A Histogram-Based Approach for Object-Based Query-by-Shape-and-Color in Multimedia Databases*

Ediz Şaykol, Uğur Güdükbay and Özgür Ulusoy
Department of Computer Engineering, Bilkent University
06533 Bilkent, Ankara, Turkey
{ediz,gudukbay,oulusoy}@cs.bilkent.edu.tr

January 17, 2002

Abstract

Considering the fact that most of the multimedia database systems include querying by object features, a novel approach for color and shape querying and retrieval is proposed. The approach is histogram-based and it is supported by an auxiliary object extraction tool in the query-by-feature component of a multimedia database system. The object extraction tool is semi-automatic, hence it can successfully capture salient object regions for most of the images and/or video frames. The use of the object extraction tool facilitates object-based querying of color and shape content. In the histogram-based approach, the color and shape can be integrated to improve the performance of querying. It is shown through performance experiments that the histogram-based approach overcomes the deficiencies of the existing methods for querying by shape. The evaluation of the effectiveness and the robustness of the histogram-based approach is also presented via performance experiments.

Keywords. histogram-based approach, query-by-shape, query-by-color, histogram comparison, object-based querying, kinematics model for polygon simplification, random noise model, retrieval effectiveness.

*This work is supported by Turkish Scientific and Technical Research Council (TÜBİTAK) under the grant number EEEAG-199E025.

1 Introduction

Advances in multimedia technology accelerate the amount of digitized information so as the data stored as image and video content. Both of these data types require application-dependent processing strategies, easy-to-handle storage and indexing methods as well as sophisticated querying mechanisms. As far as the querying process of image and video databases is concerned, spatial (for both image and video data) and spatio-temporal (for video data only) information are considered. Besides, the semantic meaning residing in images and/or videos can be queried through database querying systems. Not only the images and videos, but also the objects in these data types can be enrolled in the queries with the corresponding object features (e.g., color, shape, texture, size). Specific tools are developed in order to extract the object features from images and/or videos. These tools may require user assistance for the object extraction process of realistic data.

Most of the existing query specification techniques in multimedia database systems deal with querying of the content of the image and/or video data and the retrieval of the data types according to their content. This type of querying and retrieval is known as *content-based retrieval (CBR)*. Most of the CBR systems include color as the major querying feature. *Color histograms* [FSN$^+$95, SB91] and *color sets* [CCM$^+$97, SC96b] are some of the methods that are used within these systems. CBR systems are intended to query the color content of image and video data as a whole. Region-based color queries and the queries specifying spatial layout of the color regions are supported in some of these systems [CTB$^+$99, HKM$^+$97]. However, query-by-shape is not supported as frequent as query-by-color. The systems that are capable of handling shape queries use various pattern-matching methods (e.g., moments [TC92], turning angles [ACH$^+$91], Hausdorrf distance [HKR93]) and work only with images as a whole. The general shape information of the images can be queried as well as the shapes of objects in these systems. For object-based shape queries, edge-detection is applied to the image and then the polygonal object boundaries are compared. Besides, the texture and size features of the objects can be queried in CBR systems.

In this paper, we discuss a histogram-based approach for querying image and video data according to shape and color content. The approach is developed as the core part of the query-by-feature component of our rule-based video querying system [DUG00]. The system handles the spatial relations (in images and videos) and spatio-temporal relations (in videos) among the objects. The video data is pre-processed with a semi-automatic tool, namely *Fact Extractor*, that extracts the salient objects and the relations between these objects in video frames. During this process, the keyframes of videos are determined and the shape and color information of the objects are extracted for each keyframe. Thus, the histogram-based approach treats videos as a collection of images so that each keyframe can be considered as a single image. Another semi-automatic tool, *Object Extractor* [ŞGU01], processes the images and videos –through their keyframes– and extracts the object regions, as well as the corresponding object features. Both of the Fact Extractor and Object Extractor tools are designed to work in parallel for the sake of efficiency.

In the histogram-based approach, three histograms -distance, angle and color- are used to store shape and color content of image and video data. To the best of our knowledge, this is the first approach that employs these three histograms for object-based querying of image and video data by shape and color. The contributions of the histogram-based querying approach can be summarized as follows:

Object-based query-by-shape: The method can be applied in a very easy and intuitive way since the shape information of the objects is stored in distance and angle histograms. The shape information is gathered from the interior and boundary points of the extracted object. Since we deal with images, these points are the pixels in the extracted object region. For each object region, the center of mass of the object (c_m) forms the basis for shape information of the object. This technique resembles the visualization of the human eye and resolves the drawbacks of the existing methods since the boundaries of the objects are considered in most of the other systems.

Object-based query-by-color: Since the color information of the objects is stored in color histogram, our querying method can be used for query-by-color as well. All of the histograms for both shape and color information can be filled concurrently, which speeds up the process.

Processed data types: As opposed to most of the other methods, our histogram-based querying method is applicable not only to images but also to videos. As discussed above, the extracted keyframes are processed as single images. Thus, the features of the objects residing in any type of image and/or video data can be queried.

Sensitivity to noise: The method is sensitive to noise on the boundaries, which other methods suffer from. Since the method gathers information from the pixels, the length and shape of the boundary as well as the presence of holes in the interior have no effect on the method. Besides, there is no polynomial restriction on the object boundaries as in *turning angle* method [ACH$^+$91]. Thus, the method is successful for processing noisy objects.

The organization of the paper is as the following: Section 2 presents the related work on querying by shape and color as well as histogram comparison techniques used within querying schemes. The object extraction tool is summarized in Section 3 and the histogram-based approach is discussed in Section 4. In Section 5, two approaches for integrating the color and shape content of objects for querying are explained. The performance experiments to test the effectiveness of the histogram-based approach are presented in Section 6. Section 7 concludes the paper.

2 Related Work

2.1 Query-by-Color Approaches

Color is the most frequent feature that is used in multimedia data querying and retrieval systems. The existing CBR systems are enriched with an easy-to-index data structure in order to facilitate the query processing. Specification of a sample query image is widely used in query-by-example [Zlo77]. The images that are similar enough to the query image are retrieved from the database. As alternative to query-by-example, the users may specify the percentages of the color channels (e.g., red, green, blue for RGB color space) in the color space and form a color query. In this case, the images that have closer color percentages to the query color percentages are retrieved.

The *color histograms* [FSN$^+$95, SB91] are used to represent the color distribution in an image or a video frame. Mainly, the color histogram approach counts the number of occurrences of each unique color on a sample image. Since an image is composed of pixels and each pixel has a color, the color histogram of an image can be computed easily by visiting every pixel once. By examining the color histogram of an image, the colors existing on the image can be identified with their corresponding areas as the number of pixels. One possible way of storing the color information is to use three different color histograms for each color channel. Another possible method is to have a single color histogram for all of the color channels. In the latter approach, the color histogram is simply a compact combination of three histograms and the empty slots can be discarded easily. The histogram approach is commonly used in most of the existing systems supporting query-by-color content. A supplementary information about color for an image is the average color and it may be stored along with the color histogram for the sake of efficiency since it can be computed from the color histogram in one pass.

In [SC96a], *colorsets* are proposed instead of color histograms. The colorsets are binary masks on color histograms and they store the presence of colors as 1 without considering their amounts. For the absent colors, the colorsets store 0 in the corresponding bins. The colorsets reduce the computational complexity of the distance between two images. Besides, by employing colorsets region-based color queries are possible to some extent. On the other hand, processing regions with more than two or three colors is quite complex.

Another image content storage and indexing mechanism is *color correlograms* [HKM$^+$97]. It involves an easy-to-compute method and includes not only the spatial correlation of color regions but also the global distribution of local spatial correlation of colors. In fact, a color correlogram is a table each row of which is for a specific color pair of an image. The k-th entry in a row for color pair (i, j) is the probability of finding a pixel of color j at a distance k from a pixel of color i. The method resolves the drawbacks of the pure local and pure global color indexing methods since it includes local spatial color information as well as the global distribution of color information.

Not only the image as a whole but also the color regions that exist in the image can be queried. Since there may exist more than one region in an image, the spatial relations among the color regions can be queried. Based on this discussion, the color query types can be classified as follows:

Global Color Information

- Find all images that have a color distribution similar to the given sample image (i.e., query-by-example).
- Find all images that have color channels similar to the specified color channel percentages (e.g., `Find images having 40% Red, 40% Green, 20% Blue`).

Local Color Information

- Find all images that have a region (or a group of regions) with the color distribution similar to the given region (or a group of regions) (e.g., `Find images having a green region and a yellow region`).

Local Color Information with Spatial Relations

- Find all images that have a group of regions with the color distribution similar to the given group of regions, and the group of regions satisfy the specified spatial relation(s) (e.g., `Find images having a green region on the southwest of a yellow region`).

Since the regions may also enclose more than one color, the color channel percentages within regions may be queried as well (e.g., `Find regions having 40% Red, 40% Green, 20% Blue`).

2.2 Query-by-Shape Approaches

Shape is an important feature that identifies objects on images. In order to retrieve objects according to their features, color is inadequate in most of the circumstances. Shape strengthens the image retrieval since it encodes the object-based information in a way different from color. Thus, query-by-shape is supported by CBR systems that allow object feature-based queries.

In QBIC system [FSN$^+$95], area, circularity, eccentricity, major-axis direction, object moments, tangent angles and perimeter identify the objects in an image database. Specification of these parameters requires specific analysis during the database population phase. In [JV96], the authors propose a method for image-based shape retrieval. Their method necessitates edge-detection on the image and they use the well-known Canny edge-detection algorithm [Can86] for this purpose. The identified edges are stored in a histogram for an image. Chang et al. explain a process for gathering object-based shape information including eigenvalue analysis, generation of first and second order moments [CCM$^+$97]. In Photobook [PPS94], the authors propose a method based on eigenvectors of the object's physical model for shape representation. They use a finite element method [SP93] to construct the physical model for an object.

Basically, shape retrieval is performed in two different ways. The first method is query-by-example where a sample object is supplied and the objects that have similar shape structure are returned (e.g., [GT98]). The second shape retrieval method is query-by-visual sketch. An object can be specified with a combination of geometric primitives (e.g., rectangle, circle, triangle) and the objects that are close to this sketch are retrieved. Additionally, most of the CBR systems adopt shape comparison and similarity techniques to shape retrieval process. The following section briefly discusses the histogram comparison techniques employed for shape comparison. In [Vel01], a detailed discussion on these techniques is presented. Moreover, a taxonomy on shape description techniques can be found in [SSS00].

2.3 Histogram Comparison Techniques

In this section, the comparison techniques and metrics for histograms or point-sets are discussed. The comparison techniques are applied to the histograms in our approach for query-by-color and shape in order to determine the similarity between two histograms. In these techniques, the histograms must be normalized to have a proper similarity distance.

Let $H[1..n]$ denote a histogram with n bins enumerated from 1 to n. The normalized histogram $H_N[1..n]$ is defined as follows:

$$H_N[i] = \frac{H[i]}{\sum_i^n H[i]}. \tag{1}$$

2.3.1 Discrete Metric

The discrete metric method defines a distance value between two histograms H_1 and H_2 representing information of two objects as follows:

$$d_D(H_1, H_2) = \begin{cases} 0, & if\ H_1 = H_2 \\ 1, & otherwise \end{cases} \tag{2}$$

This method is very sharp in the sense that it classifies two point-sets as either similar or not. However, in shape retrieval this is not the case because the partial similarity between objects is needed. Thus, the method is generally inapplicable for shape retrieval.

2.3.2 L_p Distance Metric

This method constitutes a class of distance metrics such that if p is 1, L_1 is the *Manhattan* distance, and if p is 2, L_2 is the *Euclidean* distance among the point-sets. The general formulation of the L_p distance between two histograms H_1 and H_2 is as follows:

$$d_{L_p}(H_1, H_2) = (\sum_{i=0}^{n} |H_1[i] - H_2[i]|^p)^{1/p}. \tag{3}$$

This metric is also called *Minkowski* distance and it requires the inputs to be of the same size. L_2 distance is commonly used within retrieval techniques when L_1 distance is inadequate (e.g., [ACH$^+$91]).

2.3.3 Quadratic Distance Metric

This technique differs in cross-wise comparisons from the L_p distance metric. It is used for calculating color histogram distance in QBIC system [FEF$^+$94]. The distance between two histograms H_1 and H_2 is defined as:

$$d_Q(H_1, H_2) = \sum_{i=0}^{n} \sum_{j=0}^{n} (H_1[i] - H_2[i]) \cdot a_{i,j} \cdot (H_1[j] - H_2[j]), \tag{4}$$

where $a_{i,j}$ is a coefficient designating the perceptual similarity between histogram bins (e.g., for color histograms, it is based on the perceptual similarity of the corresponding colors). In order to set a value to the coefficient $a_{i,j}$, another coefficient $d_{i,j}$, denoting the distance between i and j that is normalized with respect to the maximum distance, is determined [HSE$^+$95]. Thus, assigning $(1 - d_{i,j})$ to $a_{i,j}$ reduces Equation (4) to the following:

$$d_Q(H_1, H_2) = -\sum_{i=0}^{n} \sum_{j=0}^{n} (H_1[i] - H_2[i]) \cdot d_{i,j} \cdot (H_1[j] - H_2[j]). \tag{5}$$

2.3.4 Histogram Intersection

In this technique, two normalized histograms are intersected as a whole. This technique is employed by Swain and Ballard [SB91] for color feature. The similarity between the histograms is a floating point number between 0 and 1. Equivalence is designated with similarity value 1 and the similarity between two histograms decreases when the similarity value approaches to 0. Obviously, both of the histograms must be of the same size, otherwise the similarity is undetermined.

Let $H_1[1..n]$ and $H_2[1..n]$ denote two histograms of size n, and S_{H_1,H_2} denote the similarity value between H_1 and H_2. Then, S_{H_1,H_2} can be expressed by the distance between the histograms H_1 and H_2 as:

$$S_{H_1,H_2} = d_{HI}(H_1, H_2) = \frac{\sum_i^n \min(H_1[i], H_2[i])}{\min(|H_1|, |H_2|)}. \tag{6}$$

2.4 Turning Angle Representation for Shape Comparison

This method processes two polygonal shapes and produces turning angle representations of the shapes. Then, L_2 distance metric is employed to get a similarity distance from the representations. Formally, the turning angle representation $\Theta_P(s)$ of a polygonal shape P is a collection of turning angles as a function of the arc length s. A turning angle at a point is the counter-clockwise angle between the tangent of P at that point and the x-axis [ACH+91]. Figure 1 illustrates the turning angle representation of two images initiated from the bottom-left vertex.

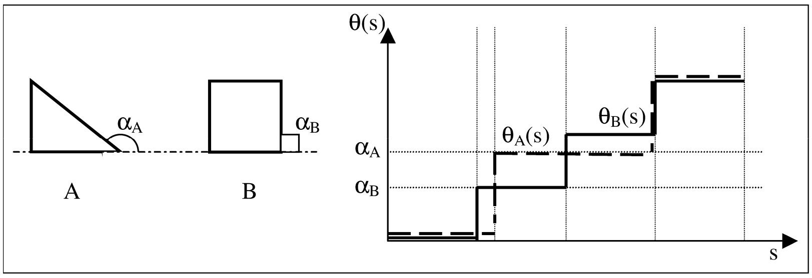


Figure 1: Turning Angle Representation

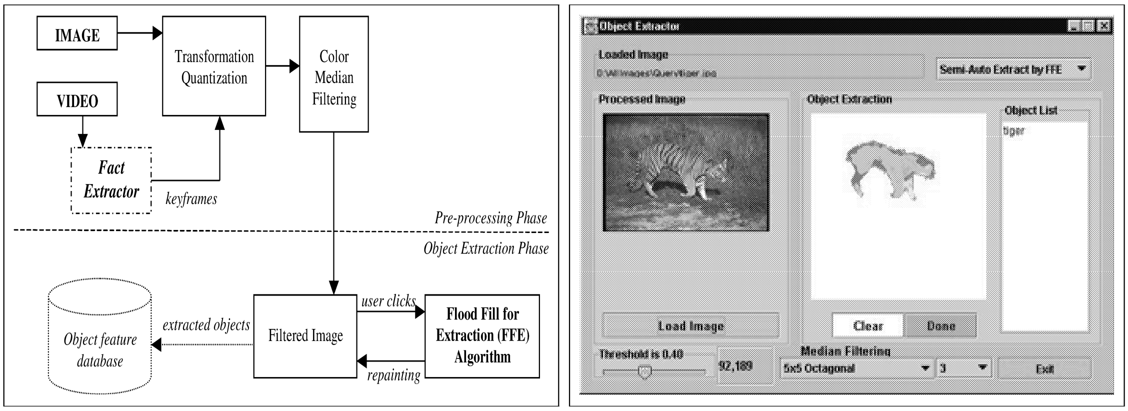
Let $\Theta_A(s)$ and $\Theta_B(s)$ be turning angle representations for polygonal shapes A and B, respectively. According to the L_p metric, the distance between two polygonal shapes is given as follows:

$$d_T(A, B) = (\int |\Theta_A(s) - \Theta_B(s)|^p \, ds)^{1/p}. \tag{7}$$

In order to satisfy invariance under rotation, $\theta_P(s)$ can be shifted with an angular amount θ. The arc length s is scaled to 1 for all of the input shapes, which satisfies scale invariance. The method is also invariant under translation. Thus, it is widely used in shape retrieval systems. In [SAF94], it is argued that this method is superior to the other methods (e.g., moments [TC92], Hausdorff distance [HKR93], sign of curvature [SAF94]) due to the human perceptual judgments.

3 Object Extraction

In this section, a semi-automatic object extraction tool, called Object Extractor [ŞGU01], is presented. The main motivation of *Object Extractor* is to facilitate object extraction phase of query-by-feature component and to provide an appropriate output structure for object-based querying. The output of the tool is used as meta-data for the histogram-based querying in which the color and shape content of the output is processed. The overall architecture and the user interface of the tool is shown in Figure 2.



(a) (b)

Figure 2: The Object Extractor. (a) Overall architecture. (b) User interface.

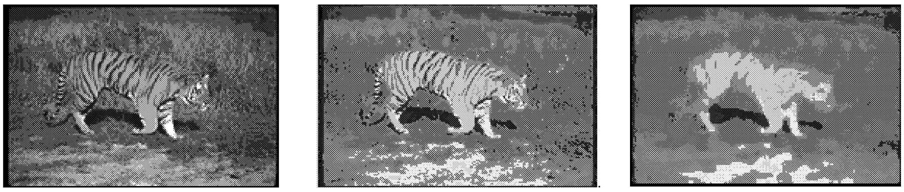
3.1 Preliminaries

One of the most important features of objects in image and video data is *color*. Each pixel in an image has a three-dimensional color vector and different color spaces encode color information based on various approaches. One of these color space models is the hardware-oriented *Red-Green-Blue Model (RGB)* where the color vector of a pixel p is the compound of red, green and blue channels $v_p = (r, g, b)$. Another color space model is the *Hue-Saturation-Value Model (HSV)* that is based on color descriptions rather than individual color components $v_p = (h, s, v)$. The RGB model has a major drawback: it is not perceptually uniform. Therefore, most of the systems use color space models other than RGB, such as HSV [HB94].

3.1.1 Transformation and Quantization

The color regions are perceptually distinguishable to some extent. The human eye cannot detect small color differences and may perceive these very similar colors as the same color.

This leads to the *quantization* of color, which means that some pre-specified colors will be present on the image and each color is mapped to some of these pre-specified colors. One obvious consequence of this is that each color space may require different levels of quantized colors, which is nothing but a different quantization scheme. In Figure 3, the effect of color quantization is illustrated. Figure 3 (a) is the original image with RGB color space and (b) is the image produced after transformation into HSV color space and quantization. A detailed explanation of color space transformations (from RGB into HSV) and quantization can be found in [SC96a].



(a) (b) (c)

Figure 3: Transformation, quantization and color median filtering of an image. (a) Original image. (b) Image produced by applying RGB to HSV color transformation and quantization. (c) Image produced after applying color median filtering.

3.1.2 Color Median Filtering

Not all the colors in an image are dominant. Dominance is in the sense that some of the colors may reside in a region relatively small than the others. The *color median filtering* technique [Rus99], a famous method for neighborhood ranking, eliminates these non-dominant colors and produces a filtered image (Figure 3(c)). This technique facilitates the object extraction process because it also eliminates the noise of the color on the object boundaries to some extent.

In order to achieve the best filtering, the color median filter procedure may be applied successively. For most types of images, applying color median filtering 3–5 times gives a proper view. Moreover, there exist different types of color median filters. The basic color median filtering types are 3×3 box, 5×5 box, 5×5 octagonal, 7×7 box and 7×7 octagonal filters. Basically, the smoothness on the edges in the image corresponding to the object boundaries vary among these color median filtering types. Figure 4 illustrates the color median filtering types. In each type, the neighbors of the black square is ranked to determine the color of the pixel.

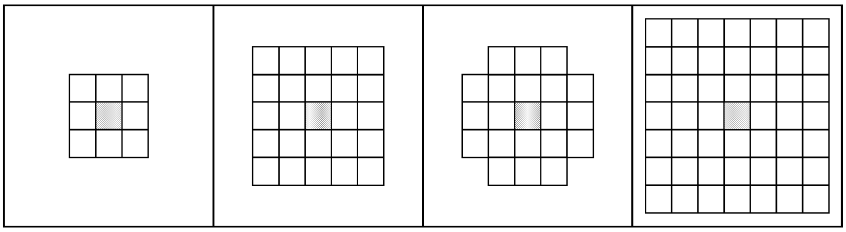


Figure 4: A collection of color median filtering types. (a) 3×3 box (b) 5×5 box (c) 5×5 octagonal (d) 7×7 box.

3.2 Object Extraction Process

The Object Extractor employs Flood Fill for Extraction (FFE) algorithm, an improved version of the flood fill algorithm for polygon filling [HB94]. The FFE algorithm is initiated with a user-clicked pixel on the object to be extracted and recursively checks the neighbors of the initiative pixel.

In the pre-processing phase of Object Extractor, filtering is used to eliminate the noise on the image edges. This pre-processing phase includes color space transformation, color quantization and color median filtering steps. The color space of the image is transformed from RGB to HSV due to the perceptual inconsistency and non-uniformity of RGB color space. Then the color, in HSV color space, is quantized into 18 hue, 3 saturation, 3 value levels as well as 4 gray levels adding up to 166 different colors. The last step in this pre-processing phase is color median filtering, and having completed all of these steps, the filtered image becomes noise-free especially on the image edges (i.e., possible object boundaries).

The Object Extractor has an easy-to-use user interface, as shown in Figure 2 (b), in which all of the user's actions are handled with simple mouse actions. The loaded image is processed via the tool with the current color difference threshold and color median filtering type. In order to achieve better smoothness, color median filtering can be applied as many times as desired. The user extracts color regions separately by initiating a separate execution of FFE algorithm. When the user is satisfied with the current extracted form of the object, he/she can press the 'done' button and the extracted color regions form the extracted object.

The output structure of Object Extractor for representing a single extracted object O can be listed as,

- the set of pixels P_I in the interior of O,
- the set of pixels P_B on the boundary of O,

- the center of mass c_m of O, and
- the center of minimum bounding rectangle c_{MBR} of O.

The first three outputs, P_I, P_B and c_m are used in the histogram-based querying approach. However, the last one, c_{MBR}, is not directly used in the formation of the histograms.

4 Histogram-Based Approach for Query-by-Color-and-Shape

Our histogram-based querying approach is basically motivated with computational geometry concepts and intended to use a similarity measure between images much like the human vision system does. For this motivation, the interrelation among pixels is very important and should be taken into account for object-based similarity. This is because each pixel provides a piece of information about objects and should be considered in the shape and color content.

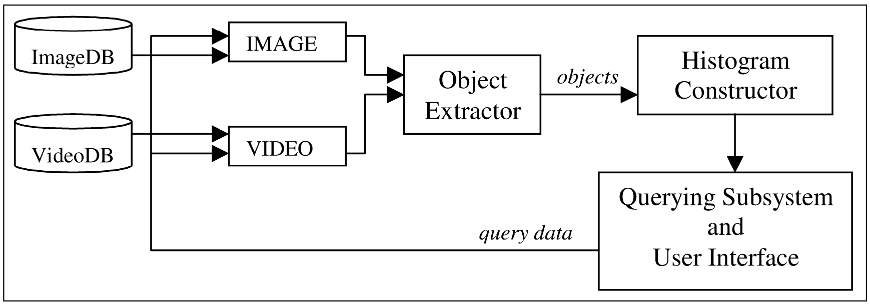


Figure 5: The overall architecture of the histogram-based querying scheme.

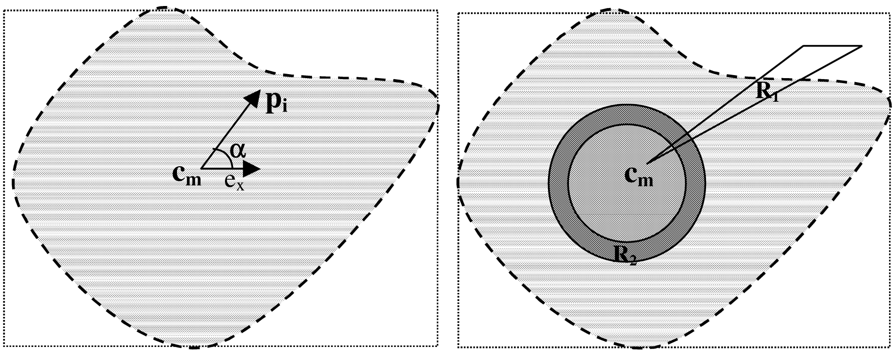
As shown in Figure 5, the histogram-based querying process can be divided into three phases: object extraction phase, histogram construction phase and querying phase. In the object extraction phase, the color information is encoded in HSV color space since RGB color space is not perceptually uniform. The three-dimensional RGB color vector is transformed into three-dimensional HSV color vector for each pixel. Then, color quantization and color median filtering techniques are performed on the image. The next step is the histogram construction phase where the color information of the filtered image is stored in one color histogram and the shape information is stored in two specialized histograms. The last step, the querying phase, involves an integrated use of the interface and the querying sub-system, in which the user specifies an example for the query. The specified query example also passes through the object extraction and histogram construction phases. The results of the query is the 'most similar' objects to the given example.

The three histograms used for storing shape and color information can be described as follows:

Distance Histogram stores the Euclidean distance between the center of mass of an object (c_m) and all of the pixels within the object. The distance between a pixel p and c_m accumulates into the corresponding bin in the distance histogram.

Angle Histogram stores the counter-clockwise angle between pixel vectors and the unit vector on the x-axis (e_x). The pixel vector v_p for a pixel p is a vector directed from c_m to p. The unit vector e_x is translated to c_m. As illustrated in Figure 6 (a), α is the counter-clockwise angle for p and increments the corresponding bin in the angle histogram.

Color Histogram stores the three-dimensional vector $c_i = (h_i, s_i, v_i)$ for all of the pixels p_i belonging to an object. Since the color information is quantized, the color histogram has to store 18 hue, 3 saturation, 3 value levels as well as 4 gray levels.



(a) (b)

Figure 6: (a) Angle Histogram Calculations (b) Quantization Principles and Equivalent Pixel Classes

The two histograms that store shape information are filled with respect to the center of mass (c_m) of the object*. Since the center of mass is a physical property of the object, it is favorable to choose c_m as a reference instead of other points. Obviously, there are more reasons for selecting c_m as a pivot as explained below:

- The upper bound of the distance histogram, d_u, is limited to a reasonable number due to the fact that the MBR encloses the object with four perpendicular tangents.

$$0 \leq d_u < \sqrt{\max(w^2 + h^2)}, \tag{8}$$

*While considering the center of mass of objects, the pixels are treated as equally-weighted; in other words the pixels have equal masses.

where w and h are the width and height of the MBR, respectively. For most of the objects, c_m is not far from the center of the MBR and the upper bound of d_u in Equation (8) is very close to $\frac{\sqrt{\max(w^2+h^2)}}{2}$.

- The angle α_i for a pixel p_i is stored in the angle histogram and with a proper quantization scheme, the number of bins in the angle histogram is very close to that of the distance histogram ($0 \leq \alpha_i \leq 2\pi$).

- The location of c_m of an object does not depend on the position and the orientation of the object. Since the coordinates of c_m can be computable in one pass on the pixels of the objects, the color and shape information is correctly referenced by c_m of the object. For an object o, $c_m = (x_{c_m}, y_{c_m})$ can be computed as follows:

$$x_{c_m} = \frac{\sum_i^n x_i}{n},\ y_{c_m} = \frac{\sum_i^n y_i}{n}, \tag{9}$$

  where n is the number of pixels in o.

The histogram-based approach takes all the pixels into consideration while gathering color and shape information about the object. All of the pixels belonging to an object participate in the overall information of the object. This way of approaching to the problem is very close to the human eye perception since the overall information that the eye captures from the image is nothing but the relative positioning and color variance among the pixels. In this method, the relative positioning among pixels encapsulates the shape information in angle and distance histograms. Similarly, the color variance among the pixels is stored in the color histogram.

Definition 4.1 *(Equivalent Pixel Class) The set of pixels $S_{H[i]}$ forms an equivalent pixel class for an histogram bin $H[i]$ such that for each $p_i \in H[i]$, $p_i \in S_{H[i]}$.*

Since we have three histograms, a pixel p_i participates in three different *equivalent pixel classes*. The situation is slightly different for color histograms. The equivalent pixel classes for color histograms contain the pixels painted with the same color in the object region. Besides, the equivalent pixel classes for angle and distance histograms form contiguous regions, but color histograms may not. As illustrated in Figure 6 (b), the triangular region R_1 is one of the angle histogram equivalent pixel classes. Similarly, the donut-like region R_2 is an equivalent pixel class for the distance histogram.

The angle and distance histograms are quantized and this leads to the fact that relatively smaller-sized histograms are enough for the purpose. For example, the angle histogram has 36 bins with a swapping angle of 10 degrees. (In Figure 6 (b), the $\widehat{c_m}$ angle in R_1 represents the angle quantization value.) While the histograms are formed, each sampled image is re-scaled to at most 200×200 pixels. Thus, the number of bins that the distance histogram stores is no more than that of the color histogram. The quantization values for the histograms are determined by a reasonable number of experiments and evaluations based on the decision to select the best that reflects the capabilities of the human visual system. In Figure 7, the three histograms constructed for an object are given. As shown in (a), the width and height of the minimum bounding rectangle (MBR) of the objects are 103 and 66, respectively. Since the c_m of the object is very close to the center of the MBR (c_{MBR})[†], the maximum distance between a point p and c_m is 54. The distance histogram is quantized with the value 3. Since the object has 6 color regions, the color histogram has 6 bins constituting the color regions. Similarly, the angle histogram is quantized with the value 10 and has 36 bins.

From the mathematical perspective, the method forms a five-dimensional vector v_p for a pixel p containing color and shape information as discussed above. Specifically, the three dimensions of the color information form the first three dimensions of v_p. Since we use two histograms for shape, namely distance and angle histograms, they provide information for the last two dimensions of v_p, respectively ($v_p = (h, s, v, d, a)$).

Lemma 4.1 *Shifting the angle histogram of an object one bin left or right specifies a different orientation for the object.*

Proof: Since angles of the pixels are stored in the angle histogram, shifting the histogram one bin designates rotating all of the pixels in counter-clockwise or clockwise direction with a degree of angle quantization value α (e.g., $\alpha = 10$ for the object in Fig. 7), thus specifies a different orientation. $\square$

Lemma 4.2 *The histogram-based approach for shape is invariant under (i) rotation, (ii) scale, and (iii) translation.*

Proof: (i) The color histogram stores color information and color is an orientation-independent property for the object. Thus, the color histogram does not change for different orientations.

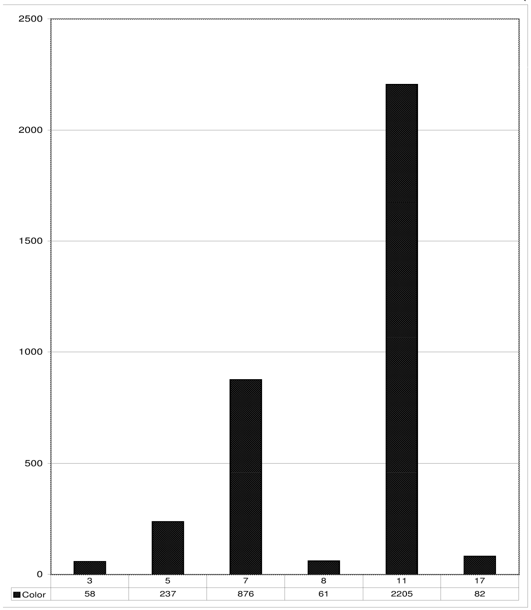
The distance histogram stores the *Euclidean* distance between two pixels (p_i and c_m). The locations of both p_i and c_m vary but the distance between them is preserved for a different orientation of the object.

The angle histogram stores exactly the actual orientation of the object. Due to Lemma 4.1, shifting the angle histogram one bin at a time produces a different orientation. For the similarity distance, all of the orientations have to be considered and this can be achieved

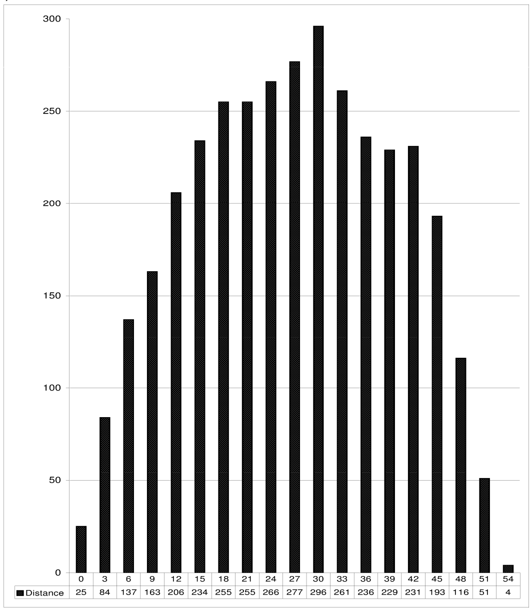
[†]For the object, c_m is at (90,57) and c_{MBR} is at (91,61) coordinates on the image.

`Object Details:` $Area = 3519$ pixels, $MBR_{Width} = 103$ pixels, $MBR_{Height} = 66$ pixels

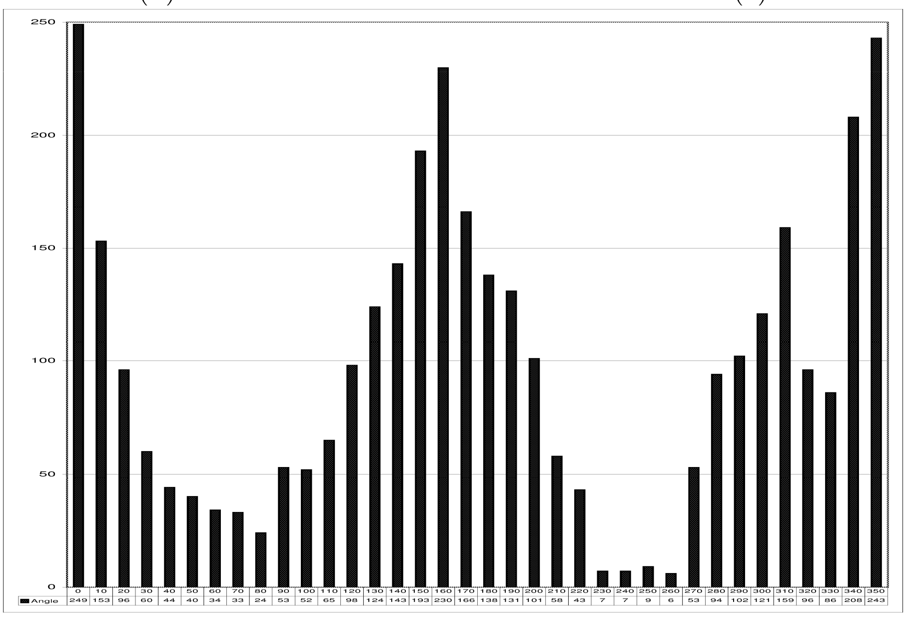
(a)



(b)



(c)



(d)

Figure 7: Histogram-Based Approach for the `tiger` object in Fig. 3. (a) Object details, (b) color histogram, (c) distance histogram, (d) angle histogram.

via shifting the angle histogram n_a times to the left or right, where n_a is the number of bins in the angle histogram. The similarity distance for the angle histogram is the maximum value among these orientations.

(ii) All of the histograms store quantity (i.e., the amount of pixels) thus to satisfy scale invariance, the histograms should be scaled. By the help of the area A_o of the object (i.e., the total number of pixels in the object), it is possible to achieve scale invariance. Every bin in the histograms is divided by A_o and that leads to the scaling of A_o to 1.

(iii) The entities stored in the histograms are location-independent. Thus, the method is translation invariant. □

It is shown by Lemma 4.2 that the histogram-based image is invariant under translation, scale and rotation. Therefore, the approach is expressive in querying by shape.

5 Integrating Object-Based Color and Shape Information

In this section, two approaches to integrate the color and shape content of objects for querying are discussed. The basic comparison algorithms to perform the similarity value calculations between the histograms of the query object and a database object are presented. Specifically, histogram-wise comparisons are made by histogram intersection method due to its simplicity and effectiveness. Figure 8 presents the pseudo-codes of the similarity algorithms employing histogram intersection method for the distance between query data and database data.

In general, the total similarity between the query object and a database object is determined as follows:

$$S_T = \frac{S_C \times w_C + \frac{S_D + S_A}{2} \times w_S}{w_C + w_S}, \tag{10}$$

where S_C, S_D and S_A denote similarities in color, distance and shape histograms, respectively. The color and shape features are integrated with pre-specified weights w_C and w_S. The two approaches for integrating color and shape can be summarized as follows:

1. Assigning equal weights to color and shape. This approach is easy and fast but for most of the datasets, color and shape features may not be treated equally.

2. Integrating features via assigning different weights. The similarity calculations for each feature is calculated separately, and the average-case accuracies of shape and color features are used as their weights in the integrated calculations [JV96]. For most of the situations, assigning weights following this manner gives more effective results for similarity queries. This approach reflects the characteristics for the dataset more than the first one.

Function ColorSimilarity(C_H^q,C_H^d)
nominator = 0;
qSum = 0;
dSum = 0;
for i = 1 **to** length(C_H^q) **do**
  begin
    qSum = qSum + $C_H^q[i]$;
    dSum = dSum + $C_H^d[i]$;
    nominator = nominator + min($C_H^q[i], C_H^d[i]$);
  end
return nominator / min(qSum,dSum);

(a)

Function DistanceSimilarity(D_H^q,D_H^d)
nominator = 0;
qSum = 0;
dSum = 0;
for i = 1 **to** length(D_H^q) **do**
  begin
    qSum = qSum + $D_H^q[i]$;
    dSum = dSum + $D_H^d[i]$;
    nominator = nominator + min($D_H^q[i], D_H^d[i]$);
  end
return nominator / min(qSum,dSum);

(b)

Function AngleSimilarity(A_H^q,A_H^d)
for j = 1 **to** length(A_H^q) **do**
  nominator[j] = 0;
qSum = 0;
dSum = 0;
for i = 1 **to** length(A_H^q) **do**
  begin
    qSum = qSum + $A_H^q[i]$;
    dSum = dSum + $A_H^d[i]$;
    for j = 1 **to** length(A_H^q) **do**
      nominator[j] = nominator[j] + min($A_H^q[i], A_H^d[j]$);
  end
AngleSimilarity[j] = nominator[j] / min(qSum,dSum);
return $\max_i$(AngleSimilarity[i]);

(c)

Figure 8: Similarity algorithms: (a)color, (b)distance, (c)angle.

The execution time for filling the histograms and querying can be estimated as follows: Let $O[n]$ represent an object with n pixels. Since all of the histograms are filled in one pass on the pixels, histogram filling takes $O(n)$ time. Let n_a, n_d and n_c denote the number of bins in the angle, distance, and color histograms, respectively. Similarly, d_a, d_d and d_c denote the similarity distances for the histograms. In order to compute d_a, the angle histogram has to be shifted n_a times. At each time, histogram intersection method is applied. Thus, d_a is computed in $O(n_a^2)$ time. d_d is computed in $O(n_d)$ time since only one histogram intersection method would be sufficient. The average of d_a and d_d will be the similarity for shape (S_S). Color histogram requires one histogram intersection method and d_c is computed in $O(n_c)$ time.

6 Performance Experiments

6.1 Evaluating Retrieval Effectiveness

In the first set of performance experiments, the histogram-based querying approach is evaluated in terms of retrieval effectiveness. A set of images from various domains including nature photographs, portraits, animal figures, etc. are first passed through Object Extractor. For the experiments, 5×5 octagonal color median filters are applied twice to have appropriate filtered images. On the average, the number of objects per image is 1.2. Having extracted the objects, the shape and color features are integrated for retrieval in two different ways as mentioned. First, they are treated as equally-weighted, thus 0.5 is assigned to each of w_C and w_S. Second, the average-case retrieval accuracy of individual features, shape and color, is assigned to each of them as weights.

In order to evaluate effectiveness of retrieval systems, two well-known metrics, *precision* and *recall*, are used. Precision is the ratio of the number of retrieved images that are relevant to the number of retrieved images. Recall is the ratio of the number of retrieved images that are relevant to the total number of relevant images [Jon81]. Prior to the experiments, the relevance degrees (0,0.5,1) are subjectively assigned to the extracted objects in order to signify the relevance of an object to the query object for each query.

In the experiments, the query object is randomly picked from the extracted objects. In order to improve the evaluation, the retrieval process is performed for three different randomly picked query objects. Then, the effectiveness is evaluated as the average of the results calculated for each query separately. As mentioned in most of the information retrieval systems, in order to facilitate the computation of average of precision and recall values, the individual precision values are interpolated to a set of 11 standard recall levels (0,0.1,0.2,...,1). The results are shown in Figure 9.

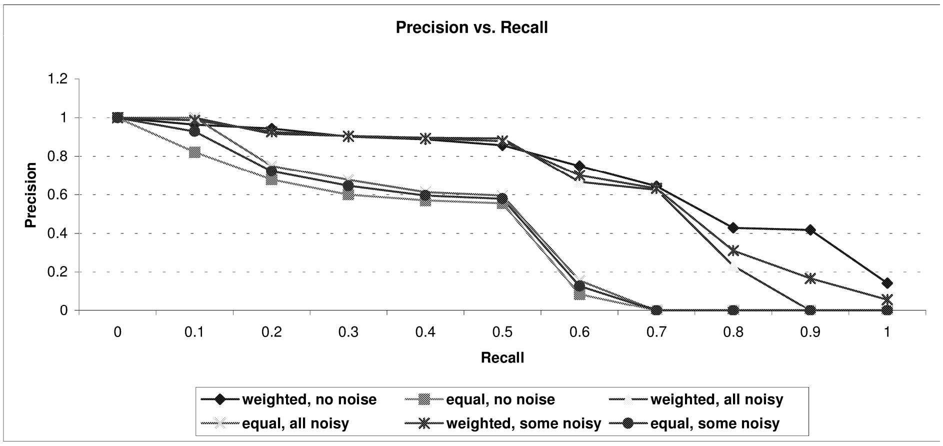


Figure 9: Retrieval effectiveness results.

Random Noise Model. In order to evaluate the effectiveness of the histogram-based approach for noisy objects, we employed a noise model, called Random Noise Model, which alters $k\%$ pixels in a test image via changing the locations of the pixels. This noise model is similar to the one proposed in [JV96] but since the color and shape information stored in the histograms is only dependent on the pixels, exchanging the locations of the $k\%$ pixels directly produces noise for both color and shape. Since only a group of the test objects are augmented noise, the effectiveness results may vary from the ones without noise. The results are shown in Figure 9 with $k = 7$. □

The experiments are repeated with the same set of images in order to show the invariance of the approach under translation, rotation and scale. A group of test objects is re-scaled, another group of test objects is rotated and another group of the test objects is translated in the image. The precision-recall analysis carried out for this case gives exactly the same precision-recall graph as the one in Figure 9.

Another experiment is conducted for evaluating the behavior of the approach to noisy objects. In this experiment, each object is compared with its noisy form. It is shown through the results that the histogram-based approach results in values between 92% and 98% for the test objects. Therefore, it can be concluded that the tool retrieves the noisy forms of the object before any other object, which is expected and desired.

6.2 Experiments with Turning Angle Method

In the experiments of this section, the histogram-based approach is compared with the well-known shape matching method, the *Turning Angle (TA)* method [ACH$^+$91]. The reason of choosing TA method is the fact that it behaves better in most of the cases than the other methods (e.g., moments [TC92], Hausdorrf distance [HKR93], sign of curvature [SAF94]) due to the human perceptual judgments [SAF94]. In order to be fair and consistent, the best model for converting the output of Object Extractor to an input to TA is chosen. Since the histogram-based approach accepts the output of Object Extractor, the experiments evaluate the approach accurately. Since the query-by-color approach is almost the same as all the other histogram-based approaches, no evaluation is made for this part.

6.2.1 Kinematics Model for Polygon Simplification

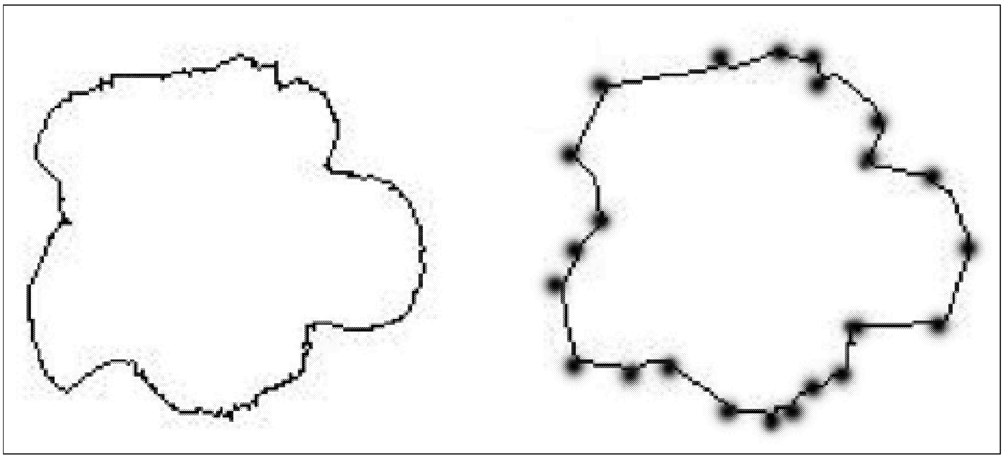
From the Turning Angle (TA) method's point of view, a (polygonal) object is a set of vertices and the shape comparison between two objects is performed based on their turning angle representation. Since objects in images and/or video keyframes are segmented via Object Extractor tool, a method is required to transform the output of Object Extractor to the input of TA method. With this motivation, *Kinematics Model for Polygon Simplification* (KMPS) is implemented.

The KMPS method processes a 360-gon that represents a boundary polygon of an object. In fact, this boundary polygon is a set of 360 vertices corresponding to angles when calculated by the centroid of an object. Hence, in general the input to the KMPS method is an 360-gon. Then, this input is processed as the following. The velocity $V_p = \frac{\Delta D_p}{\Delta a_p}$ and the acceleration $A_p = \frac{\Delta V_p}{\Delta a_p}$ are calculated for each boundary pixel p where D_p is the distance between centroid and p, a_p is the polar angle of p from centroid (cf., Fig. 6). The boundary points having the largest acceleration is the sharp and extreme points of the boundary of the object. The output of the KMPS method is an N-gon where N is a user-defined parameter. The most accelerated pixels constitute the resultant N points as an output.

Since the output of object extraction process via Object Extractor is certainly an input to this simplification method, KMPS turns out to be the best fit for evaluating the histogram-based approach. Figure 10 demonstrates the KMPS method on a sample object. In Figure 10 (a), the original object is shown as a polygon of 360 vertices. After applying KMPS method on the object, the object is simplified to 23 vertices as shown in Figure 10 (b).

6.2.2 Noise Model and Results

In determining the similarity value between two objects, the trivial but the most comprehensive way is setting 1 to equality and 0 to the least similarity. Thus, when the similarity value is closer to 1, the objects resemble each other. In the histogram-based approach, this way of similarity determination is employed. However in TA method, the way of determining similarity value is different. In the TA method, values less than 0.5 correspond to



(a) (b)

Figure 10: Application of KMPS on an input polygon. (a) Original polygon having 360 vertices (at each degree). (b) Polygon after simplification to 23 vertices.

objects resembling each other. Hence, the similarity values presented in the experiments have to be considered in the light of these discussions.

Since the TA method has drawbacks with 'noisy' objects, the main point in the experiments is to demonstrate the effectiveness of the histogram-based approach with noisy objects. For this reason, the similarity values between two versions of the objects are evaluated. The first version is the original object and the second one is the modified version after discarding relatively small regions from the object region. The modifications on the test objects are based on Boundary Random Noise Model. Table 1 presents the average retrieval accuracy of single object queries such that each query object (in original version) is queried with all of the objects in the noisy version. It is shown that the histogram-based approach retrieves the noisy version of the query object before any other object.

Table 1: Average retrieval accuracy of the single object queries. HBA is an abbreviation of the histogram-based approach, N denotes the retrieval rank of the noisy version of the query object.

	N = 1	N < 4	N < 6	N < 10
HBA	1.00	1.00	1.00	1.00
TA	0.86	0.95	1.00	1.00

Boundary Random Noise Model. This random noise model alters k% pixels on the boundary of a given object. The noisy objects produced via this model is used to evaluate the shape retrieval effectiveness of the histogram-based approach. The comparisons are made with the TA method which deals with the object boundaries of objects for shape retrieval. In the experiments, k is set to 7. □

It is known that noise on the objects has a crucial impact on the performance of the shape similarity methods. However, the histogram-based approach provides a promising result such that the embedded noise on the object can distort the similarity at most 2.7%. For the TA method, the embedded noise affects the similarity value significantly. In order to have a better understanding for the impact of the noise, the following similarity values gathered from the TA method have to be given. The TA method gives the value 1.30 for the similarity between the objects `tiger` and `rose`; 1.05 for `aircraft` and `rose` with the original versions of the objects. Interestingly, the similarity between the original and noisy versions of `aircraft` is 1.26. The similarity value between the two versions of `tiger` is 1.48. Thus, for the objects `aircraft` and `tiger`, `rose` object is more similar to them than their noisy versions, which is unexpected. The histogram-based approach has the following similarity values for these three objects: between the two versions of `aircraft`, 0.98; between the two versions of `tiger`, 0.98; between `rose` and `aircraft`, 0.78; between `rose` and `tiger`, 0,72. Thus, it can be concluded that the histogram-based approach overcomes most of the drawbacks of the existing methods caused by the noise on the object boundaries.

7 Conclusion

In this paper, a novel approach for object-based querying by the features shape and color is proposed. The approach is histogram-based in which one color histogram and two shape histograms are stored for an object. The approach is scalable in the sense that the images having more than one object can be processed without any drawback.

The main idea of employing histogram-based approach is to include the contributions of each pixel in the object region to the color and shape information of the object. Thus, the drawbacks of the existing systems are overcome because most of those systems consider only the boundaries of the objects. The object-based information can be queried without any pre-processing phase including edge-detection since the output of an extraction tool, Object Extractor, is used. The output contains the necessary data to construct the histograms for the object. It is shown that the histogram-based approach is invariant under rotation, translation and scale, thus the approach is orientation-independent for the extracted objects.

The precision-recall analysis is performed to evaluate the retrieval effectiveness of the histogram-based approach. Moreover, the performance experiments show that the approach gives better results in most of the cases than the turning angle method, which is known to be one of the best shape comparison techniques in the literature.

References

[ACH+91] E. Arkin, P. Chew, D. Huttenlocher, K. Kedem, and J. Mitchel. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209–215, 1991.

[Can86] J. Canny. A computational approach to edge-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, November 1986.

[CCM+97] S.F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. VideoQ: an automated content based video search system using visual cues. In *ACM Multimedia'97 Conference Proceedings*, pages 313–324, Seattle, USA, 1997.

[CTB+99] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proceedings of Third International Conference on Visual Information Systems*. Springer, 1999.

[DUG00] M.E. Dönderler, Ö. Ulusoy, and U. Güdükbay. A rule-based approach to represent spatio-temporal relation in video data. In Tatyana M. Yakhno, editor, *Proceedings of Advances in Information Systems ADVIS 2000, Izmir, Turkey*, volume 1909 of *LNCS*, pages 409–418. Springer-Verlag, 2000.

[FEF+94] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack, D. Petrovic, and R. Barber. Efficient and effective querying by image content. *Intelligent Information Systems*, 3(4):231–262, 1994.

[FSN+95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petrovic, D. Steele, and P. Yanker. Query by image content the QBIC system. In *IEEE Computer Magazine*, volume 28, September 1995.

[GT98] B. Günsel and A.M. Tekalp. Shape similarity matching for query-by-example. *Pattern Recognition*, 31(7):931–944, 1998.

[HB94] D. Hearn and M.P. Baker. *Computer Graphics*. Prentice Hall, Inc., 1994.

[HKM+97] J. Huang, S.R. Kumar, M. Mitra, W.J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Proceedings of IEEE Comp. Soc. Conf. Comp. Vis. and Patt. Rec.*, pages 762–768, 1997.

[HKR93] D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. Comparing images using the Haussdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, September 1993.

[HSE+95] J. Hafner, H.S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[Jon81] K.S. Jones. *Information Retrieval Experiment.* Butterworth and Co., 1981.

[JV96] A.K. Jain and A. Vailaya. Image retrieval using color and shape. *Pattern Recognition*, 29(8):1233–1244, August 1996.

[PPS94] A. Pentland, R.W. Picard, and S. Scarloff. Photobook: tools for content-based manipulation of image databases. In *Proceedings of Storage and Retrieval for Image and Video Databases II, SPIE*, volume 2, pages 34–47, 1994.

[Rus99] J. Russ. *The Image Processing Handbook.* CRC Press, in cooperation with IEEE Press, 1999.

[SAF94] B. Scassellati, S. Alexopoulos, and M. Flickner. Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments. In *Proceedings of Storage and Retrieval for Image and Video Databases II, SPIE*, volume 2, 1994.

[SB91] M.J. Swain and D.H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SC96a] J.R. Smith and S.F. Chang. Tools and techniques for color image retrieval. In I.K. Jain and R.C., editors, *Proceedings of Storage and Retrieval for Image and Video Databases IV, IS&T/SPIE*, volume 2670, pages 426–437, 1996.

[SC96b] J.R. Smith and S.F. Chang. VisualSEEk: A fully automated content-based image query system. In *ACM Multimedia'96 Conference Proceedings*, pages 87–98, 1996.

[ŞGU01] E. Şaykol, U. Güdükbay, and Ö. Ulusoy. A semi-automatic object extraction tool for querying in multimedia databases. In S. Adali and S. Tripathi, editors, *7th Workshop on Multimedia Information Systems MIS'01, Capri, Italy*, pages 11–20, November 2001.

[SP93] S. Scarloff and A. Pentland. A finite-element framework for correspondance and matching. In *Proceddings of Fourth International Conference on Computer Vision*, pages 308–313, 1993.

[SSS00] M. Safar, C. Shahabi, and X. Sun. Image retrieval by shape: A comperative study. In *Proceedings of IEEE International Conference on Multimedia and Expo (I)*, pages 141–154, 2000.

[TC92] G. Taubin and D.B. Cooper. Object recognition based on moment (or algebraic) invariants, 1992.

[Vel01] R.C. Veltkamp. Shape matching: Similarity measures and algorithms. Technical Report UU-CS-2001-03, Utrecht University, January 2001.

[Zlo77] M.M. Zloof. Query-by-example: a database language. *IBM Syst. Journal*, 16(4), 1977.