

# A Robust Color Object Analysis Approach to Efficient Image Retrieval

**Ruofei Zhang**

*Department of Computer Science, State University of New York, Binghamton, NY 13902, USA*  
*Email: rzhang@binghamton.edu*

**Zhongfei (Mark) Zhang**

*Department of Computer Science, State University of New York, Binghamton, NY 13902, USA*  
*Email: zhongfei@cs.binghamton.edu*

*Received 20 December 2002; Revised 1 December 2003*

We describe a novel indexing and retrieval methodology integrating color, texture, and shape information for content-based image retrieval in image databases. This methodology, we call CLEAR, applies unsupervised image segmentation to partition an image into a set of objects. Fuzzy color histogram, fuzzy texture, and fuzzy shape properties of each object are then calculated to be its signature. The fuzzification procedures effectively resolve the recognition uncertainty stemming from color quantization and human perception of colors. At the same time, the fuzzy scheme incorporates segmentation-related uncertainties into the retrieval algorithm. An adaptive and effective measure for the overall similarity between images is developed by integrating properties of all the objects in every image. In an effort to further improve the retrieval efficiency, a secondary clustering technique is developed and employed, which significantly saves query processing time without compromising retrieval precision. A prototypical system of CLEAR, we developed, demonstrated the promising retrieval performance and robustness in color variations and segmentation-related uncertainties for a test database containing 10 000 general-purpose color images, as compared with its peer systems in the literature.

**Keywords and phrases:** content-based image retrieval, fuzzy logic, region-based features, object analysis, clustering, efficiency.

## 1. INTRODUCTION

The dramatic improvements in hardware technology have made it possible in the last few years to process, store, and retrieve huge amount of data in image databases. Initial attempts to manage pictorial documents relied on textual description provided by a human operator. This time-consuming approach rarely captures the richness of visual content of the images. For this reason researchers have focused on the automatic extraction of the visual content of images to enable indexing and retrieval, in other word, content-based image retrieval (CBIR). CBIR is aimed at efficient retrieval of relevant images from large image databases based on automatically derived features. These features are typically extracted from shape, texture, and/or color properties of query image and images in the database. The relevancies between a query image and images in the database are ranked according to a similarity measure computed from the features.

In this paper we describe an efficient clustering-based fuzzy feature representation approach—clustering-based ef-

ficient automatic region analysis technique, as we conveniently named CLEAR, to address general purposed CBIR. We integrate semantic-intensive clustering-based segmentation with fuzzy representation of color histogram, texture, and shape to index image databases. A low computational yet robust distance metric is developed to reduce the query time of the system. The response speed is further improved significantly by using a novel secondary clustering technique to achieve high scalability for large image databases. An overview of the architecture of the proposed approach is shown in Figure 1.

The remainder of this paper is organized as follows. In Section 2, we provide a review of related work. Section 3 describes our clustering-based procedure. First, the unsupervised image segmentation by applying clustering method based on color and texture is described in Section 3.1. Then we give the definition of the fuzzy color histogram and fuzzy feature representation reflecting texture and shape properties of each region in Sections 3.2 and 3.3, respectively. The distance metric and comprehensive similarity calculation based on region-pair distance are provided in Section 4. The



pixel is of color  $c_j$ . Rao et al. [7] generalized the color spatial distribution measurements by counting the color histogram with certain geometric relationships between pixels of particular colors. It extends the spatial distribution comparison of color histogram classes. Another histogram refinement approach is given by Cinque et al. [8]. They recorded the average position of each color histogram and their standard deviation to add some kind of spatial information on traditional histogram approach. Despite the improvement efforts, these histogram refinements did not handle the inaccuracy of color quantization and human perception of colors, so the calculation of color histogram itself was inherently not refined. Apart from color histogram, other feature-extracting techniques have been tried in different ways. Ravela and Manmatha [9] used a description of the image intensity surface to be signatures. Gaussian derivative filters at several scales were applied to the image and low-order 2D differential invariants are computed to be features compared between images. In their system, users selected appropriate regions to submit a query. The invariant vectors corresponding to these regions were matched with the database counterparts both in feature and coordinate spaces to yield a match score per image. The features extracted in [9] have higher detail-depicting performance than color histogram to describe the content of one image. But this approach was time consuming and required about 6 minutes to retrieve one image.

All the above cited global-feature-based approaches share one common limit: they handle low-level semantic queries only. They are not able to identify object-level differences, so they are not semantic-related and their performance is limited.

Region-feature-based approach is an alternative in CBIR. Berkeley Blobworld [12], UCSB NeTra [13], Columbia VisualSEEK [14], and Stanford IRM [15] are representative ones. A region-based retrieval system segments images into regions (objects), and retrieves images based on the similarity between regions. Berkeley Blobworld [12] and UCSB NeTra [13] compare images based on *individual* regions. To query an image, the user was required to select regions and the corresponding features to evaluate similarly. Columbia VisualSEEK [14] partitioned an image in regions using a sequential labeling algorithm based on the selection of a single color or a group of colors, called color set. For each region, they computed a binary color set using histogram back projection. These individual-region-distance-based systems have some common drawbacks. For example, they all have complex interface and need the user's prequery interaction, which places additional burden on the user, especially when the user is not a professional image analyst. In addition, little attention has been paid to the development of similarity measures that integrate information from all of the regions. To address some of these drawbacks, Wang et al. [15] recently proposed an integrated region matching scheme called IRM for CBIR. They allowed for matching a region in one image to several regions of another image; as a result the similarity between the two images was defined as the weighed sum of distances, in the feature space, between all regions from

different images. Compared with retrieval systems based on individual regions, this scheme reduces the impact of inaccurate segmentation by smoothing over the imprecision in distance. Nevertheless, the representation of properties for each region is simple and inaccurate so that most feature information of a region is nullified. In addition, it fails to explicitly express the uncertainties (or inaccuracies) in the signature extraction; meanwhile, the weight assign scheme is very complicated and computationally intensive. Later, Chen and Wang [16] proposed an improved approach called UFM based on applying "coarse" fuzzy model to the region features to improve the retrieval effectiveness of IRM. Although the robustness of the method is improved, the drawbacks existing in the previous work [15] were not alleviated. Recently Jing et al. [17] presented a region-based modified inverted file structure analogous to that in text retrieval to index the image database; each entry of the file corresponds to a cluster (called codeword) in the region space. While Jing's method is reported to be effective, the selection of the size of the code book is subjective in nature, and the retrieval effectiveness is sensitive to this selection.

To narrow the gap between content and semantics of images, some lately reported works in CBIR, such as [18, 19], performed the image retrieval not only based on contents but also heavily based on user *preference profiles*. Machine learning techniques such as support vector machine (SVM) [20] and Bayes network [21] were applied to learn the user's query intention through leveraging preference profiles or relevance feedbacks. One drawback of such approaches is that they work fine only for one specific domain, for example, art image database or medical image database. It has been shown that for a general domain, the retrieval accuracy of these approaches are weak. In addition, these approaches are restricted by the availability of user preference profiles and the generalization limitation of machine learning techniques they applied.

The objective of CLEAR is three-fold. First, we intended to apply pattern recognition techniques to connect low-level features to high-level semantics. Therefore, our approach also falls into the region-feature-based category, as opposed to indexing images in the whole image domain. Second, we intended to address the color "inaccuracy" and image segmentation-related uncertainty issues typically found in color image retrieval in the literature. With this consideration, we applied fuzzy logic to the system. Third, we intended to improve the query processing time to avoid the typical linear search problem in the literature; this drove us to develop the secondary clustering technique currently employed in the prototype system CLEAR. As a result, compared with the existing techniques and systems, CLEAR has the following distinctive advantages: (i) it partially solves the problem of the color inaccuracy and texture (shape) representation uncertainty typically existing in color CBIR systems, (ii) it develops a balanced scheme in similarity measure between regional and global matching, and (iii) it "preorganizes" image databases to further improve retrieval efficiency without compromising retrieval effectiveness.

### 3. CLUSTERING-BASED FUZZY MATCHING

We propose an efficient, clustering-based, fuzzified feature representation approach to address the general-purpose CBIR. In this approach we integrate semantic-intensive clustering-based segmentation with fuzzy representation of color histogram, texture, and shape to index image databases.

#### 3.1. Image segmentation

In our system, the query image and all images in the database are first segmented into regions. The fuzzy feature of color, texture, and shape are extracted to be the signature of each region in one image. The image segmentation is based on color and spatial variation features using  $k$ -means algorithm [22]. We chose this algorithm to perform the image segmentation because it is unsupervised and efficient, which is crucial to segment general-purpose images such as the images on the World Wide Web.

To segment an image, the system first partitions the image into blocks with  $4 \times 4$  pixels to compromise between texture effectiveness and computation time, then extracts a feature vector consisting of six features from each block. Three of them are average color components in a  $4 \times 4$  pixel size block. We use the CIELAB color space because of its desired property that the perceptual color difference is proportional to the numerical difference. These features are denoted as  $\{C_1, C_2, C_3\}$ . The other three features represent energy in the high-frequency bands of the Haar wavelet transform [23], that is, the square root of the second-order moment of wavelet coefficients in high-frequency bands. To obtain these moments, a Haar wavelet transform is applied to the  $L$  component of each pixel. After a one-level wavelet transform, a  $4 \times 4$  block is decomposed into four frequency bands; each band contains  $2 \times 2$  coefficients. Without loss of generality, suppose the coefficients in the HL band are  $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$ . Then we compute one feature of this block in HL band as

$$f = \left( \frac{1}{4} \sum_{i=0}^1 \sum_{j=0}^1 c_{k+i,l+j}^2 \right)^{1/2}. \quad (1)$$

The other two features are computed similarly from the LH and HH bands. These three features of the block are denoted as  $\{T_1, T_2, T_3\}$ . They can be used to discern texture by showing  $L$  variations in different directions.

After we obtain feature vectors for all blocks, we perform normalization on both color and texture features to whiten them, so the effects of different feature range are eliminated. Then the  $k$ -means algorithm [22] is used to cluster the feature vectors into several classes with each class corresponding to one region in the segmented image. Because clustering is performed in the feature space, blocks in each cluster do not necessarily form a connected region in the image. This way, we preserve the natural clustering of objects in general-purpose images. The  $k$ -means algorithm does not specify how many clusters to choose. We adaptively select the number of clusters  $C$  by gradually increasing  $C$  until a stop criterion is met. The average number of clusters for all images

in the database changes in according with the adjustment of the stop criteria. In the  $k$ -means algorithm we use a color-texture weighted  $L2$  distance metric

$$\sqrt{w_c \sum_{i=1}^3 (C_i^{(1)} - C_i^{(2)})^2 + w_t \sum_{i=1}^3 (T_i^{(1)} - T_i^{(2)})^2} \quad (2)$$

to describe the distance between block features, where the  $C^{(1)}(C^{(2)})$  and  $T^{(1)}(T^{(2)})$  are color features and texture features, respectively, of the two blocks. At this time, we set weight  $w_c = 0.65$  and  $w_t = 0.35$  based on the trial-and-error experiments. Color property is assigned more weight because of the effectiveness of color to describe the image and the relative simple description of texture features.

After segmentation, three additional features are calculated for each region to describe shape property. They are normalized inertia [24] of order 1 to 3. For a region  $H$  in 2-dimensional Euclidean integer space  $Z^2$  (an image), its normalized inertia of order  $p$  is

$$l(H, p) = \frac{\sum_{(x,y): (x,y) \in H} [(x - \hat{x})^2 + (y - \hat{y})^2]^{p/2}}{[V(H)]^{1+p/2}}, \quad (3)$$

where  $V(H)$  is the number of pixels in the region  $H$  and  $(\hat{x}, \hat{y})$  is the centroid of  $H$ . The minimum normalized inertia is achieved by spheres. Denoting the  $p$ th order normalized inertia of spheres as  $L_p$ , we define the following features to describe the shape of each region:

$$\begin{aligned} S_1 &= \frac{l(H, 1)}{L_1}, & S_2 &= \frac{l(H, 2)}{L_2}, \\ S_3 &= \frac{l(H, 3)}{L_3}. \end{aligned} \quad (4)$$

#### 3.2. Fuzzy color histogram for each region

The color representation would be coarse and imprecise if we simply extract color feature of one block (the representative block) to be the color signature of each region as Wang et al. [15] did. Color is one of the most fundamental properties to discriminate images, so we should take advantage of all available information in it. Taking the uncertainty stemmed from color quantization and human perception of colors into consideration, we devised a modified color histogram descriptor utilizing the fuzzy technique [25, 26] to handle the fuzzy nature of colors in each region. The reason we treat color property this way is two-fold: (i) we want to characterize the local property of colors precisely and robustly and (ii) color component in the region features is extracted more accurate than texture and shape and it is more reliable to describe the semantics of images.

In our color descriptor, fuzzy paradigm-based techniques [27] are applied to the color distribution in each region. The key point is that we assume each color is a fuzzy set while the correlation among colors are modeled as membership functions of fuzzy sets. A fuzzy set  $F$  on the feature space  $R^n$  is defined by a mapping  $\mu_F : R^n \rightarrow [0, 1]$  named the *membership*

function. For any feature vector  $f \in R^n$ , the value of  $\mu_F(f)$  is called the degree of membership of  $f$  to the fuzzy set  $F$  (or, in short, the degree of membership to  $F$ ). A value closer to 1 for  $\mu_F(f)$  means more representative the feature vector  $f$  to the fuzzy set  $F$ . For a fuzzy set  $F$ , there is a smooth transition for the degree of membership to  $F$  besides the hard cases  $f \in F$  ( $\mu_F(f) = 1$ ) and  $f \notin F$  ( $\mu_F(f) = 0$ ). It is clear that a fuzzy set degenerates to a conventional set if the range of  $\mu_F$  is  $\{0, 1\}$  instead of  $[0, 1]$  ( $\mu_F$  is then called the *characteristic function* of the set). Readers are referred to [28] for more fundamentals of fuzzy set.

The fuzzy model of color descriptor we choose should admit that the resemblance degree decreases as the intercolor distance increases. The natural choice, according to the image processing techniques, is to impose a smooth decay of the resemblance function with respect to the intercolor distance. As we pointed out above, the LAB color space is supposed to offer the equivalence between the perceptual intercolor distance and the Euclidean distance between their coordinate representations. Practical considerations and the analytical simplification of the computational expressions demand the use of a unified formula for the resemblance degree function (equivalent to the membership function). A formula with linear descent would require little computation but could contradict the smooth descent principle. The most commonly used prototype membership functions are cone, trapezoidal,  $B$ -splines, exponential, Cauchy, and paired sigmoid functions [29]. Since we could not think of any intrinsic reason why one should be preferred to any other, we tested the cone, trapezoidal, exponential, and Cauchy functions on our system. In general, the performance of the exponential and the Cauchy functions is better than that of the cone and trapezoidal functions. Considering the computational complexity, we pick the Cauchy functions because it requires much less computations. The Cauchy function,  $C : R^n \rightarrow [0, 1]$ , is defined as

$$C(\vec{x}) = \frac{1}{1 + (\|\vec{x} - \vec{v}\|/d)^\alpha}, \quad (5)$$

where  $\vec{v} \in R^n$ ,  $d, \alpha \in R$ ,  $d > 0$ ,  $\alpha \geq 0$ ,  $\vec{v}$  is the center location (point) of the fuzzy set,  $d$  represents the width of the function, and  $\alpha$  determines the shape (or smoothness) of the function. Collectively,  $d$  and  $\alpha$  describe the grade of fuzziness of the corresponding fuzzy feature. Figure 2 illustrates the Cauchy function in  $R$  with  $v = 0$ ,  $d = 36$ , and  $\alpha$  varying from 0.01 to 100. As we can see, the Cauchy function approaches the characteristic function of open interval  $(-36, 36)$  when  $\alpha$  goes to positive infinity. When  $\alpha$  equals 0, the degree of membership for any element in  $R$  (except 0 whose degree of membership is always 1 in this example) is 0.5.

Accordingly, the color resemblance in a region is defined as

$$\mu_c(c') = \frac{1}{1 + (d(c, c')/\sigma)^\alpha}, \quad (6)$$

where  $d$  is the Euclidean distance between color  $c$  and  $c'$  in

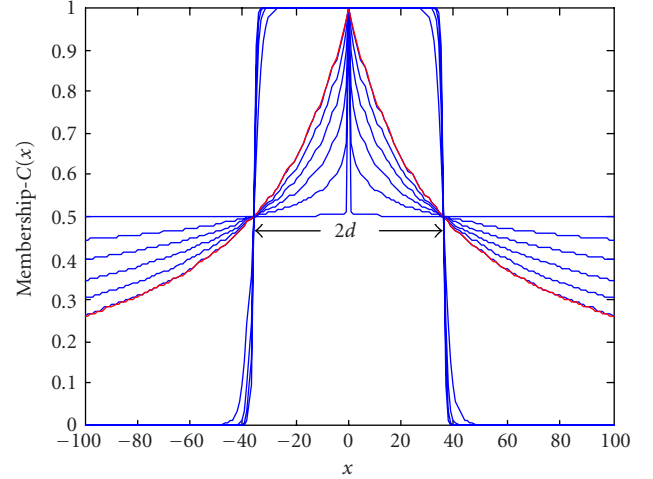


FIGURE 2: Cauchy functions in one dimension.

LAB space and  $\sigma$  is the average distance between colors,

$$\sigma = \frac{2}{B(B-1)} \sum_{i=1}^{B-1} \sum_{k=i+1}^B d(c, c'), \quad (7)$$

where  $B$  is the number of bins in the color partition. The average distance between colors is used to approximate the appropriate width of the fuzzy membership function. The experiments show that the color model performance changes insignificantly when  $\alpha$  is in the interval  $[0.7, 1.5]$ , but degrades rapidly outside the interval. So we set  $\alpha = 1$  in (6) to simplify the computation.

This fuzzy color model enables us to enlarge the influence of a given color to its neighboring colors according to the uncertainty principle and the perceptual similarity. This means that each time a color  $c$  is found in the image, it will influence all the quantized colors according to their resemblance to the color  $c$ . Numerically, this could be expressed as

$$h_2(c) = \sum_{c' \in \mu} h_1(c') \mu_c(c'), \quad (8)$$

where  $\mu$  is the color universe in the image and  $h_1(c')$  is the usual normalized color histogram. Finally the normalized fuzzy color histogram is calculated with

$$h(c) = \frac{h_2(c)}{\max_{c' \in \mu} h_2(c')} \quad (9)$$

which falls in the interval  $[0, 1]$ .

From the signal processing perspective, this fuzzy histogram operation is in fact a linear convolution between the usual color histogram and the fuzzy color model. This convolution expresses the histogram smoothing provided that the color model is indeed a smoothing, low-pass filtering kernel. The use of the Cauchy shape form as color model produces the smoothed histogram, which is a mean for the reduction of quantization errors [30].



In our system, the LAB color space is quantized into 96 bins by using uniform quantization ( $L$  by 6,  $A$  by 4, and  $B$  by 4). Then formula (9) is used to calculate the fuzzy histogram for each region. To reduce the online computation,  $\mu_c(c')$  for each bin is precomputed and implemented as a lookup table.

### 3.3. Fuzzy representation of texture and shape for each region

To accommodate the imprecise image segmentation and uncertainty of human perception, we propose to fuzzify each region generated from image segmentation by a fixed parameterized membership function. The parameter for the membership functions is calculated using the clustering results. The fuzzification of feature vectors brings in a crucial improvement on the region representation of an image: fuzzy features naturally characterize the gradual transition between regions within an image. In our proposed representation scheme, a fuzzy feature set assigns weights, called degree of membership, to feature vectors of each block in the feature space. As a result, feature vector of a block usually belongs to multiple regions with different degrees of membership as opposed to the classical region representation, in which a feature vector belongs to exactly one region. This fuzzification technique has two major advantages: (i) it makes the retrieval system more accurate and robust to image alterations such as intensity variation, color distortion, shape distortion, and so forth, (ii) it better extracts useful information under the same uncertain conditions, that is, it is more robust to imprecise segmentation.

Our approach is to treat each region as a fuzzy set of blocks. To make our fuzzification scheme unified to be consistent with the fuzzy color histogram representation, we again use the Cauchy function to be our fuzzy membership function

$$\mu_i(f) = \frac{1}{1 + (d(f, \hat{f}_i)/\sigma)^\alpha}, \quad (10)$$

where  $f \in R^k$  (in our approach,  $k = 3$ ) is the texture feature vector of each block,  $\hat{f}_i$  is the average texture feature vector of region  $i$ ,  $d$  is the Euclidean distance between  $\hat{f}_i$  and any feature  $f$ , and  $\sigma$  represents the average distance for texture features between cluster centers we get from the  $k$ -means algorithm.  $\sigma$  is defined by

$$\sigma = \frac{2}{C(C-1)} \sum_{i=1}^{C-1} \sum_{k=i+1}^C \|\hat{f}_i - \hat{f}_k\|, \quad (11)$$

where  $C$  is the number of regions in a segmented image and  $\hat{f}_i$  is the average texture feature vector of region  $i$ .

A region is described as a fuzzy set to which each block has a membership so that a hard segmentation is avoided and the uncertainties stemming from inaccurate image segmentation is addressed explicitly.

Accordingly, by making use of this block membership functions, the fuzzified texture properties of region  $i$  is represented as

resented as

$$\hat{f}_i^T = \sum_{f \in U^T} f \mu_i(f), \quad (12)$$

where  $U^T$  is the feature space composed by texture features of all blocks.

Based on the fuzzy membership function  $\mu_i(f)$  obtained in a similar fashion, we also fuzzify the shape property representation of region  $i$  by modifying (3) as

$$l(i, p) = \frac{\sum_{f \in U^S} [(f_x - \hat{x})^2 + (f_y - \hat{y})^2]^{p/2} \mu_i(f)}{[N]^{1+p/2}}, \quad (13)$$

where  $N$  is the number of blocks in an image and  $U^S$  is the block feature space in an image. Based on (4) and (13), we calculate the fuzzified shape feature  $\hat{f}_i^S \equiv \{S1, S2, S3\}$  of each region.

## 4. REGION MATCHING AND SIMILARITY CALCULATION

Now we have fuzzy histogram representation (9) to characterize color property, while the texture and shape properties are characterized by fuzzy features  $\hat{f}_i^T$  and  $\hat{f}_i^S$ , respectively, for each region. To eliminate the effect of different ranges, we apply normalization on these features before they are written to the index files. As a summary, for each region, we record following information to be its indexed feature: (1) fuzzy color histogram  $h(c)$ ; (2) fuzzy texture feature  $\vec{f}^T$ ; (3) fuzzy shape feature  $\vec{f}^S$ ; (4) the relative size of the region to the whole image  $w$ ; and (5) the central coordinate of the region area  $(\hat{x}, \hat{y})$ .

For an image in the database, such information of all regions in the image is recorded as the signature of the image.

Based on these fuzzified features for regions in every image, a fuzzy matching scheme is developed to calculate the distance between any two regions  $p$  and  $q$ ; and the overall similarity measurement between images is derived.

For fuzzy texture and shape features, we apply the  $L2$  distance formula as

$$\begin{aligned} d_T^{pq} &= \|f_p^T - f_q^T\|, \\ d_S^{pq} &= \|f_p^S - f_q^S\|, \end{aligned} \quad (14)$$

respectively.

For fuzzy histogram, we use the distance formula as

$$d_C^{pq} = \sqrt{\frac{\sum_{i=1}^B [h_p(i) - h_q(i)]^2}{B}}, \quad (15)$$

where  $B$  is the number of bins, 96 in our system, and  $h_p(i)$  and  $h_q(i)$  are fuzzy histograms of regions  $p$  and  $q$ , respectively.

The intercluster distance on color and texture between regions  $p$  and  $q$  is depicted as

$$d_{CT}^{pq} = \sqrt{d_C^{pq^2} + d_T^{pq^2}}. \quad (16)$$

The comprehensive distance between the two regions is defined as

$$\text{DIST}(p, q) = wd_{CT}^{pq} + (1 - w)d_S^{pq}. \quad (17)$$

We set  $w$  at 0.7 in our system. Since all components are normalized, this comprehensive distance between the two regions is also normalized. The reason for setting  $w$  at 0.7 stems from the fact that we find some images to be object-dependent in the testing image database, such as animals and plants. However some other images, such as scenic images comprising of land, sea water, or mountains, have shape component that vary widely between the images of the same semantics. This can cause the retrieval engine to return false positives. Note that object-based images tend to have a certain similarity in their color-texture structure generally, in the sense that their color-texture scheme does not vary wildly between images of the same semantics, that is, they have a color-texture pattern that will be one of the some patterns that belong to that particular objects' image class. So we decided to give less weight to shape feature and it is appropriate per our experiment results.

It is clear that the resemblance (or, equivalently, distance) of two images is conveyed through the similarities between regions from both images. Thus it is desirable to construct the image-level distances (dissimilarity) using region-level distances. Since image segmentation is usually not perfect, a region in one image could correspond to several regions in another image. For example, a segmentation algorithm may segment an image of dog into two regions: the dog and the background. The same algorithm may segment another image of a dog into five regions: the body of the dog, the front leg(s) of the dog, the rear leg(s) of the dog, the background grass, and the sky. There are similarities between the dog in the first image and the body, the front leg(s), or the rear leg(s) of the dog in the second image. The background of the first image is also similar to the background grass or the sky of the second image. However, the dog in the first image is unlikely to be similar to the background grass and sky in the second image.

Using the fuzzy feature representation, these similarity (or, equivalently, distance) observations can be expressed as

- (i) the distance measure, given by (17), for the fuzzy features of the dog in the first image and the fuzzy features of the dog body, front leg(s), or rear leg(s) in the second image is low (e.g., close to 0);
- (ii) the distance measure for the fuzzy feature of the background in the first image and the fuzzy features of the background grass or sky in the second image is also low;
- (iii) the distance measure for the fuzzy feature of the dog in the first image and the fuzzy feature of the background

grass in the second image is high (i.e., close to 1). The distance measure for the fuzzy feature of the dog in the first image and the fuzzy feature of the sky in the second image is also high.

Based on these qualitative illustrations, it is natural to think of the mathematical meaning of the word "or," that is, the union operation. What we have described above is essentially the matching of a fuzzy feature with the union of some other fuzzy features. The distance function  $d(i, J) = \text{Min}_k[d(i, J_k)]$  between a region  $i$  and a region set  $J$  ( $J_k$  enumerates regions in  $J$ ) in the region distance metric space has the property of the required union operation. Based on this motivation, we construct the image (a set of regions) distance measure through the following steps.

Suppose now we have  $M$  regions in image 1 and  $N$  regions in image 2.

*Step 1.* Calculate the distance between one region in image 1 and all regions in image 2. For each region  $i$  in image 1, the distance between it to whole image 2 is

$$R_{i\text{Image2}} = \text{Min} \{ \text{DIST}(i, j) \}, \quad (18)$$

where  $j$  is each region in image 2. Thus, we calculate the minimal distance between a region with all regions in another image (image 2) to be the distance between this region and the image, which means that we maximize the potential similarity between a region and an image.

*Step 2.* Similarly, we get the distance between a region  $j$  in image 2 to image 1

$$R_{j\text{Image1}} = \text{Min} \{ \text{DIST}(j, i) \}, \quad (19)$$

where  $i$  is each region in image 1.

*Step 3.* After obtaining  $M + N$  distances, we define the distance between the two images (1 and 2) as

$$\text{DistIge}(1, 2) = \frac{\sum_{i=1}^M w_{1i} R_{i\text{Image2}} + \sum_{j=1}^N w_{2j} R_{j\text{Image1}}}{2}, \quad (20)$$

where  $w_{1i}$  is the weight for each region in image 1. We set  $w_{1i} = N_{1i}/N_1$ , where  $N_{1i}$  is the number of blocks in region  $i$  and  $N_1$  is the total number of blocks in image 1.  $w_{2j}$  is defined similarly for image 2. In this way bigger regions are given more significance than smaller regions because we think that big regions are more semantically related to the subject of one image. We can compensate for the inaccuracy of clustering algorithm by using this integrated-region-distance formula so that the error of similarity calculated is reduced greatly.

For each query, the  $\text{DistIge}(q, d)$  is calculated for each image  $d$  in the database and sort their value to retrieve relevant images.

We briefly discuss the advantages of this image distance measures as follows.

- (i) It can be shown that, if images 1 and 2 are the same,  $\text{DistIge}(1, 2) = 0$ ; if images 1 and 2 are quite different, that is, region distances between region pairs from the two images are high,  $\text{DistIge}(1, 2)$  is high too. This property is desirable for CBIR ranking.
- (ii) To provide a comprehensive and robust “view” of distance measure between images, the region-level distances are combined, weighted, and added up to produce the image-level distance measure which depicts the overall difference of images in color, texture, and shape properties. The comprehensiveness and robustness of this distance metric can be examined from two perspectives. On one hand, each entry in (20) signifies the degree of closeness between a fuzzy feature in one image and all fuzzy features in the other image. Intuitively, an entry expresses how similar a region of one image is to all regions of the other image. Thus one region is allowed to be matched with several regions in case of inaccurate image segmentation in which practice occurs quite often. On the other hand, by weighted summation, every fuzzy feature in both images contributes a portion to the overall distance measure. This further reduces the sensitivity of the distance measure.

Based upon the above comparison, we expect that, under the same uncertain conditions, the proposed region-matching scheme can maintain more information from the image.

## 5. SECONDARY CLUSTERING AND IMAGE RETRIEVAL

The time of image retrieval depends largely on the number of images in the database in almost all CBIR systems. Many existing systems attempt to compare the query image with every image in the database to find the top matching images, resulting in an essentially linear search, which is time-prohibitive when the database is large. We believe that it is not necessary to conduct a whole database comparison. In fact, it is possible to exploit a priori information regarding the “organization” of the images in the database in the feature space before a query is posed, such that when a query is received, only a part of the database needs to be searched while a large portion of the database may be eliminated. This certainly reduces significant query processing time without compromising the retrieval precision.

To achieve this goal, in CLEAR we add a preretrieval screening phase to the feature space after a database is indexed by applying a secondary  $k$ -means clustering algorithm in the region feature vector space to cluster all the regions in the database into classes with the distance metric  $\text{DIST}_{pq}$ . The rationale is that regions with similar (color, texture, shape) features should be grouped together in the same class. This secondary clustering is performed offline, and each region’s indexing data along with its associated class information are recorded in the index files. Consequently, in the prototype implementation of CLEAR, the image database is indexed in terms of a three-level tree structure, one for the

region level, one for the class level, and one for the image level.

Assuming that an image database is indexed based on the features defined in Sections 3 and 4, and is “organized” based on the secondary clustering, given a query image, CLEAR processes the query in 4 steps.

*Step 1.* Perform the query image segmentation to obtain regions,  $Q_i, i \in [0, V - 1]$ , where  $V$  is the number of regions in the query image.

*Step 2.* Compute the distances between each region  $Q_i$  and all class centroids in the database to determine which class  $Q_i$  belongs to by the minimum-distance-win principle. Assume that the region  $Q_i$  belongs to class  $C_j, j \in [0, K - 1]$ , where  $K$  is the number of classes to which all regions are partitioned.

*Step 3.* Retrieve all regions in the database which belongs to the class  $C_j$ . A region set  $T_{jd}$  comprises these regions. The images containing any regions in the set  $T_{jd}$  are subsequently retrieved from the index structure. These images comprise an image set  $I_d$ .

*Step 4.* Compare the query image with the images in the image set  $I_d$ . The distance  $\text{DistIge}$  is used for each pair and the top-least-distance images are returned in the retrieval.

Three advantages are achieved through this secondary clustering procedure. First, it enhances the robustness of the image retrieval. Minor appearance variations in color, texture, and shape within and among regions do not distort the similarity measures due to the clustering in the region feature space which groups similar region features together in respective classes. Therefore, minor alterations in region features are nullified. Second, linear search is prevented with this retrieval algorithm. In other words, many statistically dissimilar images are excluded from comparison; only those potentially relevant images are chosen to be compared with the query image. Third, the effects of imprecise secondary clustering is controlled and mitigated because the second clustering is performed on the region feature space while the final image similarity measures are in the image space and are based on integrated region matching. In this way, the final image distance calculated with (20) is the “real” distance (not approximated) and the retrieval precision is not compromised.

The efficiency improvement of the proposed retrieval algorithm is analyzed as follows. Suppose  $n$  is the number of images in the database,  $l$  is the average number of regions of an image, and  $c$  is the number of classes obtained with the secondary clustering technique in the region feature space. Then  $nl$  is the total number of regions. In the average case, the number of regions associated with a class is  $q = nl/c$ , which is also the number of regions to compare with a query region (one query region is associated with only 1 class in the proposed algorithm). We call these regions “candidate regions.” Each candidate region corresponds to one image in the database. Thus, the total number of *different* images





FIGURE 3: Sample images in the testing database. The images in each column are assigned to one category. From left to right, the categories are Africa rural area, historical building, waterfalls, British royal event, and model portrait, respectively.

in the database to be compared with the query image is  $\lambda lq = \lambda nl^2/c$ , where  $\lambda$  is the ratio that describes the region-to-image correspondence relationship,  $\lambda \in [1/l, 1]$ . Then we observe that the average number of *different* images to be compared is bounded in  $[nl/c, nl^2/c]$ .  $l$  is determined by the resolution of the image segmentation and is typically small (4 to 6 in our implementation), while  $c$  is determined by the granularity of the secondary clustering in the region feature space (in our experiment on the testing database, the value of  $c$  has the magnitude order of the number of categories in the database, i. e., 100–200). When  $l^2/c < 1$ , which is realistic and feasible in large size databases with many different semantic categories, it is guaranteed that the number of *different* images chosen to compare with the query image is smaller than  $n$ . The size of candidate images is reduced (the reduction ratio is in  $[c/l^2, c/l]$ ), thus the query processing time is saved proportionally with reduced I/O accesses and computation needed assuming that the class information resides in main memory.

## 6. EXPERIMENTS AND RESULTS

We implemented the CLEAR method in a prototype system. For the discussion and reference purpose, we also call the prototype CLEAR. The following reported evaluations

were performed in a general-purpose color image database containing 10 000 images from the COREL collection of 96 semantic categories, including people, nature scene, building, and vehicles. No prerestriction on camera models, lighting conditions, and so forth are specified in the image database for the testing. These images are all in JPEG format. We chose this database to test the CLEAR method because it is accessible to the public and is used in the evaluations of several state-of-the-art CBIR systems, for example, IRM [15] and UFM [16]. The database is accessible at <http://www.fortune.binghamton.edu/download.html>. Figure 3 shows some samples of the images belonging to a few semantic categories in the database. Each semantic category in this image database has 85–120 associated images. From this database 1 500 images were randomly chosen from all categories as the query set. A retrieved image is considered a match if it belongs to the same category of the query image. We note that the category information in the COREL collection is only used to simplify the evaluation; we did not make use of any such information in the indexing and retrieval processing.

We implemented the system on a Pentium III 800 MHz computer with 256 M memory. After performing the image segmentation described in Section 3.1, the homogenous regions of each image were obtained. The original  $k$ -means

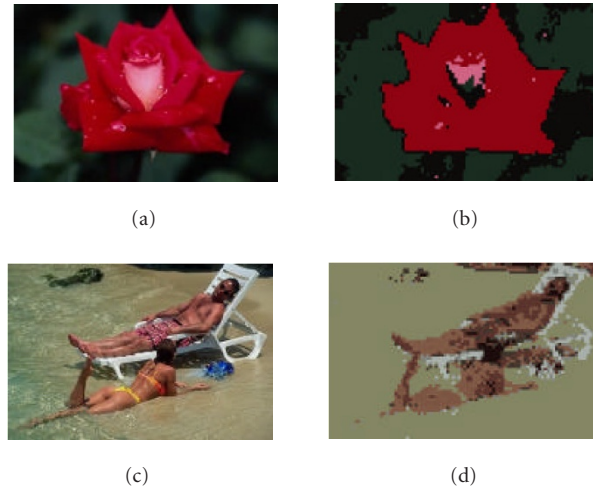


FIGURE 4: Regions obtained for two example images; each region is labeled with the average color of blocks belonged to it. (a) Image 65003. (b) Segmented image (4 regions). (c) Image 17821. (d) Segmented image (5 regions).

clustering algorithm was altered to address unknown number of regions in an image for image segmentation. We adaptively selected the number of clusters  $C$  by gradually increasing  $C$  until a stop criterion was met. The average number of regions for all images in the database changes in accordance with the adjustment of the stop criteria. Figure 4 shows the segmentation results for two example images. In this figure, (a) and (c) are two images in the database, and (b) and (d) are their region representations, respectively. Each region segmented is labeled by the average color of all the blocks associated with the region. As noted, 4 regions were obtained for image 65003 and 5 regions were obtained for image 17821. The segmentation results indicate that the regions extracted are related to the objects embodying image semantics. In our experiment totally 56 722 regions were extracted for all 10 000 images in the database, which means that in average 5.68 regions are extracted in image. Image segmentation for the testing database took 5.5 hours to be done, about 1.9 seconds for each image.

Consequently the fuzzy color histogram, fuzzy texture, and fuzzy shape features are determined for each region. Based on these feature of all regions extracted for the database, a three-level indexing structure was built offline. All regions are partitioned into several classes through performing adaptive  $k$ -means algorithm. For our testing database, the number of classes is determined to be 677 with the maximal number of regions in one class being 194 and the minimal number of regions in one class being 31. For each class, a hash table mapping the associated regions and the corresponding image names in the database is maintained. The generation of the three-level indexing structure took 70 minutes in the experiment. Although it is time consuming for offline indexing, the online query is fast. In average, the query time for returning top 30 images was less than 1 second. The retrieval interface of the prototype system is shown in Figure 5.

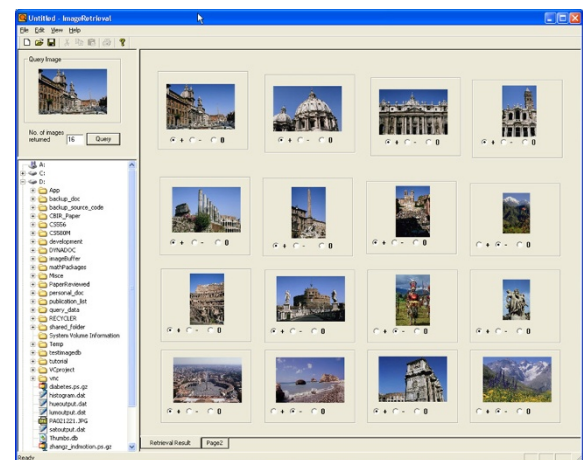


FIGURE 5: A screenshot of the prototype system CLEAR. The query image is in the top left pane and the retrieval results are returned in the right pane.

To illustrate the performance of the approach, several examples are shown in Figure 6 where 5 images with different semantics: flowers, dinosaurs, vehicles, African people, and dishes are picked as query images. For each query example, we examine the precision of the query results depending on the relevance of the image semantics. The semantic relevance evaluation is based on the group membership of the query image, which is done by human subjective observation. In Figure 6, the top-left corner image is the query and the ranking goes rightward and downward.

To evaluate our approach more quantitatively, we compared CLEAR with the UFM [16] system, one of the state-of-the-art CBIR systems, on the retrieval effectiveness. Retrieval effectiveness is measured by recall and precision metrics [31]. For a given query and a given number of images

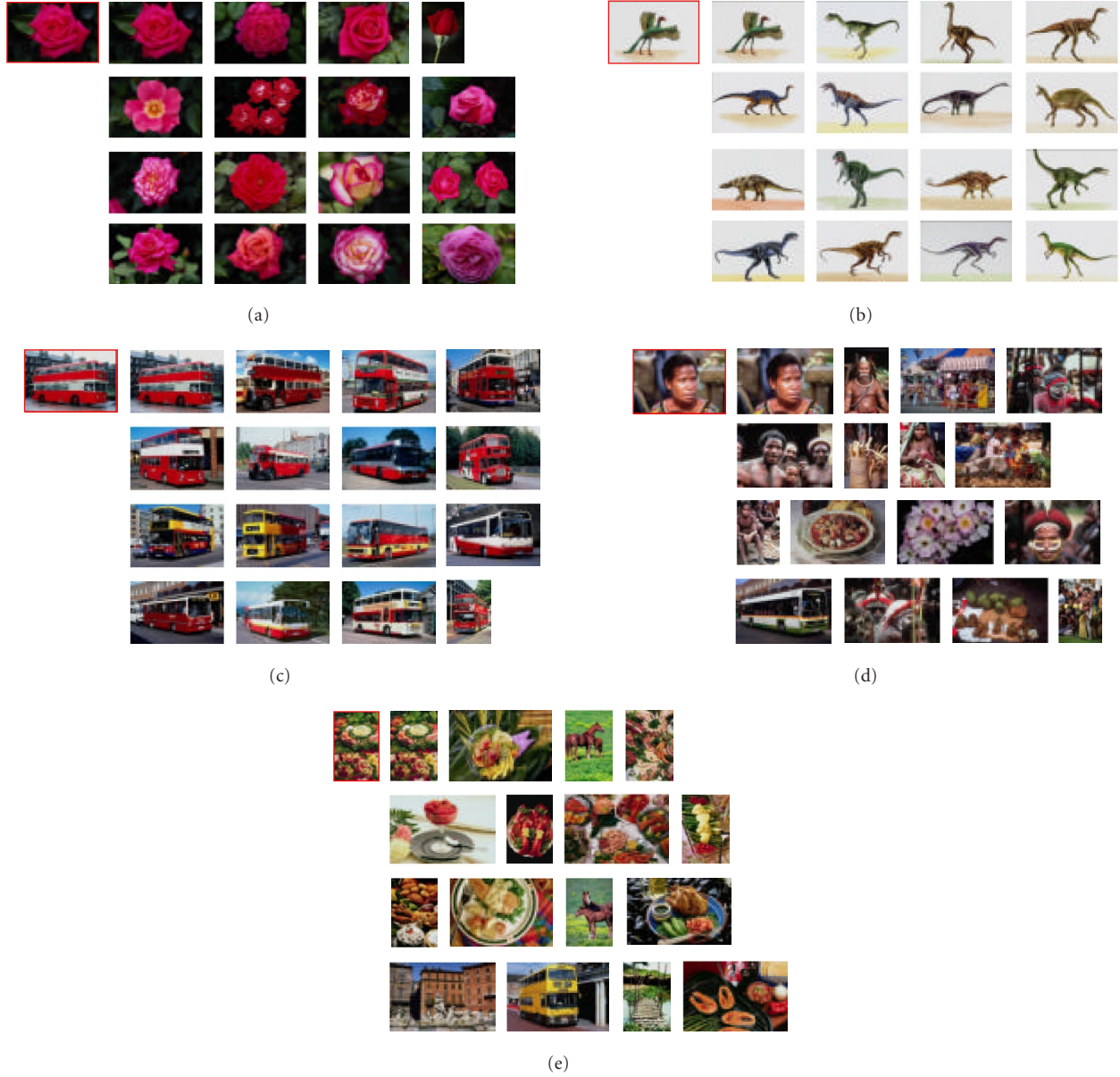


FIGURE 6: Retrieval results of the five queries evaluated. The top-left corner image is the query and the ranking goes rightward and downward. (a) Flower; 16 matches out of 16. (b) Dinosaur; 16 matches out of 16. (c) Vehicle; 16 matches out of 16. (d) African people; 12 matches out of 16. (e) Dish; 11 matches out of 16.

retrieved, precision gives the ratio between the number of relevant images retrieved and the number of retrieved images (measures the retrieval accuracy). Recall gives the ratio between the number of relevant images retrieved and the total number of relevant images in the collection considered (measures the missing ratio):

$$\begin{aligned} \text{Precision} &= \frac{|\text{relevant} \cap \text{results}|}{|\text{results}|}, \\ \text{Recall} &= \frac{|\text{relevant} \cap \text{result}|}{|\text{relevant}|}. \end{aligned} \quad (21)$$

The average precision results of the 1500 query images for

different number of returned images are recorded in Figure 7, which demonstrates that the retrieval precision of CLEAR is generally superior to that of UFM.

Because the number of relevant images for a query image in the large image database is difficult to be determined accurately in advance, for the sake of simplicity we supposed that the number of images in the semantic category to which the query image belongs is the number of relevant images in the database. In this way of evaluation, the recalls of CLEAR and UFM for different number of returned images are plotted in Figure 8.

The average recalls of CLEAR and UFM are comparable and the advantages of CLEAR to UFM are shown more



TABLE 1: Retrieval efficiency and scalability results.

| Database size | Average number of compared images | Average percentage of images examined | Average search overhead in CLEAR (s) | Average query processing time in CLEAR (s) | Average query processing time in linear search (s) |
|---------------|-----------------------------------|---------------------------------------|--------------------------------------|--------------------------------------------|----------------------------------------------------|
| 3000          | 795                               | 26.5%                                 | 0.08                                 | 0.55                                       | 1.78                                               |
| 6000          | 1668                              | 27.8%                                 | 0.12                                 | 0.79                                       | 3.04                                               |
| 10000         | 2832                              | 28.3%                                 | 0.15                                 | 0.98                                       | 3.96                                               |

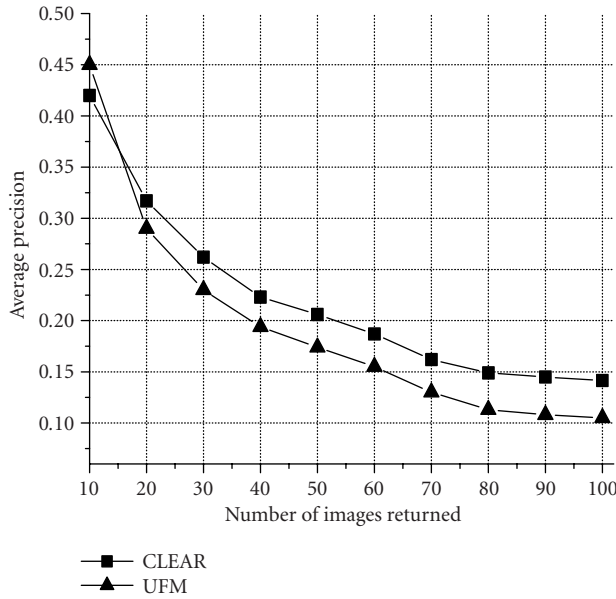


FIGURE 7: Average precision comparisons between CLEAR and UFM.

clearly when the number of images returned on which we calculate the average recall statistics increases. In other word, CLEAR has more potential than UFM on the retrieval recall performance. The better effectiveness is attributed in part to the more accurate representation of color, texture, and shape with the fuzzy model in our system.

For an example of retrieval precision comparison, Figure 9 shows the top 16 retrieved images in our system and UFM, respectively, using one image in the “medieval building” category as a query. For this query, 14 out of top 16 returned images by CLEAR are relevant in comparison with 9 of those returned by UFM.

To study the scalability of CLEAR, we incrementally sample the original 10 000 image database to generate two smaller databases, one with 3000 images and the other with 6000 images. These two databases contain sampled images from all the 96 categories. For each of the three databases, we randomly sample 100 images as the query set from the corresponding database for this evaluation. We recorded the average number of images compared in each of the three databases using CLEAR secondary clustering technique. The average indexing structure search overhead, the average query processing time in CLEAR, and the average

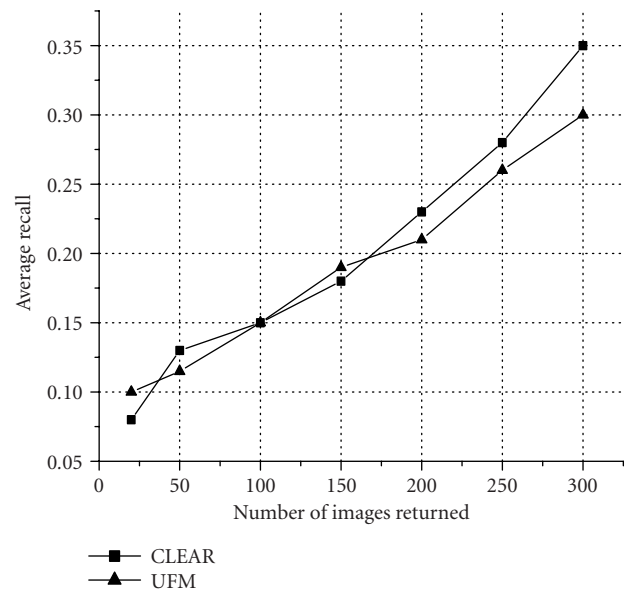


FIGURE 8: Average recall comparisons between CLEAR and UFM.

query processing time in learn search were also recorded. The results are documented in Table 1. It shows that the average number of compared images is significantly reduced in comparison with the database size. In addition, as indicated in Table 1, although the indexing structure search introduces the computation overhead, the average total query processing time is still much less than the average query processing time in linear search due to the reduced number of images to be compared with. The computation overhead for the indexing structure search is small because the search performs only several distance calculations and highly efficient hash table searches. With the increase of the database size, the percentage of the images examined and the average computation overhead remain relatively stable. The average query processing time is much less than that in the linear search in all the three testing databases. The average efficiency improvement on the query processing time to the linear search is 72.7%. This result, combined with the results observed in Figures 7 and 8, confirms CLEAR’s efficiency in handling large image databases without sacrificing retrieval effectiveness.

Since in CLEAR the size of the class level (clusters in the region feature space) information is much smaller than the index files for image in the database (in our experiments, the



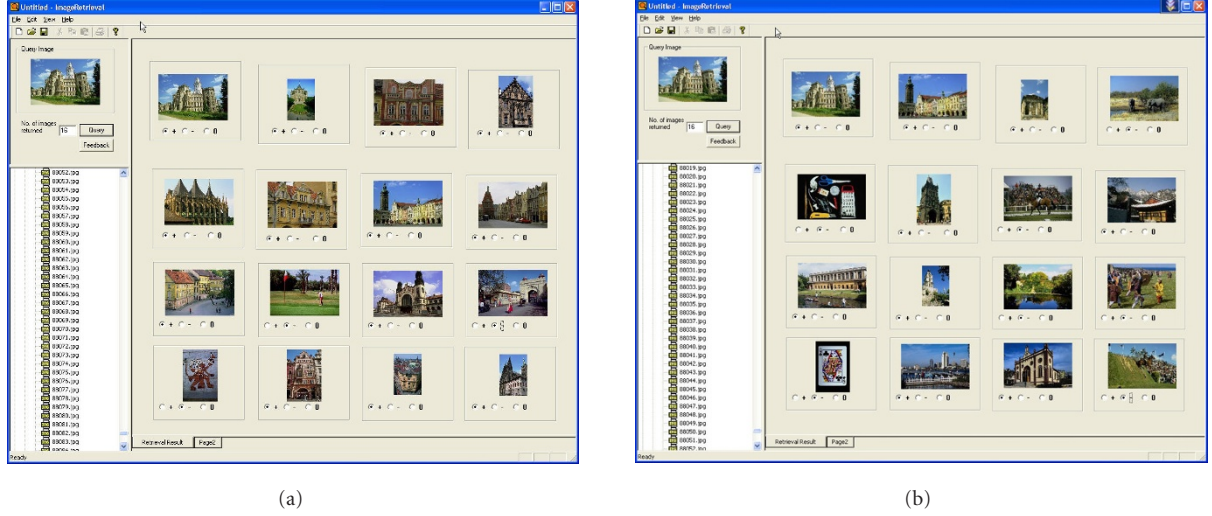


FIGURE 9: Retrieval comparisons of CLEAR and UFM using the image at the top-left pane of the window as the query. (a) Images found by CLEAR (14 of 16 images are relevant). (b) Images found by UFM (9 of 16 images are relevant).

size ratio is 1/95–1/120), it is practical and desirable to put the class level information in main memory. With such design, the I/O costs for each query are only proportional to the number of images compared. The reduced I/O costs in the CLEAR query processing were observed as shown in Table 1 as well.

In our indexing scheme, we use a Cauchy function to correlate color descriptions and to smooth the regions (equivalent to a convolution in computer vision) so that the color perception uncertainty and segmentation inaccuracy issue are addressed explicitly. To evaluate the effectiveness of the indexing scheme for improving the robustness to color variations and segmentation-related uncertainties, we compare the performance of CLEAR and UFM approaches for color variations and coarseness of image segmentation. Color variations can be simulated by changing colors to their adjacent values for each image, and the segmentation-related uncertainties in an image can be characterized by entropy. For image  $i$  with  $C$  segmented regions, its entropy,  $E(i)$ , is defined as

$$E(i) = - \sum_{j=1}^C P(R_j^i) \log [P(R_j^i)], \quad (22)$$

where  $P(R_j^i)$  is the percentage of image  $i$  covered by region  $R_j^i$ . The larger the value of entropy, the higher the uncertainty level. As we can see, the entropy  $E(i)$  increases with the increase of the number of regions  $C$ . Thus, we can adjust the uncertainty level by changing the value of  $C$ .  $C$  is controlled by modifying the stop criteria of the modified  $k$ -means algorithm. For a fair comparison between CLEAR indexing scheme and UFM at different color variation and uncertainty levels, we perform the same experiments for different degrees of color changes and average values of  $C$  (4.31, 6.32, 8.64, 11.62, and 12.25) on the 3000 image database in-

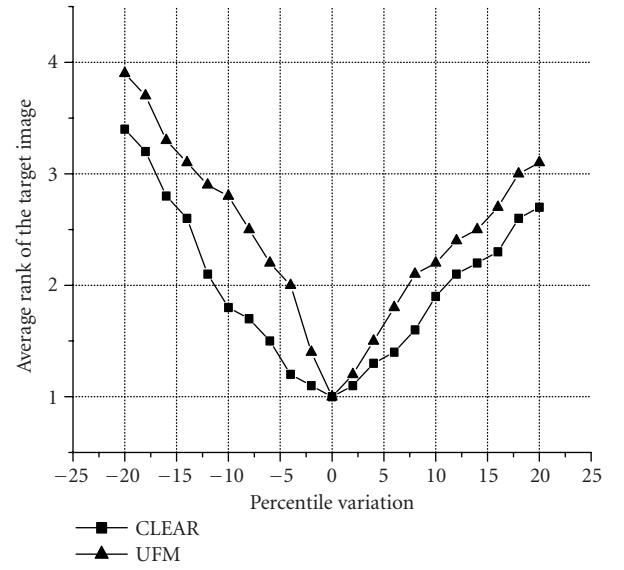


FIGURE 10: Comparison of CLEAR indexing scheme and UFM method on the robustness in the color variations. Every image in the 3000 image database is altered and used as the query image.

troduced above. To evaluate the robustness in the color variations, we apply color changes to an image (target image) in the database. The modified image is then used as the query image, and the rank of the retrieved target image is recorded. Repeating the process for all images in the testing database, the average rank for target images are computed for CLEAR and UFM. The result is shown in Figure 10. The average rank of the target image of CLEAR is lower than UFM for each level of color variations (in an acceptable range of color changes which do not affect semantics perception).

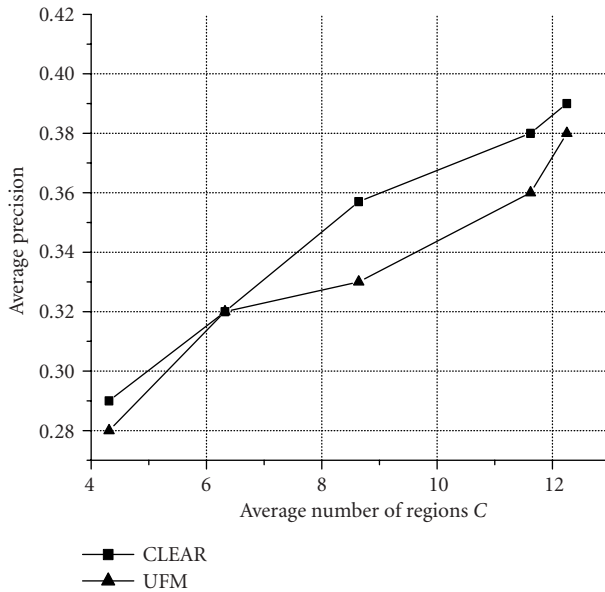


FIGURE 11: Comparison of CLEAR indexing scheme and UFM method on the robustness in image segmentation uncertainties.

To evaluate the robustness in the segmentation-related uncertainties, the performance in terms of overall average precision in top 30 returned images are evaluated for both approaches. The result is given in Figure 11. As we have known, the entropy  $E(i)$  (uncertainty) level increases when the image is segmented into more regions. At all uncertainty levels, CLEAR performs better than or as well as the UFM method. Combining these two experiments of robustness, we observed that CLEAR indexing scheme is more robust than UFM for color variations and segmentation-related uncertainties. The performance differences between CLEAR and UFM can be explained as follows. The UFM method uses the representative feature (one feature vector) of each region to model the segmentation uncertainty, which is coarse and artificial. The model generated is not accurate enough to fit the segmented images well. However, CLEAR indexing scheme leverages all block features in every region to generate fuzzy models for each feature component, thus describing the segmentation-related uncertainty more precisely and effectively.

## 7. CONCLUSIONS

A novel image indexing and retrieval methodology, CLEAR, is described. The methodology integrates color, texture, and shape information along with the conventional geometric information as an indexing vector, and applies the indexing vector to regions as opposed to a whole image. The overall image similarity is developed through regional similarity based on all the feature components. In order to address the color feature uncertainty problem and segmentation inaccuracy, our approach applies fuzzy set model to regional color histograms as well as texture and shape representa-

tions. CLEAR incorporates a secondary clustering technique to construct an indexing tree structure of the database to significantly reduce the search time. Experimental evaluation based on a 10 000 COREL image database shows that this approach outperforms the peer image retrieval systems previously described in the literature. In addition, the robustness of the fuzzy indexing scheme to color variations and segmentation-related uncertainties is proved to be another advantage of this approach.

Compared with existing techniques and systems, our approach has the following distinctive advantages: (i) it partially solves the color uncertainty problem typically found in color-based CBIR systems and exploits the inaccurate segmentation effect for the texture and shape features, (ii) it develops a balanced scheme in similarity measure between regional matching and global matching in order to capture as much semantic information as possible with no sacrifice in efficiency, and (iii) it preorganizes image databases by constructing an indexing structure to further improve retrieval efficiency without compromising retrieval effectiveness.

## REFERENCES

- [1] M. De Marsicoi, L. Cinque, and S. Levialdi, "Indexing pictorial documents by their content: a survey of current techniques," *Image and Vision Computing*, vol. 15, no. 2, pp. 119–141, 1997.
- [2] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.
- [3] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 1, pp. 56–63, 1999.
- [4] G. Pass and R. Zabih, "Histogram refinement for content-based image retrieval," in *Proc. 3rd IEEE Workshop on Applications of Computer Vision*, pp. 96–102, Sarasota, Fla, USA, December 1996.
- [5] M. Flickner, H. Sawhney, W. Niblack, et al., "Query by image and video content: the QBIC system," *IEEE Computer*, vol. 28, no. 9, pp. 23–32, 1995.
- [6] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762–768, San Juan, Puerto Rico, June 1997.
- [7] A. Rao, R. K. Srihari, and Z. Zhang, "Spatial color histograms for content-based image retrieval," in *Proc. 11th IEEE International Conference on Tools with Artificial Intelligence*, pp. 183–186, Chicago, IL, USA, November 1999.
- [8] L. Cinque, G. Ciocca, S. Levialdi, A. Pellicanò, and R. Schettini, "Color-based image retrieval using spatial-chromatic histograms," *Image and Vision Computing*, vol. 19, no. 13, pp. 979–986, 2001.
- [9] S. Ravela and R. Manmatha, "Retrieving images by appearance," in *Proc. 6th IEEE International Conference on Computer Vision (ICCV '98)*, pp. 608–613, Bombay, India, January 1998.
- [10] H. Wang, F. Guo, and D. D. Feng, "A signature for content-based image retrieval using a geometrical transform," in *Proc. 6th ACM International Multimedia Conference*, pp. 229–234, Bristol, UK, September 1998.
- [11] V. E. Ogle and M. Stonebraker, "Chabot: retrieval from a relational database of images," *IEEE Computer*, vol. 28, no. 9, pp. 40–48, 1995.

- [12] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," in *3rd International Conference on Visual Information Systems*, pp. 509–516, Springer, Amsterdam, The Netherlands, June 1999.
- [13] W. Y. Ma and B. Manjunath, "NeTra: a toolbox for navigating large image databases," in *Proc. International Conference on Image Processing*, vol. 1, pp. 568–571, Santa Barbara, CA, USA, 1997.
- [14] J. R. Smith and S. F. Chang, "VisualSEEk: a fully automated content-based image query system," in *Proc. ACM Multimedia '96*, pp. 87–98, ACM Press, Boston, Mass, USA, 1996.
- [15] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLiCity: semantics-sensitive integrated matching for picture libraries," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, 2001.
- [16] Y. Chen and J. Z. Wang, "A region-based fuzzy feature matching approach to content-based image retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1252–1267, 2002.
- [17] F. Jing, M. Li, H. Zhang, and B. Zhang, "An effective region-based image retrieval framework," in *Proc. 10th ACM Multimedia*, pp. 456–465, Juan-les-Pins, France, 2002.
- [18] K. Yu, W.-Y. Ma, V. Trespet, et al., "Knowing a tree from the forest: art image retrieval using a society of profiles," in *Proc. 11th Annual ACM International Conference on Multimedia*, Berkeley, Calif, USA, November 2003.
- [19] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papatomas, and P. N. Yianilos, "The Bayesian image retrieval system PicHunter: theory, implementation, and psychophysical experiments," *IEEE Transactions on Image Processing*, vol. 9, no. 1, pp. 20–37, 2000.
- [20] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1995.
- [21] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.
- [22] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied Statistics*, vol. 28, no. 1, pp. 100–108, 1979.
- [23] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, Montpelier, Vt, 1992.
- [24] A. Gersho, "Asymptotically optimal block quantization," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 373–380, 1979.
- [25] S. K. Pal, A. Ghosh, and M. K. Kundu, Eds., *Soft Computing for Image Processing*, Physica-Verlag, Heidelberg, 2000.
- [26] C. Vertan and N. Boujemaa, "Embedding fuzzy logic in content based image retrieval," in *19th International Meeting of the North American Fuzzy Information Processing Society (NAFIPS '00)*, pp. 85–89, Atlanta, Ga, USA, July 2000.
- [27] Z. Chi, H. Yan, and T. Pham, *Fuzzy Algorithms: with Applications to Image Processing and Pattern Recognition*, World Scientific, River Edge, NJ, USA, 1996.
- [28] J. J. Buckley and E. Eslami, *An Introduction to Fuzzy Logic and Fuzzy Sets*, Physica-Verlag, New York, 2002.
- [29] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, John Wiley & Sons, 1999.
- [30] J. Kautsky, N. K. Nichols, and D. L. B. Jupp, "Smoothed histogram modification for image processing," *Graphical Models and Image Processing*, vol. 26, no. 3, pp. 271–291, 1984.
- [31] C. J. van Rijsbergen, *Information Retrieval*, Butterworths, London, 2nd edition, 1979.

**Ruofei Zhang** received his B.S. degree in computer science and engineering from Xi'an Jiaotong University, China, in 1996, and his M.S. degree in electronics from Tsinghua University, China, in 1999. He is currently a Ph.D. candidate and Research Assistant in the Computer Science Department at the State University of New York, Binghamton, USA. He has worked as a Software Engineer in Tsinghua Tongfang Ltd., Beijing, China, and Ebase Interactive, Binghamton, USA, respectively. His research interests include computer vision and image understanding, multimedia database, multimedia information retrieval, pattern recognition, machine intelligence, and reusable object design. He has published a number of papers in these fields.



**Zhongfei (Mark) Zhang** is an Assistant Professor in the Computer Science Department at the State University of New York (SUNY) at Binghamton. He received his B.S. degree in electronics engineering (with Honors), M.S. degree in information sciences, both from Zhejiang University, China, and Ph.D. degree in computer science from the University of Massachusetts at Amherst, USA. He was on the faculty of Computer Science and Engineering Department, and a Research Scientist at the Center of Excellence for Document Analysis and Recognition (CEDAR), both at SUNY Buffalo. His research interests include computer vision and image understanding, pattern recognition, data mining and information fusion, and multimedia information indexing and retrieval, as well as biomedical engineering. He has been Principal Investigator/Coprincipal Investigator for several projects in these areas supported by the Federal Government, the New York State Government, as well as private industries. He holds four inventions, has served as a Reviewer/PC Member for many conferences and journals, as a Grant Review Panelist for governmental and private funding agencies, and is in the editorial board of Pattern Recognition. He has also served as a Technical Consultant for a number of industrial and governmental organizations. He is a recipient of NRC Visiting Fellow and SUNY Chancellor's Promising Inventor Award.

