numbered 1–9. The region is likely to overlap number of cells in the image space. The index assigned is the cell number that is maximally covered by the region. If a region overlaps more than one grid cell, then the central area of the region is used to determine the location index.

It is often desired to retrieve images based on regions present at particular spatial locations within the image space. Hence, we determine the grid cell that the extracted image region maximally covers in the image space and assign the corresponding cell number as its location index (Fig. 15). The overall location index LI is formed as follows:

$$\mathrm{LI} = \sum_{i=1}^{n} 9^{n-i} L_i, \tag{8}$$

where $L_1, L_2, L_3, \ldots$ are the location indices of the dominant regions in decreasing order of size.

5.4. Composite indexing

5.4.1. Color-Shape index

Cascaded retrieval of images, initially indexing by color and then by shape results in some of the images not being retrieved from the database. So it is proposed to form a unique index for image regions based on composite color and shape. The composite index CSI is formed as follows:

$$\mathrm{CSI} = \sum_{i=1}^{n} [(25^{n-i} C_i) * 2^{10} + (8^{n-i} S_i)], \tag{9}$$

where $C_1, C_2, C_3, \ldots$ and $S_1, S_2, S_3, \ldots$ are the color and shape indices, respectively, of the dominant regions in the decreasing order of size.

The indexing method considers the eccentricity of the region and its homogeneous color category to form the combined color–shape index for representation of the image regions. Images are matched for shape similarity using the row and column coverage as the discriminating factor within some threshold [36].

5.4.2. Color–shape–location index

In order to give importance to location of specific regions, an integrated color–shape–location index is proposed. The integrated index CSLI is formed as follows:

$$\mathrm{CSLI} = \sum_{i=1}^{n} [(25^{n-i} C_i) * 2^{20} + (8^{n-i} S_i) * 2^{10} + (9^{n-i} L_i)], \tag{10}$$

where $C_1, C_2, C_3, \ldots, S_1, S_2, S_3, \ldots$, and $L_1, L_2, L_3, \ldots$ are the color, shape, and location indices, respectively, of the dominant regions in the decreasing order of size.

The indexing method considers the eccentricity of the region, its homogeneous color category and the location index of the image region to form the combined color–shape–location index for representation of the image regions. Images are matched for shape similarity using the row and column coverage as the discriminating factor within some threshold, as discussed in the following section.

6. Storage and similarity matching criteria

6.1. Hash structure

An image structure based on hashing technique is used to store the combined index of the image region-wise. At query time, only those images that are in the same hash bucket as those of the queried image are compared for similarity, thus reducing the search space and time.

A hash structure is used to store the keys of all images belonging to the same group in separate buckets. An instance of hash table is shown in Fig. 16. Along

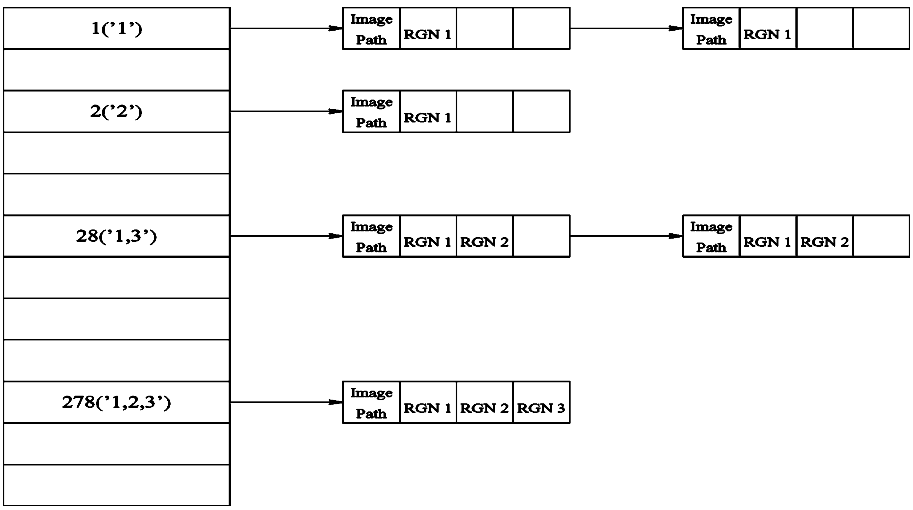


Fig. 16. Instance of Hash table.

with the composite image index, feature data like number of dominant regions, color information, color index, area, location index, shape string, shape index, column coverage total, and row coverage total of shape representation of individual regions as well as for regions taken pair-wise are stored as a vector for each image entry in the hash table. Each of these constitutes a feature vector based on the combined features. The mapping of region information to the hash index for a typical image is shown in Fig. 17 for all combinations of regions within the image.The CSL index for a typical region is shown below

$$[\mathrm{CSL}]\mathrm{Index} = (16, 8, 1) = 16 * 2^{20} + 8 * 2^{10} + 1 = 16785409,$$

which is the color–shape–location index for the top left region having a CI = '16,' SI = '8,' and LI = '1.' All image regions having a similar composite index will be mapped to the hash bucket with value '16785409.' Similar analysis for images with multiple regions is done to obtain unique indices for all combinations considering up to three dominant regions in a given image.

6.1.1. Similarity measure for shape matching

To find the best match between the query image and the images indexed to same hash location, we make use of row and column coverage of individual regions. The similarity measure is computed as follows:

(1) Calculate the row coverage RC_j and column coverage CC_j of the regions in the query image by counting number of cells along each direction.
(2) Calculate the row coverage RC'_j and column coverage CC'_j of the regions in the image extracted from database by counting number of cells along each direction.
(3) Find the row and column difference between query image regions and regions in the image selected from database using:

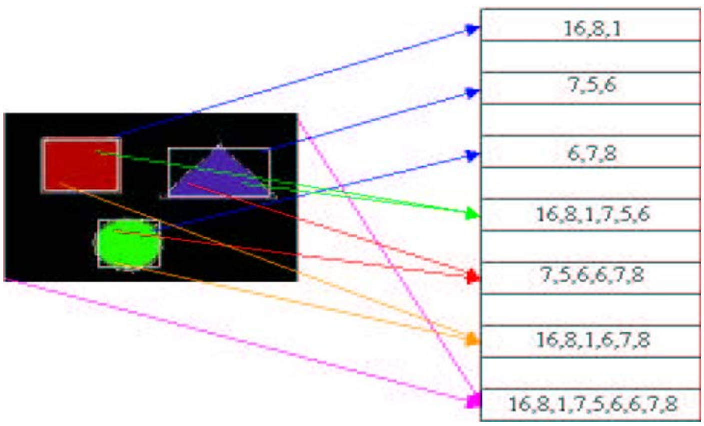


Fig. 17. Hash index structure for composite color–shape–location index.

$$R_{\mathrm{d}} = \sum_{j=1}^{8} |\mathrm{RC}_j - \mathrm{RC}'_j|,$$

$$C_{\mathrm{d}} = \sum_{j=1}^{8} |\mathrm{CC}_j - \mathrm{CC}'_j|.$$

(4) If $(R_{\mathrm{d}} + C_{\mathrm{d}}) < T$ (a suitable threshold), then the images match.

7. Experimental results and performance

7.1. Retrieval by individual color and shape indices

The experimental database consists of 200 basic images of flags and 140 basic images of fruits, flowers, and simulated objects. About 900 images were generated in all by carrying out rotation and scaling transformation on the basic images. Each image in the database is indexed on color and shape. A hash table stores images of similar index based on the features extracted. Each of the 200 basic flag images and 140 basic images of flowers, fruits, vegetables, and simulated shapes were used as a query image. The retrieved images are displayed using a JAVA-based GUI employing query-by-examples. Images are retrieved first based on the color index and displayed.

Relevant judgement on each query was made by three independent users. An example output for retrieval by color from an image database of flowers, fruits and vegetables (top row) and that from a database of flag images (bottom row) is shown

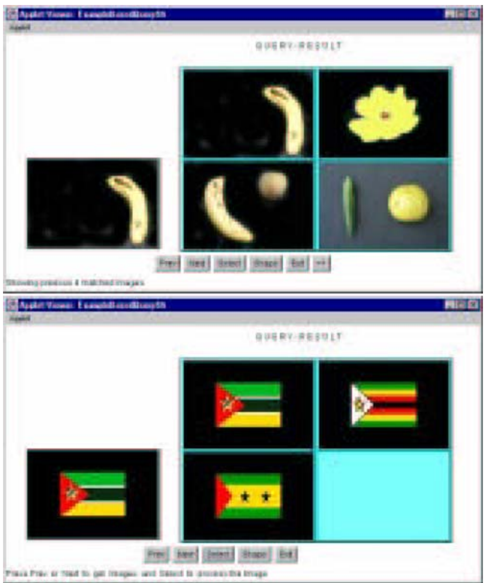


Fig. 18. Matching on color regions.

in Fig. 18, where the image on the left of each box is the query image followed by the 4 top matching images in the database. The threshold T for computing similarity measure was experimented with values of 2, 4, and 6 and later fixed at 4.

The sample results for retrieval by cascaded color and shape is shown in Fig. 19 for the respective database of images. It can be observed that images non-similar in shape are eliminated.

The retrieval performance is measured using recall and precision. Recall measures the ability of retrieving all relevant or similar items in the database. It is defined as the ratio between the number of relevant or perceptually similar items retrieved and the total relevant items in the database. Precision measures the retrieval accuracy and is defined as the ratio between the number of relevant or perceptually similar items and the total number of items retrieved.

Table 2 shows the retrieval performance in terms of precision and recall rates for the image database of flowers, fruits, and simulated shapes in terms of color only and cascaded color–shape indexing respectively. Similar analysis is done for the flag image database and shown in Table 3.

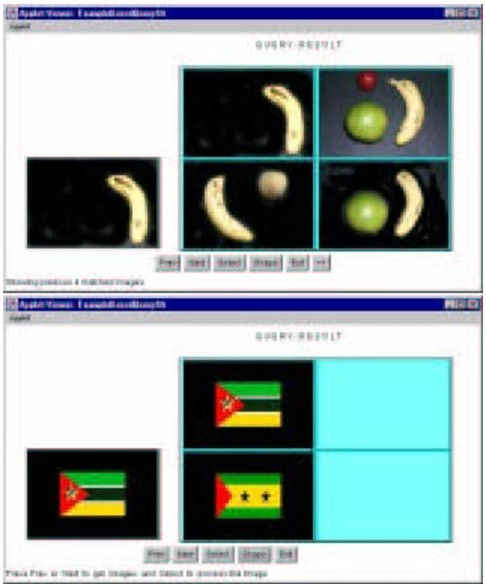


Fig. 19. Matching on shape regions.

Table 2
Retrieval results of precision and recall rates for image database of flowers, etc

	Based on color		Cascaded color and shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.708851	0.282746	0.845806	0.381269
Standard deviation	0.200282	0.197172	0.224701	0.270009

Table 3
Retrieval results of precision and recall rates for image database of flags

	Based on color		Cascaded color and shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.759438	0.31985	0.861375	0.248775
Standard deviation	0.229162	0.196683	0.225948	0.279476

We have compared the results of our technique with that proposed by Lu and Sajjanhar [34]. They use the similar grid-based approach but a different index and similarity measure for retrieval of shape regions. In their method the index is the simple number of 1s which represent the coverage of the shape region on the normalized grid. This leads to many dissimilar images being clubbed to same index (not desirable).

The search results for two typical queries using method in [34] is shown in Fig. 20 for matching by color and shape index. Fig. 21 shows the corresponding retrieval results by our approach. It is observed that there is better pruning of the matched images using our proposed indexing method along with the row and column coverage-based similarity measure technique for matching images based on cascaded color and shape.

We have compared our method with that of Lu and Sajjanhar [34] and the relative values of precision and recall measure are tabulated in Tables 4 and 5, respectively. It is observed that our indexing method and similarity measure provides better retrieval effectiveness.

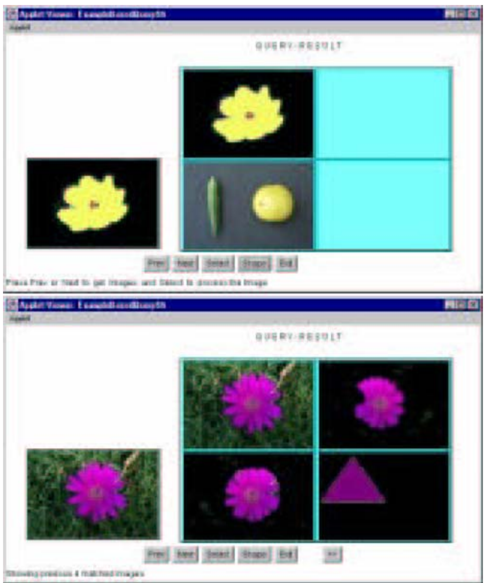


Fig. 20. Retreival results based on similarity measure of [34].

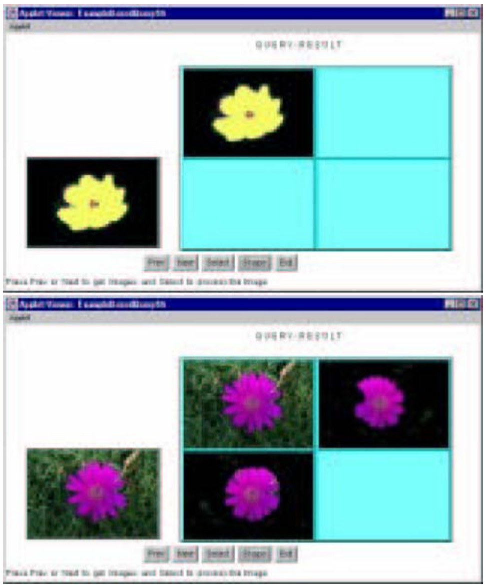


Fig. 21. Retrieval results based on our proposed similarity measure.

Table 4
Retrieval results of precision and recall rates for image database of flowers, etc

	Based on method of [34]		Our approach by cascaded color–shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.719643	0.302143	0.845806	0.381269
Standard deviation	0.325991	0.184953	0.224701	0.270009

Table 5
Retrieval results of precision and recall rates for image database of flags

	Based on method of [34]		Our approach by cascaded color–shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.710217	0.185435	0.861375	0.248775
Standard deviation	0.236695	0.148033	0.225948	0.279476

7.2. Region-wise querying by combined index

The same database as earlier is chosen to illustrate our combined indexing concept. Each image in the database is indexed on color, shape, and location features region-wise. The hash table is made to store images of similar indices based on the three features individually as well as for various combinations of these indices.

The system was made to respond to query images based on color, shape, location, composite color–shape, and composite color–shape–location. The retrieved images were displayed using a JAVA-based GUI using query-by-examples. Regions in the

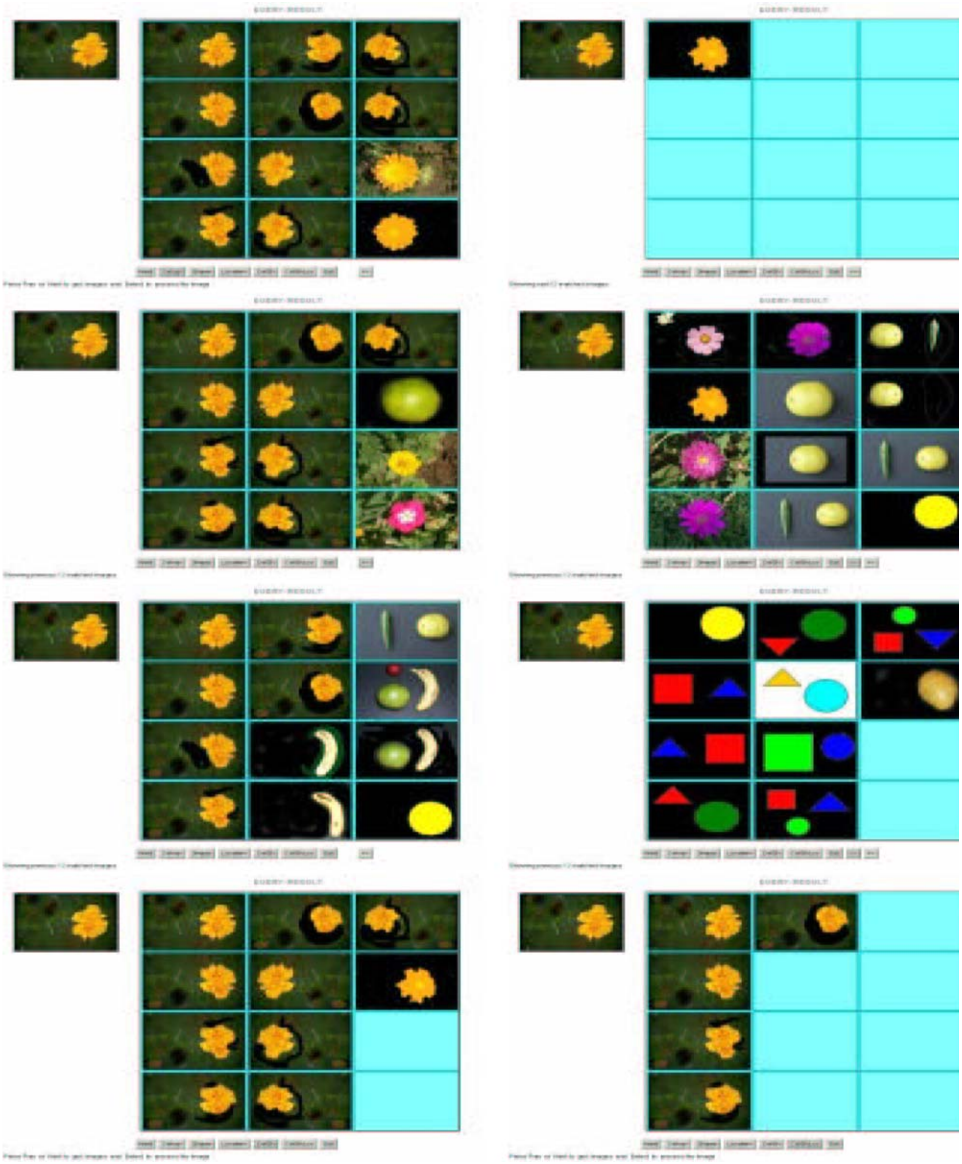


Fig. 22. Retrieval results on database of flowers, fruits, vegetables, and simulated figures based on: (A) color (first row), (B) shape (second row), (C) location (third row), (D) color–shape index (fourth row, left), and (E) color–shape–location index (fourth row, right).

query image are compared to those in the database image and only images containing a similar region pertaining to color, shape, location or their combination are retrieved based on the particular index used and suitable matching criteria based on

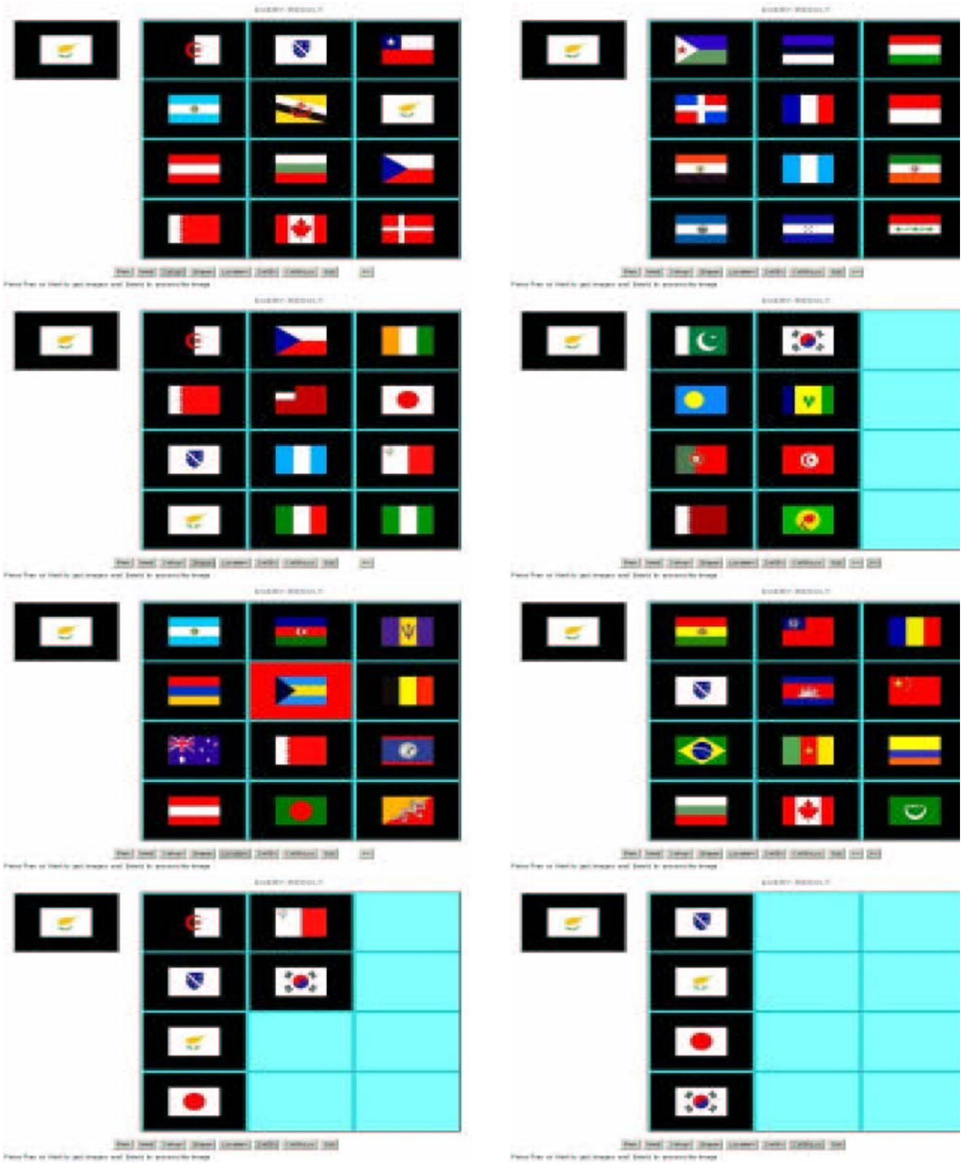


Fig. 23. Retrieval results on database of flags based on: (A) color (first row), (B) shape (second row), (C) location (third row), (D) color–shape index (fourth row, left), and (E) color–shape–location index (fourth row, right).

some similarity measures at certain pre-determined threshold (4 in our case). Example outputs are captured and illustrated. Fig. 22 depicts results for the database of fruits, flowers, vegetables, and simulated shape regions. It is seen that retrieval just by color index yielded 13 images, while 24 images were retrieved based just on shape

Fig. 24. Retrieval results on combined database based on: (A) color (first row), (B) shape (second row), (C) location (third row), (D) color–shape index (fourth row, left), and (E) color–shape–location index (fourth row, right).

matching. Going by location index, 22 images were retrieved (third row in Fig. 22). The composite color–shape index retrieved 10 matches while the composite color–shape–location index retrieved almost exact matches of the input query (right image of fourth row).

Fig. 25. Retrieval results on combined database based on: (A) color (first row), (B) shape (second row), (C) location (third row), (D) color–shape index (fourth row, left), and (E) color–shape–location index (fourth row, right).

A similar exercise was done for the flag image database and results are shown in Fig. 23 where the composite color–shape–location index leads to exact matching images from the database. It is observed that combined indexing results in better pruning of images. The last row in Figs. 22 and 23 shows that output using combined indexing produces correct retrieval of images compared to retrieving images on individual features.

We have also experimented with our system by applying it on a combined database of images comprising of the earlier two datasets. The results are promising and effective as depicted in Figs. 24 and 25.

To measure performance of our proposed indexing methods, we have tabulated the average precision and recall rates of retrieval results from different image databases in Tables 6 and 7. It is found that the combined region-based color–shape–location indexing approach provides better precision and recall rates compared to methods involving only color and shape indexing. Also the non-cascaded indexing approach is shown to perform better compared to the earlier cascaded approach of retrieving first on color and then on shape. In both the image database sets it has given better performance characteristics.

Table 6
Retrieval results of precision and recall rates for image database of flowers, etc

	Based on color		Based on shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.743401	0.528	0.58338	0.393825
Standard deviation	0.26772	0.218	0.205423	0.187294
	Based on color–shape		Based on color–shape–location	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.932563	0.355601	0.788688	0.593624
Standard deviation	0.176627	0.257311	0.279232	0.33318

Table 7
Retrieval results of precision and recall rates for image database of flags

	Based on color		Based on shape	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.932798	0.1587	0.478209	0.545064
Standard deviation	0.121553	0.0796	0.241892	0.250614
	Based on color–shape		Based on color–shape–location	
	Precision rate	Recall rate	Precision rate	Recall rate
Average (mean)	0.482703	0.525617	0.719977	0.442038
Standard deviation	0.254327	0.329105	0.269078	0.261109

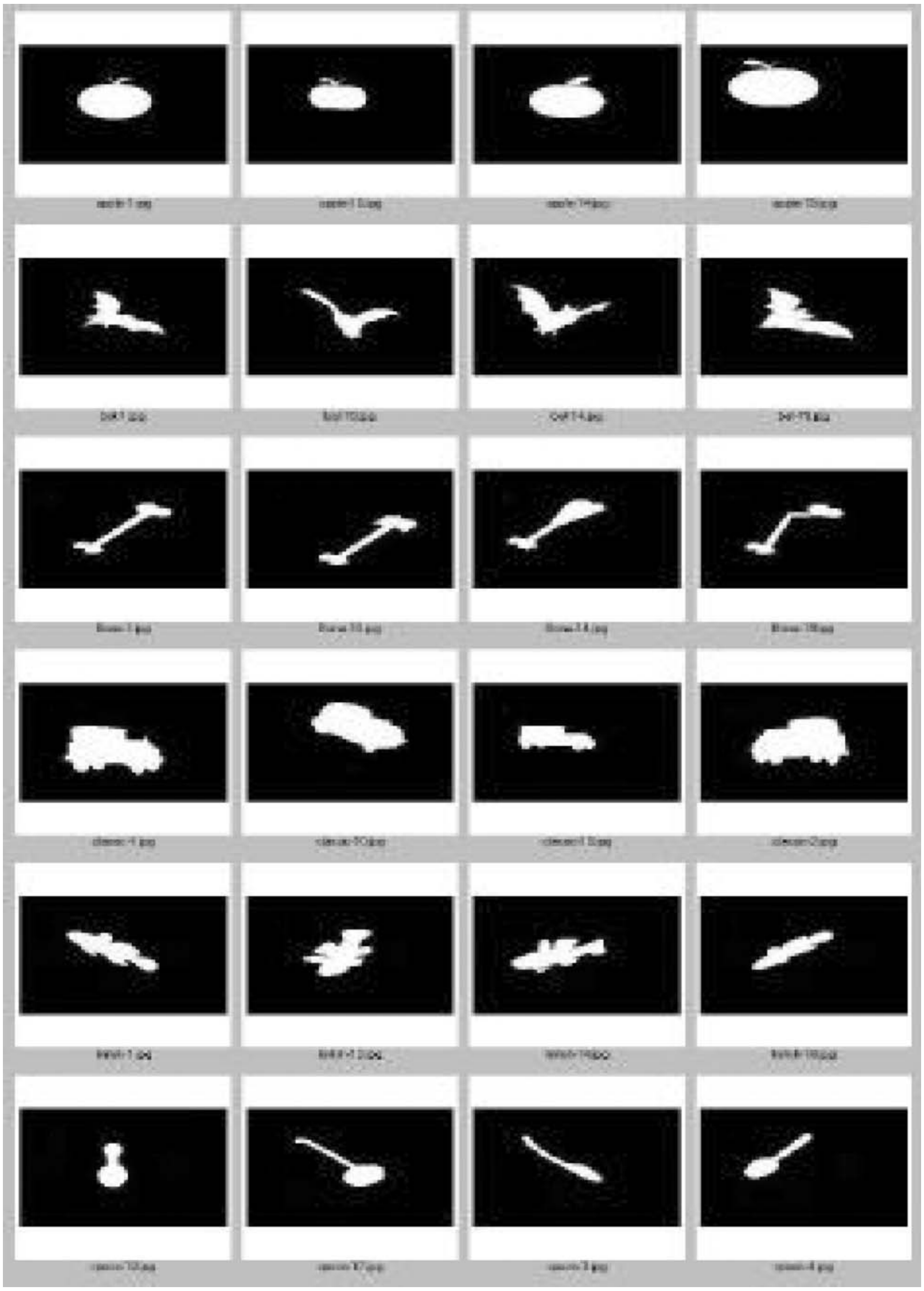


Fig. 26. Sample of shape images used in our experiments. Shapes in each row belong to same class.

Table 8
Retrieval results of percentage precision rate for top 20 retrievals

Classes	Number of retrievals									
	2	4	6	8	10	12	14	16	18	20
Apple	100	100	100	100	100	100	100	100	100	100
Bat	100	93.75	83.33	78.12	78	77.08	69.64	60.93	54.16	48.75
Bell	100	100	100	100	100	95.8	83.92	75	66.66	60
Bird	100	91.66	68.33	62.5	52.5	44.44	40.47	37.5	33.33	30
Bone	100	100	95.83	93.75	92.5	91.66	85.71	81.25	77.77	71.25
Bottle	100	75	58.33	50	47.5	41.66	39.28	39	38.88	38.75
Car	100	87.5	75	73.87	72.95	72.91	71.42	67.18	62.5	62.5
Carriage	75	62.5	58	50	40	38.33	38	37.5	37.5	37.5
Chicken	87.5	81.25	66.6	60	56.6	52.77	50	50	48	43
Lmfish	100	100	100	87.5	80	70.8	64	56	50	50
Crown	100	100	83.33	68.75	55	53.12	50	50	50	50
Device0	100	100	100	97.5	96	93.33	82.8	72.5	65.5	61
Device1	100	100	100	90	86	86	80	78	76.66	73.75
Device2	100	100	100	100	95	87.5	83.33	82.75	81.25	80
Device5	100	100	100	100	100	100	98.6	95	90	85
Average	97.5	92.77	85.92	80.8	76.6	73.69	69.14	65.46	62.11	59.43

7.3. Retrieval experiments on MPEG-7 shape database

Additional simulation studies have been carried out by experimenting with a sizeable database with an established ground truth using a well defined and well described process to establish the suitability of our approach.

Table 9
Retrieval results of percentage recall rate for top 40 retrievals

Classes	Recall rate
Apple	100
Bat	50
Bell	60
Bottle	40
Bone	71.25
Car	100
Carriage	72.5
Chicken	37.5
Device0	61.25
Device1	81.25
Device2	45
Device5	87.5
Device8	41.25
Key	36.25
Teddy	37.5
Crown	51.25
Average	57.86

We describe the testing method and present the experimental results of our proposed shape-based indexing and retrieval approach performed on the image database testbed used in the MPEG-7 core experiments [48], available at "http://give-lab.cs.uu.nl/mpeg7-ce-shape1b/." It provides data for 70 objects (classes), with each class containing number of shapes. Shapes in each class are similar (labeled by humans). This provides the ground truth.

For our experiment we have considered 46 classes from this database, with each class containing four shapes to give a total of 184 base images. These were then

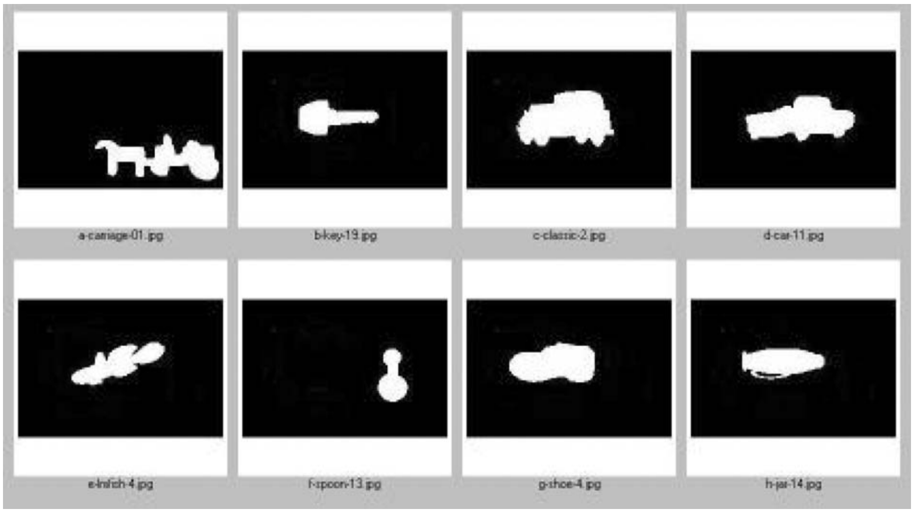


Fig. 27. Example shapes showing similar shaped objects from different classes. Shapes in first row belong to some class, but are more similar to images in second row in corresponding columns than they are to images from their own classes.

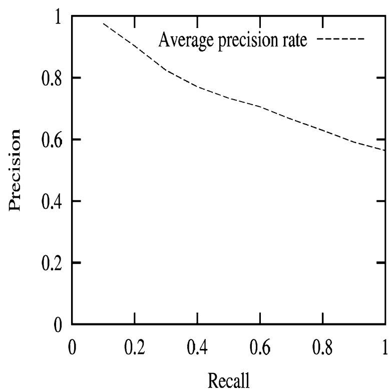


Fig. 28. Precision recall graph for top 40 retrievals.

subjected to three rotations (180° and horizontal and vertical flips of the same) and 1 scaling (0.5 factor) to result in total of (184×5) 920 images in all.

The database consists of binary images which are projections of 3D objects. Shapes are restricted to simple pre-segmented shapes which are defined by their outer contours. Some examples are shown in Fig. 26. The shape may change due to change in viewpoint of observer or camera, with respect to object. Also, noise due to digitization and errors in segmentation may affect the shape region. Since any good shape retrieval method is expected to satisfy the property of invariance, we evaluate performance of our proposed region-based shape descriptor and similarity matching criteria under these conditions.

The experimentation was carried out to check the efficacy of the proposed scheme for robustness to scaling and rotation invariance, as well as to study the performance of our similarity-based retrieval results. The indexing method has been explained

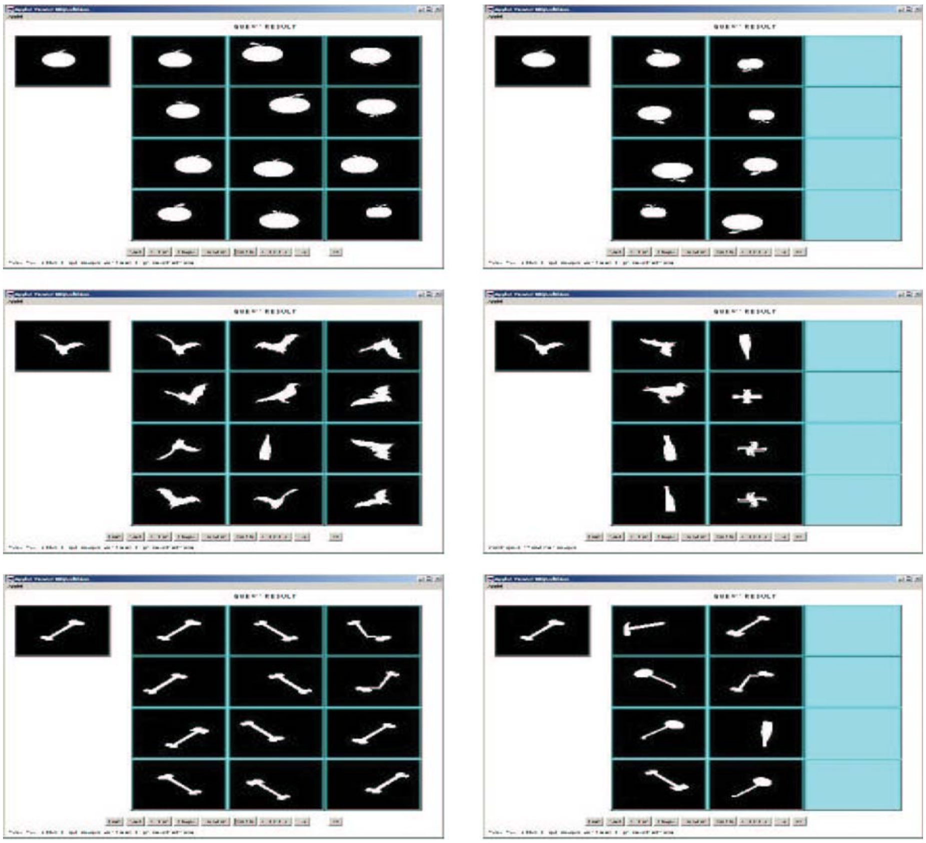


Fig. 29. Retrieval results on MPEG-7 standard database: (A) query on class Apple (top row), (B) query on class Bat (second row), (C) query on class Bone (third row), (D) query on class Car (fourth row), and (E) query on class Device1 (last row).

earlier in Section 5. An 8×8 grid is overlapped on the region formed by the minimum bounding rectangle for each object. This is then normalized for transformation, scaling, and rotation invariance. Similarity matching is done by using the row and column coverage of shape regions within this grid as explained in Section 5. Threshold for matching was set to different values of 4, 6, and 8. Finally, we selected value of 8 as it was found to give the best results.

Each image in the database is indexed on shape features. A hash table stores images of similar index based on the shape features extracted. Images are retrieved based on the matching shape index. Then shape of the region in the query image is compared to the region shapes in the displayed set to find images similar on the basis of criteria used.

Each of the 184 base images was used as a query image and precision performance evaluated. Matching was computed from the number of similar images (i.e., images belonging to the same class) in the top 20 retrieved images. The best result possible is thus $184 \times 20 = 3680$ matches. Table 8 presents results depicting the *Precision rates* for some of the different classes tabulated for retrieval from top 2 to top 20 retrieved images. Each row pertains to the average precision value for all query images on that class. Each column in the table represents precision rates computed using the specified number of retrievals. For example, consider the first row in Table 8. For top 20 retrievals, the average number of correct matches (over the four different query images for that class) is 100% (20 images of apple match the query image out of the possible 20 images in the database, i.e., the precision is $20/20 = 1.0$ or 100%). The other values in the tables have been computed in a similar way.

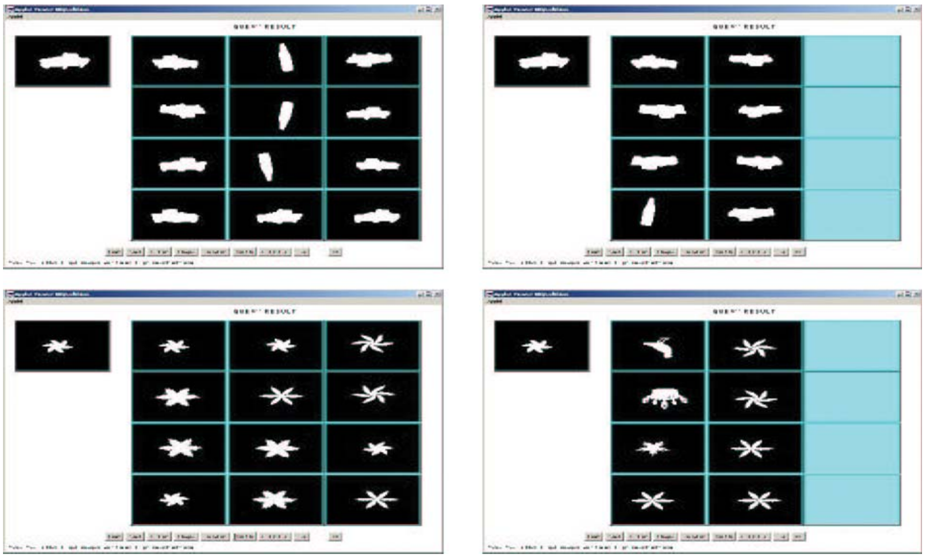


Fig. 29. (*continued*)

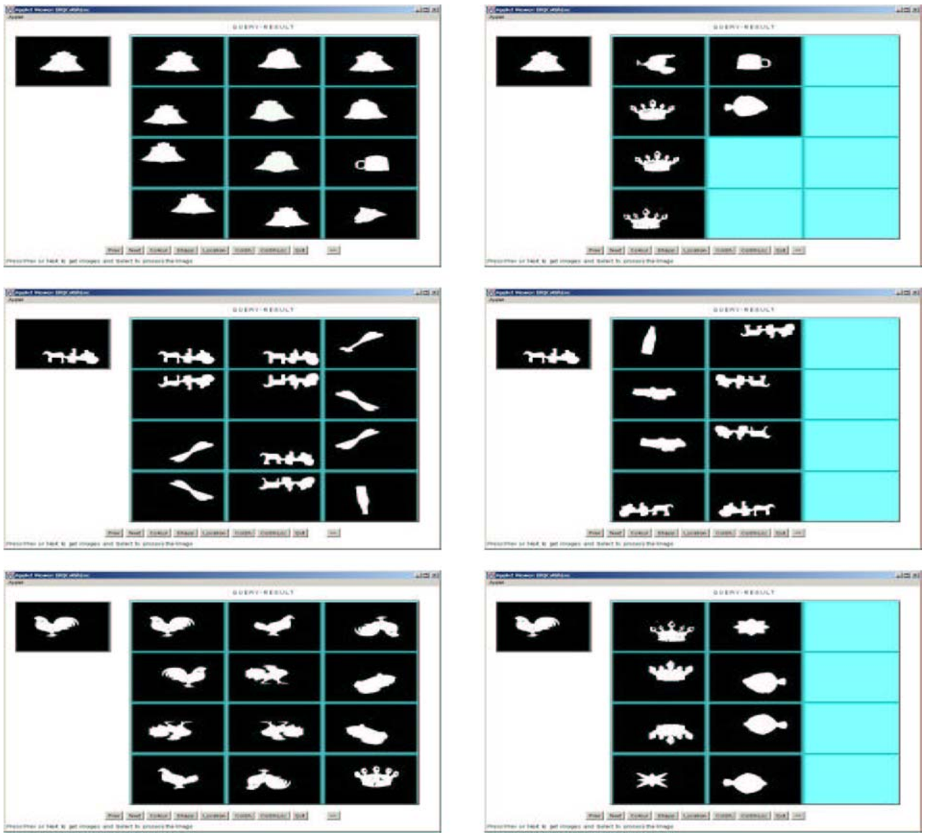


Fig. 30. Retrieval results on MPEG-7 standard database: (A) query on class Bell (top row), (B) query on class Carriage (second row), (C) query on class Chicken (third row), (D) query on class Lmfish (fourth row), and (E) query on class Device5 (last row).

Table 9 depicts the *Recall rates* for some of the representative classes in the database. Here also, each of the 184 base images were used as a query image and number of matches in the top 40 retrieved items was counted.

It is observed that the values of recall rates are reasonable for most of the image classes. Some of the classes do not return favorable values due to the fact that objects in these classes are similar in shape features to the shapes of objects in other classes. Hence, the retrieval rates are low in such cases. Also, another contributing factor could be our normalizing process which converts every image to a standard size of 192×128 for our experimentation. In some cases this has resulted in a change in the aspect ratio of the objects causing a distortion in the shape.

The low retrieval rate for some of the classes can also be attributed to the fact that rotational and scaling transformations may cause the object shape to match shape of another object rather than members of its own class. Also, in similarity matching 100% retrieval rate is not possible. This is because some classes contain objects with

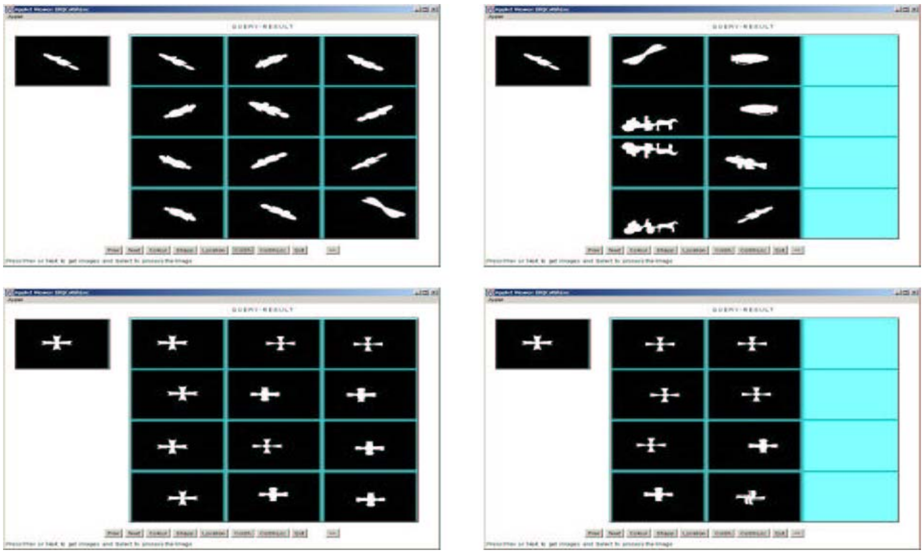


Fig. 30. (*continued*)

significantly different shapes, so that it is not possible to group them into the same class using only shape information. Examples of such cases are shown in Fig. 27. It is seen that image objects in the first row are more similar to image objects in second row (in corresponding columns) when compared to image objects from their own classes. This can be effectively taken care of by including color information.

The search method generates a ranked list of results. Then for queries in each class, we plot the set of precision and recall values by setting the cut-off point successively at $10, 20, \ldots$ of the retrieved items. In this case, the precision decreases monotonically as a function of recall. Fig. 28 shows the precision recall graph plotting the average precision retrieval rates for some of the sample classes.

Using our JAVA based search engine (query-by-example), sample outputs for retrieval are shown in Figs. 29 and 30 for matching on shape. It can be observed that images which are non-similar in shape get eliminated.

8. Conclusions

A combined color–shape–location based low-dimensional indexing technique has been implemented. Results obtained are favorable for a small database whose search space is about 1800 (25 colors $\times$ 8 shapes $\times$ 9 locations). The performance of the system is seen to be effective and has been measured using the standard recall versus precision graphs.

Our scheme is limited by the segmentation process in the initial phase to obtain homogeneous color regions. The color clustering is made robust by applying a pre-processing step of using anisotropic diffusion to smoothen out the color regions.

Also there may be disambiguities in the extent of the level of positioning the regions within the given sub-locations in the image space. The scheme is easily extendible to domains containing more colors, shapes, and locations so as to overcome these limitations.

Although we have implemented the integrated color–shape–location indexing on domains of flags, flowers, fruits, vegetables, etc., and also on the image database testbed used in the MPEG-7 core experiments, the indexing scheme can be applied to a domain which is amenable to this environment where homogeneous regions can be segmented out of the images like in the case of Logo databases of companies, identifying and inspecting articles in a factory floor, etc.

An attempt has been made to use color and shape descriptors which are similar to those specified by the MPEG-7 standard framework for visual content description. We have implemented a system for indexing images using an integrated index obtained by combining the color, shape, and location feature of regions in an image.

Presently we query the system by query-by-example image. We are now working towards storing the extracted features of the images in a relational database and using SQL to implement queries to combine multiple query images using AND, OR, and NOT operators. This would greatly enhance the users requirements of querying interactively to provide better results relative to user relevance.

The SQL based querying can be extended to handle textual queries so as to fit into the metadata based indexing paradigm advocated and standardized by systems such as MPEG-7 and Dublin core.

Acknowledgments

The authors acknowledge the numerous suggestions as well as critical and useful comments of the reviewers facilitating improvements to the paper.

References

[1] R.C. Veltkamp, M. Tanase, Content-Based Image Retrieval Systems: A Survey, UU-CS-2000-34, Department of Computer Science, Utretch University, October 2002.

[2] R.C. Veltkamp, M. Tanase, A survey of content-based image retrieval systems, in: O. Marques, B. Furht (Eds.), Content-Based Image and Video Retrieval, Kluwer, Dordrecht, 2002.

[3] A.W.M. Smuelders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.

[4] A. Del Bimbo, Visual Information Retrieval, Morgan Kaufmann (Ed.), San Francisco, USA, 1999.

[5] M. de Marsicoi, L. Cinque, S. Levialdi, Indexing pictorial documents by their content: a survey of current techniques, Image Vision Comput. 15 (2) (1997) 119–141.

[6] Y. Rui, T.S. Huang, S.F. Chang, Image retrieval: current techniques, promising directions and open issues, J. Visual Commun. Image Representation 10 (1999) 39–62.

[7] A.A. Goodrum, Image information retrieval: an overview of current research, Inform. Sci. 3 (2) (2000) 63–67.

[8] M. Flickner et al., Query by image and video content: the QBIC system, IEEE Comput. 28 (9) (1995) 23–32.

[9] W. Niblack et al., The QBIC project: querying images by content using color, texture and shape, Proc. SPIE, Storage Retrieval Image Video Database 1908 (1993) 173–187.
[10] W.Y. Ma, B.S. Manjunath, Netra: a toolbox for navigating large image database, Proc. IEEE Internat. Conf. on Image Process. 1 (1997) 568–571.
[11] J.R. Smith, S.F. Chang, VisualSEEK: a fully automated content-based image query system, ACM Multimedia (1996) 87–98.
[12] V.E. Ogle, M. Stonebaker, Chabot: retrieval from a relational database of images, IEEE Comput. 28 (9) (1995) 40–48.
[13] C. Carson et al., Region-Based Image Querying, in: Proc. CVPR'97 Workshop on Content-based Access to Image and Video libraries (CAIVL'97), 1997.
[14] R. Schettini, G. Ciocca, S. Zuffi, Color imaging science: exploiting digital media, in: R. Luo, L. MacDonald, Wiley, New York, 2001. Available from <http://www.citeseer.nj.nec.com/489355.html>.
[15] M.J. Swain, D.H. Ballard, Color indexing, Internat. J. Comput. Vision 7 (1) (1991) 11–32.
[16] J. Hafner et al., Efficient color histogram indexing for quadratic form distance functions, IEEE Trans. Pattern Anal. Mach. Intell. 17 (7) (1995) 729–736.
[17] M. Stricker, A. Dimai, Color indexing with weak spatial constraints, Proc. SPIE, Storage Retrieval Still Image Video Databases IV 2670 (1996) 29–40.
[18] J. Huang et al., Image indexing using color correlograms, Internat. Conf. Computer Vision Pattern Recognition (1997) 762–768.
[19] H. Zhang et al., Image retrieval based on color features: an evaluation study, Proc. SPIE 2606 (1995) 212–220.
[20] K.C. Ravishankar, B.G. Prasad, S.K. Gupta, K.K. Biswas, Dominant color region based indexing technique for CBIR in: Proc. Internat. Conf. on Image Analysis and Processing (ICIAP'99) Venice, Italy, September 1999, pp. 887–892.
[21] M.S. Kankanahalli, B.M. Mehtre, J.K. Wu, Cluster-based color matching for image retrieval, Pattern Recognition 29 (4) (1996).
[22] X. Wan, C.J. Kuo, A multiresolution color clustering approach to image indexing and retrieval, in: Proc. ICASSP, 1998.
[23] D. Androutsos et al., A perceptually motivated technique to query-by-example using color cardinality, in: Proc. Multimedia Storage & Archiving Systems IV, Boston, USA, September 19–22, 1999.
[24] V. Chitkara et al., Content-based image retrieval using binary signatures, Technical Report TR-00-18, Department of CS, University of Alberta, Edmonton, Alberta, Canada, September 2000.
[25] R.O. Stehling et al., Color-shape histogram for image representation and retrieval, Technical Report IC-00-11, Institute of Computing, State University of Campinas, Brazil, July 2000.
[26] B.C. Ko et al., Image retrieval using flexible image subblocks, in: 15th ACM Symposium on Applied Computing (SAC), Como, Italy, 2000.
[27] R.C. Veltkamp, Shape matching: similarity measure and algorithms, UU-CS-2001-03, Department of Computer Science, Utretch University , February 2001.
[28] R.C. Gonzalez, P. Wintz, Digital Image Processing, second ed., Addison-Wesley, Reading, MA, 1987.
[29] D. Mohamad, G. Sulong, S.S. Ipson, Trademark matching using invariant moments, in: Second Asian Conf. on Computer Vision, Singapore, December 5–8, 1995.
[30] G. Cortelazzo et al., Trademark shapes description by string-matching techniques, Pattern Recognition 27 (8) (1994) 1005–1018.
[31] A.K. Jain, A. Vailaya, Image retrieval using color and shape, in: Second Asian Conf. on Computer Vision, December 5–8, Singapore, 1995, pp. 529–533.
[32] R. Mehrotra, J.E. Gary, Similar-shape retrieval in shape data management, IEEE Comput. 28 (9) (1995) 57–62.
[33] H.V. Jagadish, A retrieval technique for similar shapes, in: Proc. ACM SIGMOD, Colorado, ACM, New York, May 1991, pp. 208–217.
[34] G. Lu, A. Sajjanhar, Region-based shape representation and similarity measure suitable for content-based image retrieval, Multimedia Systems 7 (1999) 165–174.

[35] B.G. Prasad, S.K. Gupta, K.K. Biswas, Content-based image retrieval using low-dimensional shape index, in: 1st National Conf. on Document Analysis and Retrieval (NCDAR2001), Mandya, India, July 13–14, 2001.
[36] B.G. Prasad, S.K. Gupta, K.K. Biswas, Color and shape index for region-based image retrieval, in: 4th Internat. Workshop on Visual Form (IWVF4), Capri, Italy, May 28–30, 2001. Proceedings published in LNCS 2059 (2001) 716–725.
[37] S.K. Chang, Q.Y. Shi, C.Y. Yan, Iconic indexing by 2-D strings, IEEE Trans. Pattern Anal. Mach. Intell. 9 (3) (1987) 413–428.
[38] H. Samet, The quadtree and related hierarchical data structures, ACM Comput. Surveys 16 (2) (1984) 187–260.
[39] V.N. Gudivada, V.V. Raghavan, Design and evaluation of algorithms for image retrieval by spatial similarity, ACM Trans. Inform. Systems 13 (2) (1995) 115–144.
[40] C. Faloutsos et al., Efficient and effective querying by image content, J. Intell. Inform. Systems 3 (1994) 231–262.
[41] M. Stricker and M. Orengo, Color indexing with weak spatial constraints, in: Proc. SPIE Conf. on Visual Commun., 1996.
[42] T. Sikora, The MPEG-7 visual standard for content description—an overview, IEEE Trans. Circuits Systems Video Technol. 11 (6) (2001) 696–702.
[43] T.F. Syeda-Mahmood, Data and model-driven selection using color regions, Internat. J. Comput. Vision 21 (1/2) (1997) 9–36.
[44] M.M. Mitschke, Flexible content-based image retrieval system, Diploma Report, Institute of Mathematics, Fredrich-Alexander University, Erlangan-Nurenburg, 1997.
[45] M.J. Black, G. Sapiro, D.H. Marimont, H. David, Robust anisotropic diffusion, IEEE Trans. Image Process. 7 (3) (1998) 421–432.
[46] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639.
[47] J.M. Corridoni, A. Del Bimbo, E. Vicario, Image retrieval by color semantics with incomplete knowledge, J. Am. Soc. Inform. Sci. 49 (3) (1998) 267–282.
[48] S. Jeannin and M. Bober, Description of core experiments for MPEG-7 motion/shape, ISO/IEC JTC1/SC29/WG11/MPEG99/N2690, Seoul March, 1999.