# A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval

Yixin Chen, *Student Member, IEEE,* James Z. Wang, *Member, IEEE*

Yixin Chen is with the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA (e-mail: yixchen@cse.psu.edu).

James Z. Wang is with the School of Information Sciences and Technology and the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA (e-mail: jwang@ist.psu.edu).

**Abstract**

This paper proposes a fuzzy logic approach, UFM (unified feature matching), for region-based image retrieval. In our retrieval system, an image is represented by a set of segmented regions each of which is characterized by a fuzzy feature (fuzzy set) reflecting color, texture, and shape properties. As a result, an image is associated with a family of fuzzy features corresponding to regions. Fuzzy features naturally characterize the gradual transition between regions (blurry boundaries) within an image, and incorporate the segmentation-related uncertainties into the retrieval algorithm. The resemblance of two images is then defined as the overall similarity between two families of fuzzy features, and quantified by a similarity measure, UFM measure, which integrates properties of all the regions in the images. Compared with similarity measures based on individual regions and on all regions with crisp-valued feature representations, the UFM measure greatly reduces the influence of inaccurate segmentation, and provides a very intuitive quantification. The UFM has been implemented as a part of our experimental SIMPLIcity image retrieval system. The performance of the system is illustrated using examples from an image database of about 60,000 general-purpose images.

**Index Terms**— Content-based image retrieval, image classification, similarity measure, fuzzified region features, fuzzy data analysis.

# 1 Introduction

In various application domains such as entertainment, commerce, education, biomedicine, and crime prevention, the volume of digital data archives is growing rapidly. The very large repository of digital information raises challenging problems in retrieval and various other information manipulation tasks. Content-based image retrieval (CBIR) is aimed at efficient retrieval of relevant images from large image databases based on automatically derived imagery features. These features are typically extracted from shape, texture, or color properties of query image and images in the database (target images). The relevance between a query image and any target image is ranked according to a similarity measure computed from the features.

## 1.1 Previous Work

Similarity comparison is an important issue in CBIR. In general, the comparison is performed either globally using techniques such as histogram matching and color layout indexing, or locally based on decomposed regions (objects) of the images. There is a rich resource of prior work on

this subject [2]–[5],[7],[9],[10],[13]–[17],[19]–[22],[24],[25],[27]–[29],[31],[32],[34]. Due to limited space, we only review work most related to ours, which represents by no means the complete set.

As a relatively mature method, histogram matching has been applied to many general-purpose image retrieval systems such as IBM QBIC [7], MIT Photobook [22], Virage System [10], and Columbia VisualSEEK and WebSEEK [27], etc. Features (color, texture, shape, and geometry) in the above systems are usually represented in the form of histograms. A major drawback of the global histogram search lies in its sensitivity to intensity variations, color distortions, and cropping.

Many approaches have been proposed to improve the retrieval performance. The PicToSeek [9] system uses color models invariant to object geometry, object pose, and illumination. VisualSEEK and Virage systems attempt to reduce the influence of intensity variations and color distortions by employing spatial relationships and color layout in addition to those elementary color, texture, and shape features. The same idea of color layout indexing is extended in a later system, Stanford WBIIS [32], which, instead of averaging, characterizes the color variations over the spatial extent of an image by Daubechies' wavelet coefficients (in the lowest few frequency bands) and their variances. Schmid *et al.* [24] propose a method of indexing images based on local features of automatically detected interest points of images. Minka *et al.* [20] describe a learning algorithm for selecting and grouping features. The user guides the learning process by providing positive and negative examples. The approach presented in [29] uses what is called the Most Discriminating Features for image retrieval. These features are extracted from a set of training images by optimal linear projection. The Virage system allows users to adjust weights of implemented features according to their own perceptions. The PicHunter system [5] and the UIUC MARS [19] system are self-adaptable to different applications and different users based upon user feedbacks. To approximate the human perception of the shapes of the objects in the images, Del Bimbo *et al.* [3] introduce a measure of shape similarity using elastic matching. In [21], matching and retrieval are performed along what is referred to as perceptual dimensions which are obtained from subjective experiments and multidimensional scaling based on the model of human perception of color patterns. In [2], two distinct similarity measures, concerning respectively with fitting human perception and with the efficiency of data organization and indexing, are proposed for content-based image retrieval by shape similarity.

In a human visual system, although color and texture are fundamental aspects of visual

perceptions, human discernment of certain visual contents could potentially be associated with interesting classes of objects, or semantic meanings of objects in the image. A region-based retrieval system segments images into regions (objects), and retrieves images based on the similarity between regions. If image segmentation is ideal, it is relatively easy for the system to identify objects in the image and to match similar objects from different images. Region-based retrieval systems include the UCSB NeTra system [17], the Berkeley Blobworld system [4], and the query system with color region templates [28]. Researchers at Stanford have recently developed SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), an integrated region-based image retrieval system, using semantics classification [34].

The NeTra and Blobworld systems compare images based on individual regions. To query an image, the user is required to select regions and the corresponding features to evaluate similarity. Both systems tend to partition one object into several regions (fractions) with none of them being representative for the object, especially for images without distinctive objects and scenes. Consequently, it is often difficult for users to determine which fractions and features should be used for retrieval. While each fraction only signifies some local information about the object, this does not preclude them from representing the object as a group. However, not much attention has been paid to developing similarity measures that combine information from all of the regions. Jia *et al.* [13] describe a similarity measure using the information of the region contours of objects. Each contour is split into several segments. To compare two images, each segment of object contours in both images needs to go through a decision tree trained offline to obtain its class distribution. Their method is invariant to scale, translation, and rotation, but it is only effective for images containing objects with similar sizes and shapes as their main scene contents such as fishes and hand tools.

Recently, Li *et al.* [16] propose an integrated region matching (IRM) scheme for CBIR. In order to reduce the influence of inaccurate segmentation, the IRM measure allows for matching a region of one image to several regions of another image. That is, the region mapping between any two images is a many-to-many relationship. As a result, the similarity between two images is defined as the weighted sum of distances, in the feature space, between all regions from different images. Compared with retrieval systems based on individual regions, the IRM approach, decreases the impact of inaccurate segmentation by smoothing over the imprecision in distances. Nevertheless, the inaccuracies (or uncertainties) are not explicitly expressed in the IRM measure. Can we further improve the retrieval accuracy and robustness (against inac-

curate image segmentation) by using a similarity measure capable of representing imprecision that stems from imperfect segmentation?

### 1.2 Overview of Our Approach

Semantically precise image segmentation by an algorithm is very difficult [18], [26], [33], [35]. However, a single glance is sufficient for human to identify circles, straight lines, and other complex objects in a collection of points and to produce a meaningful assignment between objects and points in the image. Although those points cannot always be assigned unambiguously to objects, human recognition performance is hardly affected. We can often identify the object of interest correctly even when its boundary is very blurry. This is probably because the prior knowledge of similar objects and images may provide powerful assistance for human in recognition. Unfortunately, this prior knowledge is usually unavailable to most of the current CBIR systems. However, we argue that a similarity measure allowing for blurry boundaries between regions may increase the performance of a region-based CBIR system. To improve the robustness of a region-based image retrieval system against segmentation-related uncertainties, which always exist due to inaccurate image segmentation, we propose unified feature matching (UFM) scheme based on fuzzy logic theory.

Applying fuzzy processing techniques to CBIR has been extensively investigated in the literature. In [14], fuzzy logic is developed to interpret the overall color information of images. Nine colors that match human perceptual categories are chosen as features. Vertan *et al.* propose a fuzzy color histogram approach in [31]. A class of similarity distances is defined based on fuzzy logic operations. Our scheme is distinct from the above methods in two aspects:

- It is a region-based fuzzy feature matching approach. Segmentation-related uncertainties are viewed as blurring boundaries between segmented regions. Instead of a feature vector, we represent each region as a multidimensional fuzzy set, named fuzzy feature, in the feature space of color, texture, and shape. Thus, each image is characterized by a class of fuzzy features. Fuzzy features naturally characterize the gradual transition between regions (blurry boundaries) within an image. It assigns weights, called *degrees of membership*, to every feature vectors in the feature space. As a result, a feature vector 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.
- A novel image similarity measure, UFM measure, is derived from fuzzy set operations. The

matching of two images is performed in three steps. First, each fuzzy feature of the query image is matched with all fuzzy features of the target image in a *Winner Takes All* fashion. Then, each fuzzy feature of the target image is matched with all fuzzy features of the query image using the same strategy as in the previous step. Finally, overall similarity, given as the UFM measure, is calculated by properly weighting the results from the above two steps.

### 1.3 Outline of the Paper

The remainder of the paper is organized as follows. In Section 2, image segmentation and fuzzy feature representation of an image are described. Section 3 presents an algorithm to calculate the fuzzy similarity measure for two regions based upon the corresponding fuzzy features. The algorithm is also extended to characterize the image-level similarities. We then define the UFM measure which reflects the overall similarity between two images. An algorithmic presentation of the resulting CBIR system is provided in Section 4. Section 5 describes the experiments we have performed, and provides the results. And finally, we conclude in Section 6 together with a discussion of future work.

## 2 Image Segmentation and Representation

The building blocks for the UFM approach are segmented regions and the corresponding fuzzy features. In our system, the query image and all images in the database are first segmented into regions. Regions are then represented by multidimensional fuzzy sets in the feature space. The collection of fuzzy sets for all regions of an image constitutes the *signature* of the image.

### 2.1 Image Segmentation

Our system segments images based on color and spatial variation features using $k$-means algorithm [11], a very fast statistical clustering method. For general-purpose images such as the images in a photo library or the images on the World-Wide Web, precise object segmentation is nearly as difficult as computer semantics understanding. However, semantically-precise segmentation is not crucial to our system because our UFM approach is insensitive to inaccurate segmentation.

To segment an image, the system first partitions the image into small blocks. A feature vector is then extracted for each block. The block size is chosen to compromise between texture effectiveness and computation time. Smaller block size may preserve more texture details but increase the computation time as well. Conversely, increasing the block size can reduce the

computation time but lose texture information and increase the segmentation coarseness. In our current system, each block has $4 \times 4$ pixels. The size of the images in our database is either $256 \times 384$ or $384 \times 256$. Therefore each image corresponds to 6144 feature vectors. Each feature vector, $\vec{f_i}$, consists of six features, i.e., $\vec{f_i} \in \mathbb{R}^6$, $1 \leq i \leq 6144$. Three of them are the average color components in a $4 \times 4$ block. We use the well-known LUV color space, where L encodes luminance, and U and V encode color information (chrominance). The other three represent energy in the high frequency bands of the wavelet transforms [6], that is, the square root of the second order moment of wavelet coefficients in high frequency bands.

To obtain these moments, a Daubechies-4 wavelet transform is applied to the L component of the image. After a one-level wavelet transform, a $4 \times 4$ block is decomposed into four frequency bands: the LL, LH, HL, and HH bands. Each band contains $2 \times 2$ coefficients. Without loss of generality, suppose the coefficients in the HL band are $\{c_{k,l}, c_{k,l+1}, c_{k+1,l}, c_{k+1,l+1}\}$. One feature is

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

The other two features are computed similarly from the LH and HH bands. The motivation for using the features extracted from high frequency bands is that they reflect texture properties. Moments of wavelet coefficients in various frequency bands have been shown to be effective for representing texture [30]. The intuition behind this is that coefficients in different frequency bands show variations in different directions. For example, the HL band shows activities in the horizontal direction. An image with vertical strips thus has high energy in the HL band and low energy in the LH band.

The $k$-means algorithm is used to cluster the feature vectors into several classes with every class corresponding to one region in the segmented image, i.e., for an image with the set of feature vectors $\mathbf{F} = \{\vec{f_i} \in \mathbb{R}^6 : 1 \leq i \leq 6144\}$, $\mathbf{F}$ is partitioned into $C$ groups $\{\mathbf{F}_1, \cdots, \mathbf{F}_C\}$, and consequently, the image is segmented into $C$ regions $\{\mathbf{R}_1, \cdots, \mathbf{R}_C\}$ with $\mathbf{R}_j \subset \mathbb{N}^2$ being the region corresponding to the feature set $\mathbf{F}_j$, $1 \leq j \leq C$. Because clustering is performed in the feature space, blocks in each cluster do not necessarily form a connected region in the images. This way, we preserve the natural clustering of objects in textured images and allow classification of textured images [15]. 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

accordance with the adjustment of the stop criteria. As we will see in Section 5, the average number of clusters is closely related to segmentation-related uncertainty level, and hence affects the performance of the system.

After segmentation, three extra features are calculated for each region to describe shape properties. They are normalized inertia [8] of order 1 to 3. For a region $\mathbf{R}_j \subset \mathbb{N}^2$ in the image plane, which is a finite set, the normalized inertia of order $\gamma$ is given as

$$I_{(\mathbf{R}_j,\gamma)} = \frac{\sum_{(x,y):(x,y)\in\mathbf{R}_j}[(x-\hat{x})^2+(y-\hat{y})^2]^{\frac{\gamma}{2}}}{\mathrm{V}(\mathbf{R}_\mathrm{j})^{1+\frac{\gamma}{2}}},$$

where $(\hat{x},\hat{y})$ is the centroid of $\mathbf{R}_j$, $\mathrm{V}(\mathbf{R}_\mathrm{j})$ is the volume of $\mathbf{R}_j$. The normalized inertia is invariant to scaling and rotation. The minimum normalized inertia is achieved by spheres. Denote the $\gamma$th order normalized inertia of spheres as $\mathbf{I}_\gamma$. We define shape feature $\vec{h}_j$ of region $\mathbf{R}_j$ as $I_{(\mathbf{R}_j,\gamma)}$ normalized by $\mathbf{I}_\gamma$, i.e.,

$$\vec{h}_j = \left[\frac{I_{(\mathbf{R}_j,1)}}{\mathbf{I}_1}, \frac{I_{(\mathbf{R}_j,2)}}{\mathbf{I}_2}, \frac{I_{(\mathbf{R}_j,3)}}{\mathbf{I}_3}\right]^T.$$

### 2.2 Fuzzy Feature Representation of an Image

A segmented image can be viewed as a collection of regions, $\{\mathbf{R}_1,\cdots,\mathbf{R}_C\}$. Equivalently, in the feature space, the image is characterized by a collection of feature sets, $\{\mathbf{F}_1,\cdots,\mathbf{F}_C\}$, which form a partition of $\mathbf{F}$. We could use the feature set $\mathbf{F}_j$ to describe the region $\mathbf{R}_j$, and compute the similarity between two images based on $\mathbf{F}_j$'s. Representing regions by feature sets incorporates all the information available in the form of feature vectors, but it has two drawbacks:

- It is sensitive to segmentation-related uncertainties. For any feature vector in $\mathbf{F}$, under this region representation form, it belongs to exactly one feature set. But, in general, image segmentation cannot be perfect. As a result, for many feature vectors, a unique decision between in and not in the feature set is impossible.
- The computational cost for similarity calculation is very high. Usually, the similarity measure for two images is calculated based on the distances (Euclidean distance is the one that is commonly used in many applications) between feature vectors from different images. Therefore, for each image in the database, we need to compute $\frac{6144\times6145}{2}$ such distances. Even with a rather conservative assumption, one CPU clock cycle per distance, it takes about half an hour just to compute the Euclidean distances for all $60,000$ images in our database on a 700MHz PC. This amount of time is certainly too much for system users to tolerate.

In an improved region representation form [16], which mitigates the above drawbacks, each region ($\mathbf{R}_j$) is represented by the center ($\widehat{\vec{f}}_j$) of the corresponding feature set ($\mathbf{F}_j$) with $\widehat{\vec{f}}_j$ defined as

$$\widehat{\vec{f}}_j = \frac{\sum_{\vec{f} \in \mathbf{F}_j} \vec{f}}{\mathrm{V}(\mathbf{F}_\mathrm{j})}, \tag{1}$$

which is essentially the mean of all elements of $\mathbf{F}_j$, and in general may not be an element of $\mathbf{F}_j$. While averaging over all features in a feature set decreases the impact of inaccurate segmentation, at the same time, lots of useful information is also submerged in the smoothing process because a set of feature vectors are mapped to a single feature vector. Moreover, the segmentation-related uncertainties are not explicitly expressed in this region representation form.

Representing regions by fuzzy features, to some extent, combines the advantages and avoids the drawbacks of both region representation forms mentioned above. In this representation form, each region is associated with a fuzzy feature that assigns a value (between 0 and 1) to each feature vector in the feature space. The value, named *degree of membership*, illustrates the degree of wellness that a corresponding feature vector characterizes the region, and thus models the segmentation-related uncertainties. In Section 3, we will show that this representation leads to a computationally efficient region matching scheme if appropriate membership functions are selected.

A fuzzy feature $\widetilde{\mathbf{F}}$ on the feature space $\mathbb{R}^6$ is defined by a mapping $\mu_{\widetilde{\mathbf{F}}} : \mathbb{R}^6 \rightarrow [0, 1]$ named the *membership function*. For any feature vector $\vec{f} \in \mathbb{R}^6$, the value of $\mu_{\widetilde{\mathbf{F}}}(\vec{f})$ is called the degree of membership of $\vec{f}$ to the fuzzy feature $\widetilde{\mathbf{F}}$ (or, in short, the degree of membership to $\widetilde{\mathbf{F}}$). A value closer to 1 for $\mu_{\widetilde{\mathbf{F}}}(\vec{f})$ means more representative the feature vector $\vec{f}$ is to the corresponding region. For a fuzzy feature $\widetilde{\mathbf{F}}$, there is a smooth transition for the degree of membership to $\widetilde{\mathbf{F}}$ besides the hard cases $\vec{f} \in \widetilde{\mathbf{F}}$ ($\mu_{\widetilde{\mathbf{F}}}(\vec{f}) = 1$) and $\vec{f} \notin \widetilde{\mathbf{F}}$ ($\mu_{\widetilde{\mathbf{F}}}(\vec{f}) = 0$). It is clear that a fuzzy feature degenerates to a conventional feature set if the range of $\mu_{\widetilde{\mathbf{F}}}$ is $\{0, 1\}$ instead of $[0, 1]$ ($\mu_{\widetilde{\mathbf{F}}}$ is then called the *characteristic function* of the feature set).

Building or choosing a proper membership function is an application dependent problem. Some most commonly used prototype membership functions are cone, exponential, and Cauchy functions [12]. Two factors are considered when we select the membership function for our system: retrieval accuracy and computational intensity for evaluating a membership function. For different membership functions, although the discrepancies among the efforts of computing degrees of membership are small, it is not negligible for large-sized image databases as, in a

retrieval process, it is magnified by the product of the number of regions in the query image and the number of images in the database. As shown in Section 5.4, under proper parameters, the cone, exponential, and Cauchy functions can capture the uncertainties in feature vectors almost equally well, which is reflected by retrieval accuracies of the resulting systems. But computational intensities vary. As a result, we pick the Cauchy function due to its good expressiveness and high computational efficiency. A detailed comparison of all three membership functions are given in Section 5.4.

The Cauchy function, $\mathcal{C} : \mathbb{R}^k \rightarrow [0, 1]$, is defined as

$$\mathcal{C}(\vec{x}) = \frac{1}{1 + \left(\frac{\|\vec{x}-\vec{v}\|}{d}\right)^{\alpha}} \tag{2}$$

where $\vec{v} \in \mathbb{R}^k$, $d$ and $\alpha \in \mathbb{R}$, $d > 0$, $\alpha \geq 0$. $\vec{v}$ is the *center location* (point) of the function (or called the center location of the fuzzy set), $d$ represents the *width* ($\|\vec{x} - \vec{v}\|$ for $\mathcal{C}(\vec{x}) = 0.5$) of the function, and $\alpha$ determines the *shape* (or *smoothness*) of the function. Collectively, $d$ and $\alpha$ portray the *grade of fuzziness* of the corresponding fuzzy feature. For fixed $d$, the grade of fuzziness increases as $\alpha$ decreases. If $\alpha$ is fixed, the grade of fuzziness increases with the increasing of $d$. Figure 1 illustrates Cauchy functions in $\mathbb{R}$ 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 $\mathbb{R}$ (except 0, whose degree of membership is always 1 in this example) is 0.5.

Accordingly, region $\mathbf{R}_j$ is represented by fuzzy feature $\widetilde{\mathbf{F}}_j$ whose membership function, $\mu_{\widetilde{\mathbf{F}}_j} : \mathbb{R}^6 \rightarrow [0, 1]$, is defined as

$$\mu_{\widetilde{\mathbf{F}}_j}(\vec{f}) = \frac{1}{1 + \left(\frac{\|\vec{f}-\widehat{\vec{f}}_j\|}{d_f}\right)^{\alpha}} \tag{3}$$

where

$$d_f = \frac{2}{C(C-1)} \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} \|\widehat{\vec{f}}_i - \widehat{\vec{f}}_k\|$$

is the average distance between cluster centers, $\widehat{\vec{f}}_i$'s, defined by (1). An interesting property intrinsic to membership function (3) is that the farther a feature vector moves away from the cluster center, the lower its degree of membership to the fuzzy feature. At the same time, its degrees of membership to some other fuzzy features may be increasing. This nicely describes the gradual transition of region boundaries.

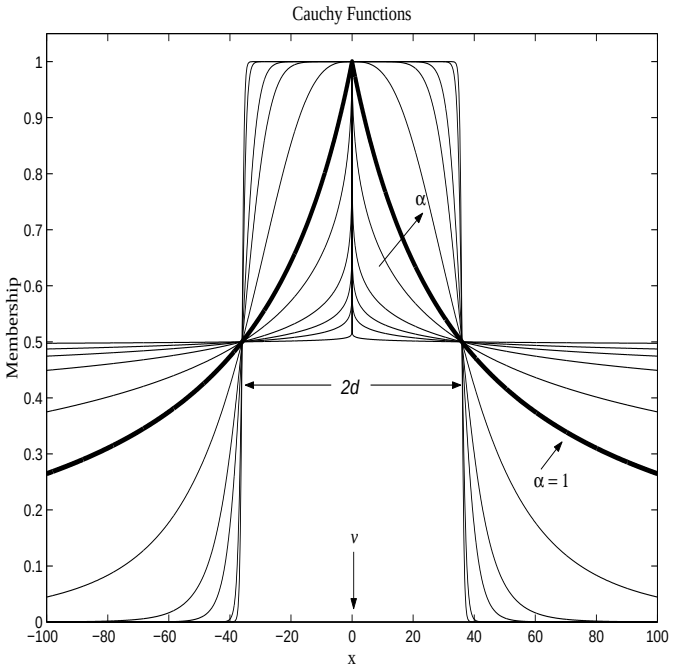



Fig. 1. Cauchy functions in $\mathbb{R}^1$.

As stated in Section 2.1, the shape properties of region $\mathbf{R}_j$ is described by shape feature $\vec{h}_j$. Considering the impacts of inaccurate segmentation on the shapes of regions, it is reasonable to use fuzzy sets to illustrate shape properties. Thus, for region $\mathbf{R}_j$, the shape feature $\vec{h}_j$ is extended to a fuzzy set $\widetilde{\mathbf{H}}_j$ with membership function, $\mu_{\widetilde{\mathbf{H}}_j} : \mathbb{R}^3 \rightarrow [0, 1]$, defined as

$$\mu_{\widetilde{\mathbf{H}}_j}(\vec{h}) = \frac{1}{1 + \left(\frac{\|\vec{h} - \vec{h}_j\|}{d_h}\right)^{\alpha}} \tag{4}$$

where

$$d_h = \frac{2}{C(C-1)} \sum_{i=1}^{C-1} \sum_{k=i+1}^{C} \|\vec{h}_i - \vec{h}_k\|$$

is the average distance between shape features. The experiments show that the performance changes insignificantly when $\alpha$ is in the interval $[0.9, 1.2]$, but degrades rapidly outside the interval. This is probably because, as $\alpha$ decreases, the Cauchy function becomes sharper within its center region ($[-d, d]$ for the example in Figure 1) and flatter outside. As a result, many useful feature vectors within that region are likely to be overlooked since their degrees of membership become smaller. Conversely, when $\alpha$ is large, the Cauchy function becomes flat within the center region. Consequently, the noise feature vectors in that region are likely to be selected as their degrees of membership are high. We set $\alpha = 1$ in both (3) and (4) based on the experimental results in Section 5.4.

For an image with regions $\mathbf{R}_j$, $1 \le j \le C$, $(\mathcal{F}, \mathcal{H})$ is named the *fuzzy feature representation* (or *signature*) of the image, where $\mathcal{F} = \{\widetilde{\mathbf{F}}_j : 1 \le j \le C, j \in \mathbb{N}\}$ with $\widetilde{\mathbf{F}}_j$ defined by (3), $\mathcal{H} = \{\widetilde{\mathbf{H}}_j : 1 \le j \le C, j \in \mathbb{N}\}$ with $\widetilde{\mathbf{H}}_j$ defined by (4). The color and texture properties are characterized by $\mathcal{F}$, while the shape properties are captured by $\mathcal{H}$.

### 2.3 An Algorithmic View

The image segmentation and fuzzy feature representation process can be summarized as follows. $\epsilon_1 > 0$ and $\epsilon_2 > 0$ are given stop criteria. The input is an image in raw format. The outputs is the signature of the image, $(\mathcal{F}, \mathcal{H})$, which is characterized by $\hat{\vec{f}}_j \in \mathbb{R}^6$ (center location) and $d_f > 0$ (width) of color/texture fuzzy features, and $\vec{h}_j \in \mathbb{R}^3$ (center location) and $d_h > 0$ (width) of shape fuzzy features. $j = 1 \ldots C$, $C$ is the number of regions.

**Algorithm 1: Image Segmentation and Fuzzy Features Extraction**

Partition the image into $B$ $4 \times 4$ blocks
**FOR** $i = 1$ **TO** $B$
    Extract feature vector $\vec{f}_i$ for block $i$
$k \leftarrow 2$, $D[1] \leftarrow 0$
**WHILE** $k \le M$
    Group $\{\vec{f}_i : 1 \le i \le B\}$ into $k$ clusters using the $k$-means algorithm
    $C \leftarrow k$
    **FOR** $j = 1$ **TO** $C$
        Compute the mean, $\hat{\vec{f}}_j$, for cluster $j$
    $D[k] \leftarrow \sum_{i=1}^{B} \min_{1 \le j \le C} \|\vec{f}_i - \hat{\vec{f}}_j\|^2$
    **IF** $D[k] < \epsilon_1$ **OR** $D[k] - D[k-1] < \epsilon_2$
        $k \leftarrow M + 1$
    **ELSE**
        $k \leftarrow k + 1$
**FOR** $j = 1$ **TO** $C$
    Compute shape feature $\vec{h}_j$ for region $j$
$d_f \leftarrow 0$, $d_h \leftarrow 0$
**FOR** $i = 1$ **TO** $C - 1$
    **FOR** $j = i + 1$ **TO** $C$
        $d_f \leftarrow d_f + \|\hat{\vec{f}}_i - \hat{\vec{f}}_j\|$

21 $\quad d_h \leftarrow d_h + \|\vec{h}_i - \vec{h}_j\|$

22 $d_f \leftarrow \frac{2d_f}{C(C-1)}$, $d_h \leftarrow \frac{2d_h}{C(C-1)}$

# 3 Unified Feature Matching

In this section, we describe the unified feature matching (UFM) scheme which characterizes the resemblance between images by integrating properties of all regions in the images. Based upon fuzzy feature representation of images, characterizing the similarity between images becomes an issue of finding similarities between fuzzy features. We first introduce a *fuzzy similarity measure* for two regions. The result is then extended to construct a *similarity vector* which includes the region-level similarities for all regions in two images. Accordingly, a *similarity vector pair* is defined to illustrate the resemblance between two images. Finally, the UFM measure maps a similarity vector pair to a scalar quantity, within the real interval $[0, 1]$, which quantifies the overall image-to-image similarity.

## 3.1 Similarity Between Regions: Fuzzy Similarity Measure

Considering the fuzzy feature representation of images, the similarity between two regions can be captured by a fuzzy similarity measure of the corresponding fuzzy features (fuzzy sets). In the classical set theory, there are many definitions of similarity measure for sets. For example, a similarity measure of set $\mathbf{A}$ and $\mathbf{B}$ can be defined as the maximum value of the characteristic function of $\mathbf{A} \cap \mathbf{B}$, i.e., if they have common elements then the similarity measure is 1 (most similar), otherwise 0 (least similar). If $\mathbf{A}$ and $\mathbf{B}$ are finite sets, another definition is $\frac{\mathrm{V}(\mathbf{A} \cap \mathbf{B})}{\sqrt{\mathrm{V}(\mathbf{A})\mathrm{V}(\mathbf{B})}}$, meaning the more elements they have in common, the more similar they are. Almost all similarity measures for conventional sets have their counterparts in fuzzy domain [1]. Taking the computational complexity into account, in this paper, we use a definition extended from the first definition mentioned above.

Before giving the formal definition of the fuzzy similarity measure for two fuzzy sets, we first define elementary set operations, intersection and union, for fuzzy sets. Let $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ be fuzzy sets defined on $\mathbb{R}^k$ with corresponding membership functions $\mu_{\widetilde{\mathbf{A}}} : \mathbb{R}^k \rightarrow [0, 1]$ and $\mu_{\widetilde{\mathbf{B}}} : \mathbb{R}^k \rightarrow [0, 1]$, respectively. The intersection of $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$, denoted by $\widetilde{\mathbf{A}} \cap \widetilde{\mathbf{B}}$, is a fuzzy set on $\mathbb{R}^k$ with membership function, $\mu_{\widetilde{\mathbf{A}} \cap \widetilde{\mathbf{B}}} : \mathbb{R}^k \rightarrow [0, 1]$, defined as

$$\mu_{\widetilde{\mathbf{A}} \cap \widetilde{\mathbf{B}}}(\vec{x}) = \min[\mu_{\widetilde{\mathbf{A}}}(\vec{x}), \mu_{\widetilde{\mathbf{B}}}(\vec{x})]. \tag{5}$$

The union $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$, denoted by $\widetilde{\mathbf{A}} \cup \widetilde{\mathbf{B}}$, is a fuzzy set on $\mathbb{R}^k$ with membership function, $\mu_{\widetilde{\mathbf{A}} \cup \widetilde{\mathbf{B}}} : \mathbb{R}^k \rightarrow [0, 1]$, defined as

$$\mu_{\widetilde{\mathbf{A}} \cup \widetilde{\mathbf{B}}}(\vec{x}) = \max[\mu_{\widetilde{\mathbf{A}}}(\vec{x}), \mu_{\widetilde{\mathbf{B}}}(\vec{x})]. \tag{6}$$

Note that there exists different definitions of intersection and union, the above definitions are computationally simplest [1].

The *fuzzy similarity measure* for fuzzy sets $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$, $\mathcal{S}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$, is given by

$$\mathcal{S}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \sup_{\vec{x} \in \mathbb{R}^k} \mu_{\widetilde{\mathbf{A}} \cap \widetilde{\mathbf{B}}}(\vec{x}). \tag{7}$$

It is clear that $\mathcal{S}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}})$ is always within the real interval $[0, 1]$ with a larger value denoting a higher degree of similarity between $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$. For the fuzzy sets defined by Cauchy functions, as in (2), calculating the fuzzy similarity measure according to (7) is relatively simple. This is because Cauchy function is unimodal, and therefore the maximum of (5) can only occur on the line segments connecting the center locations of two functions. It is not hard to show that for fuzzy sets $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$ on $\mathbb{R}^k$ with Cauchy membership functions

$$\mu_{\widetilde{\mathbf{A}}}(\vec{x}) = \frac{1}{1 + \left( \frac{\|\vec{x} - \vec{u}\|}{d_a} \right)^{\alpha}}$$

and

$$\mu_{\widetilde{\mathbf{B}}}(\vec{x}) = \frac{1}{1 + \left( \frac{\|\vec{x} - \vec{v}\|}{d_b} \right)^{\alpha}} \quad ,$$

the fuzzy similarity measure for $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{B}}$, which is defined by (7), can be equivalently written as

$$\mathcal{S}(\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}) = \frac{(d_a + d_b)^{\alpha}}{(d_a + d_b)^{\alpha} + \|\vec{u} - \vec{v}\|^{\alpha}} \quad . \tag{8}$$

### 3.2 Fuzzy Feature Matching

It is clear that the resemblance of two images is conveyed through the similarities between regions from both images. Thus it is desirable to construct the image-level similarity using region-level similarities. 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 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 fuzzy feature representation, these similarity observations can be expressed as:

- The similarity measure, given by (8), for the fuzzy feature 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 high (e.g., close to 1).
- The similarity 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 high.
- The similarity 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 small (e.g., close to 0). The similarity 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 small.

Based on these qualitative illustrations, it is natural to think of the mathematical meaning of the word **OR**, i.e., the union operation. What we have described above is essentially the matching of a fuzzy feature with the union of some other fuzzy features. Based on this motivation, we construct the *similarity vector* for two classes of fuzzy sets through the following steps.

Let $\mathcal{A} = \{\widetilde{\mathbf{A}}_i : 1 \leq i \leq C_a, i \in \mathbb{N}\}$, and $\mathcal{B} = \{\widetilde{\mathbf{B}}_j : 1 \leq j \leq C_b, j \in \mathbb{N}\}$ denote two collections of fuzzy sets. First, for every $\widetilde{\mathbf{A}}_i \in \mathcal{A}$, we define the similarity measure for it and $\mathcal{B}$ as

$$l_i^{\mathcal{B}} = \mathcal{S}(\widetilde{\mathbf{A}}_i, \bigcup_{j=1}^{C_b} \widetilde{\mathbf{B}}_j). \tag{9}$$

Combining $l_i^{\mathcal{B}}$'s together, we get a vector

$$\vec{l}^{\mathcal{B}} = [l_1^{\mathcal{B}}, l_2^{\mathcal{B}}, \cdots, l_{C_a}^{\mathcal{B}}]^T.$$

Similarly, for every $\widetilde{\mathbf{B}}_j \in \mathcal{B}$, we define the similarity measure between it and $\mathcal{A}$ as

$$l_j^{\mathcal{A}} = \mathcal{S}(\widetilde{\mathbf{B}}_j, \bigcup_{i=1}^{C_a} \widetilde{\mathbf{A}}_i). \tag{10}$$

Combining $l_j^{\mathcal{A}}$'s together, we get a vector

$$\vec{l}^{\mathcal{A}} = [l_1^{\mathcal{A}}, l_2^{\mathcal{A}}, \cdots, l_{C_b}^{\mathcal{A}}]^T.$$

It is clear that $\vec{l}^{\mathcal{B}}$ describes the similarity between individual fuzzy features in $\mathcal{A}$ and all fuzzy features in $\mathcal{B}$. Likewise, $\vec{l}^{\mathcal{A}}$ illustrates the similarity between individual fuzzy features in $\mathcal{B}$ and all fuzzy features in $\mathcal{A}$. Thus we define a *similarity vector* for $\mathcal{A}$ and $\mathcal{B}$, denoted by $\vec{L}^{(\mathcal{A},\mathcal{B})}$, as

$$\vec{L}^{(\mathcal{A},\mathcal{B})} = \begin{bmatrix} \vec{l}^{\mathcal{B}} \\ \vec{l}^{\mathcal{A}} \end{bmatrix},$$

which is a $C_a + C_b$ dimensional vector with values of all entries within the real interval $[0, 1]$.

It can be shown that if $\mathcal{A} = \mathcal{B}$ [1] then $\vec{L}^{(\mathcal{A},\mathcal{B})}$ contains all 1's. If a fuzzy set of $\mathcal{A}$ ($\mathcal{B}$) is quite different from all fuzzy sets of $\mathcal{B}$ ($\mathcal{A}$), in the sense that the distances between their centers are much larger than their widths, the corresponding entry in $\vec{L}^{(\mathcal{A},\mathcal{B})}$ would be close to 0. Using the definition of the union of fuzzy sets, which is given by (6), equations (9) and (10) can be equivalently written as

$$l_i^{\mathcal{B}} = \max_{j=1,\cdots,C_b} \mathcal{S}(\widetilde{\mathbf{A}}_i, \widetilde{\mathbf{B}}_j), \tag{11}$$

$$l_j^{\mathcal{A}} = \max_{i=1,\cdots,C_a} \mathcal{S}(\widetilde{\mathbf{B}}_j, \widetilde{\mathbf{A}}_i). \tag{12}$$

Equations (11) and (12) shows that computing the similarity measure for $\widetilde{\mathbf{A}}_i$ ($\widetilde{\mathbf{B}}_j$) and $\mathcal{B}$ ($\mathcal{A}$) is equivalent to calculating the similarity measures for $\widetilde{\mathbf{A}}_i$ ($\widetilde{\mathbf{B}}_j$) and $\widetilde{\mathbf{B}}_j$ ($\widetilde{\mathbf{A}}_i$) with $j$ taking integer values from 1 to $C_b$ ($i$ taking integer values from 1 to $C_a$), and then picking the maximum value, i.e., in a Winner Takes All fashion.

Let $(\mathcal{F}_q, \mathcal{H}_q)$ and $(\mathcal{F}_t, \mathcal{H}_t)$ be fuzzy feature representations, as defined in Section 2, for query image ($q$) and target image ($t$), respectively. The similarity between the query and target images is then captured by a *similarity vector pair* $(\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}, \vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)})$ where $\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ depicts the similarity in colors and textures, and $\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$ describes the similarity in shapes. Within the similarity vectors, $\vec{l}^{\mathcal{F}_q}$ and $\vec{l}^{\mathcal{H}_q}$ refer to the similarity between the query image and regions of the target image. Likewise, $\vec{l}^{\mathcal{F}_t}$ and $\vec{l}^{\mathcal{H}_t}$ designate the similarity between the target image and regions of the query image.

### 3.3 The UFM Measure

Endeavoring to provide an overall image-to-image and intuitive similarity quantification, the UFM measure is defined as the summation of all the weighted entries of similarity vectors $\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ and $\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$. We have discussed the methods of computing similarity vectors in Sections 3.1 and 3.2. The problem is then converted to designing a weighting scheme. The UFM

[1] $\mathcal{A} = \mathcal{B}$ if and only if the membership functions of fuzzy sets in $\mathcal{A}$ are the same as those of fuzzy sets in $\mathcal{B}$.

measure is computed in two stages. First, the inner products of similarity vectors $\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ and $\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$ with weight vectors $\vec{w}_1$ and $\vec{w}_2$, respectively, are calculated. The results are then weighted by $\rho_1$ and $\rho_2$, and added up to give the UFM measure $m_{(q,t)}$.

There are many ways of choosing weight vectors $\vec{w}_1$ and $\vec{w}_2$. For example, in a *uniform weighting scheme* we assume every region being equally important. Thus all entries of $\vec{w}_1$ and $\vec{w}_2$ equal to $\frac{1}{C_q+C_t}$ where $C_q$ ($C_t$) is the number of regions in the query (target) image. Such weight vectors favor the image with more regions because, in both $\vec{w}_1$ and $\vec{w}_2$, the summation of weights associated with the regions of the query (target) image is $\frac{C_q}{C_q+C_t}$ ($\frac{C_t}{C_q+C_t}$). If the regions within the same image are regarded as equally important, then the weights for entries of $\vec{l}^{\mathcal{F}_q}$ and $\vec{l}^{\mathcal{H}_q}$ ($\vec{l}^{\mathcal{F}_t}$ and $\vec{l}^{\mathcal{H}_t}$) can be chosen as $\frac{1}{2C_q}$ ($\frac{1}{2C_t}$). It is clear that regions from the image with less regions are allocated larger weights (if $C_q = C_t$ then the weights are identical to those under the uniform weighting scheme). We can also take the location of the regions into account, and assign higher weights to regions closer to the center of the image (*center favored scheme*, assuming the most important objects are always near the image center) or conversely to regions adjacent to the image boundary (*border favored scheme*, assuming images with similar semantics have similar backgrounds). Another choice is *area percentage scheme*. It uses the percentage of the image covered by a region as the weight for that region based on the viewpoint that important objects in an image tend to occupy larger areas.

In the UFM measure, both area percentage and border favored schemes are used. The weight vectors $\vec{w}_1$ and $\vec{w}_2$ are defined as

$$\begin{aligned} \vec{w}_1 &= (1-\lambda)\vec{w}_a + \lambda\vec{w}_b, \\ \vec{w}_2 &= \vec{w}_a, \end{aligned}$$

where $\vec{w}_a$ contains the normalized area percentages of the query and target images, $\vec{w}_b$ contains normalized weights [2] which favor regions near the image boundary, $\lambda \in [0,1]$ adjusts the significance of $\vec{w}_a$ and $\vec{w}_b$ in $\vec{w}_1$. The weights $\rho_1$ and $\rho_2$ are given by

$$\begin{aligned} \rho_1 &= 1-\rho, \\ \rho_2 &= \rho, \end{aligned}$$

where $\rho$ is within the real interval $[0,1]$. Consequently, the UFM measure for query image $q$

[2]Both the summation of all entries of $\vec{w}_a$ and that of $\vec{w}_b$ equal 1.

and target image $t$ is defined as

$$m_{(q,t)} = (1-\rho)\left[(1-\lambda)\vec{w}_a + \lambda\vec{w}_b\right]^T \vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)} + \rho\vec{w}_a^T\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}. \quad (13)$$

As shown by equation (13), the UFM measure incorporates three similarity components captured by $\vec{w}_a^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$, $\vec{w}_b^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$, and $\vec{w}_a^T\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$:

- $\vec{w}_a^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ contributes to the UFM measure from a color and texture perspective because $\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ reflects the color and texture resemblance between the query and target images. In addition, the matching of regions with larger areas is favored which is the direct consequence of the area percentage weighting scheme.
- $\vec{w}_b^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ also expresses the color and texture resemblance between images. But, unlike in $\vec{w}_a^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$, regions adjacent to the image boundaries are given a higher preference because of the border favored weight vector $\vec{w}_b$. Intuitively, $\vec{w}_b^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ characterizes the similarity between the backgrounds of images.
- Similarly, $\vec{w}_a^T\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$ describes the similarity of the shape properties of the regions (or objects) in both images since $\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$ contains similarity measures for shape features.

Weighted by $\lambda$ and $\rho$, the aforementioned similarity components are then synthesized into the UFM measure, in which $\left[(1-\lambda)\vec{w}_a + \lambda\vec{w}_b\right]^T\vec{L}^{(\mathcal{F}_q,\mathcal{F}_t)}$ represents the color and texture similarity with contributions from the area percentage and the border favored schemes weighted by $\lambda$, while $\rho$ determines the significance of the shape similarity, $\vec{w}_a^T\vec{L}^{(\mathcal{H}_q,\mathcal{H}_t)}$, with respect to the color and texture similarity. In our system, the query image is automatically classified as either a textured or a non-textured image (for details see [15]). For textured images, the information of the shape similarity is skipped ($\rho = 0$) in the UFM measure since region shape is not perceptually important for such images. For non-textured images, $\rho$ is chosen to be 0.1. Experiments indicate that including shape similarity as a small fraction of the UFM measure can improve the overall performance of the system. We intentionally stress color and texture similarities more than shape similarity because, compared with the color and texture features, shape features used in our system are more sensitive to image segmentation as demonstrated by sample images in Figure 5. The weight parameter $\lambda$ is set to be 0.1 for all images. Experiments show that large $\lambda$ is beneficial to categorizing images with similar background patterns. For example, the background of images of flowers often consists of green leaves and images of elephants are very likely to have trees in them. Thus emphasizing backgrounds can help grouping images, such as flowers or elephants, together. But the above background assumption is in general not

true. In our observation, the overall image categorization performance degrades significantly for $\lambda > 0.5$. When $\rho$ and $\lambda$ are within $[0.05, 0.3]$, no major system performance deterioration is noticed in our experiments.

$m_{(q,t)}$ is always in the real interval $[0, 1]$ because $\vec{w}_a$ and $\vec{w}_b$ are normalized weight vectors, and $\rho$ and $\lambda$ are within $[0, 1]$. It is easy to check that $m_{(q,t)} = 1$ if two images are same. The experiments show that there is little resemblance between images if $m_{(q,t)} \leq 0.5$. In this sense, the UFM measure is very intuitive for query users.

### 3.4 An Algorithmic View

An algorithmic outline of the UFM algorithm is given as below. Weights $\rho, \lambda \in [0, 1]$ are fixed. Inputs are $(\mathcal{F}_q, \mathcal{H}_q)$ (characterized by $\hat{\vec{f}}_j \in \mathbb{R}^6$, $d_f \in \mathbb{R}$, $\vec{h}_j \in \mathbb{R}^3$, $d_h \in \mathbb{R}$, $1 \leq j \leq C_q$), $(\mathcal{F}_t, \mathcal{H}_t)$ (characterized by $\hat{\vec{f}'}_j \in \mathbb{R}^6$, $d'_f \in \mathbb{R}$, $\vec{h}'_j \in \mathbb{R}^3$, $d'_h \in \mathbb{R}$, $1 \leq j \leq C_t$), and weight vectors $\vec{w}_a, \vec{w}_b \in \mathbb{R}^{C_q + C_t}$. The UFM measure $m_{(q,t)}$ is the output.

**Algorithm 2: Unified Feature Matching**

**FOR** $i = 1$ **TO** $C_q$
  $\vec{L}^{(\mathcal{F}_q, \mathcal{F}_t)}[i] \leftarrow \frac{d_f + d'_f}{d_f + d'_f + \min_{j=1,\ldots,C_t} \|\hat{\vec{f}}_i - \hat{\vec{f}'}_j\|}$
  **IF** the query image is non-textured
    $\vec{L}^{(\mathcal{H}_q, \mathcal{H}_t)}[i] \leftarrow \frac{d_h + d'_h}{d_h + d'_h + \min_{j=1,\ldots,C_t} \|\vec{h}_i - \vec{h}'_j\|}$
**FOR** $i = 1$ **TO** $C_t$
  $\vec{L}^{(\mathcal{F}_q, \mathcal{F}_t)}[i + C_q] \leftarrow \frac{d_f + d'_f}{d_f + d'_f + \min_{j=1,\ldots,C_q} \|\hat{\vec{f}'}_i - \hat{\vec{f}}_j\|}$
  **IF** the query image is non-textured
    $\vec{L}^{(\mathcal{H}_q, \mathcal{H}_t)}[i + C_q] \leftarrow \frac{d_h + d'_h}{d_h + d'_h + \min_{j=1,\ldots,C_q} \|\vec{h}'_i - \vec{h}_j\|}$
$m_{(q,t)} \leftarrow [(1 - \lambda)\vec{w}_a + \lambda \vec{w}_b]^T \vec{L}^{(\mathcal{F}_q, \mathcal{F}_t)}$
**IF** the query image is non-textured
  $m_{(q,t)} \leftarrow (1 - \rho) m_{(q,t)} + \rho \vec{w}_a^T \vec{L}^{(\mathcal{H}_q, \mathcal{H}_t)}$

## 4 An Algorithmic Summarization of the System

Based on the results given in Section 2 and Section 3, we describe the overall image retrieval and indexing scheme as follows.

1. Pre-processing image database

To generate the codebook for an image database, signatures for all images in the database are

extracted by Algorithm 1. Each image is classified as either a textured or a non-textured image using techniques in [15]. The whole process is very time-consuming. Fortunately, for a given image database, it is performed once for all.

2. Pre-processing query image

Here we consider two scenarios, namely inside query and outside query. For inside query, the query image is in the database. Therefore, the fuzzy features and *semantic types* (textured or non-textured image) can be directly loaded from the codebook. If a query image is not in the database (outside query), the image is first expanded or contracted so that the maximum value of the resulting width and height is 384 and the aspect ratio of the image is preserved. Fuzzy features are then computed for the resized query image. Finally, the query image is classified as textured or non-textured image.

3. Computing the UFM measures

Using Algorithm 2, the UFM measures are evaluated for the query image and all images in the database, which have semantic types identical to that of the query image.

4. Returning query results

Images in the database are sorted in a descending order according to the UFM measures obtained from the previous step. Depending on a user specified number $n$, the system returns the first $n$ images. The *quick sort* algorithm is applied here.

## 5 Experiments

We implemented the UFM in our experimental SIMPLIcity image retrieval system. The system is tested on a general-purpose image database (from COREL) including about $60,000$ pictures, which are stored in JPEG format with size $384 \times 256$ or $256 \times 384$. These images were automatically classified into two semantic types: textured photograph, and non-textured photograph [15]. For each image, the features, locations, and areas of all its regions are stored. In Section 5.1, we provide several query results on the COREL database to demonstrate qualitatively the accuracy and robustness (to image alterations) of the UFM scheme. Section 5.2 presents systematic evaluations of the UFM scheme, and compares the performance of UFM with those of the IRM [16] and EMD-based color histogram [23] approaches based on a subset of the COREL database. The speed of the UFM scheme is compared with that of two other region-based methods in Section 5.3. The effect of the choice of membership functions on the performance of the system is presented in Section 5.4.

### 5.1 Query Examples

To qualitatively evaluate the accuracy of the system over the 60,000-image COREL database, we randomly pick 5 query images with different semantics, namely natural out-door scene, horses, people, vehicle, and flag. For each query example, we exam the precision of the query results depending on the relevance of the image semantics. We admit that the relevance of image semantics depends on the standpoint of the user. Thus our relevance criteria, specified in Figure 2, may be quite different from those used by a user of the system. Due to space limitation, only top 19 matches to each query are shown in Figure 2. We also provide the number of relevant images among top 31 matches. More matches can be viewed from the on-line demonstration site by using the query image ID, given in Figure 2, to repeat the retrieval.

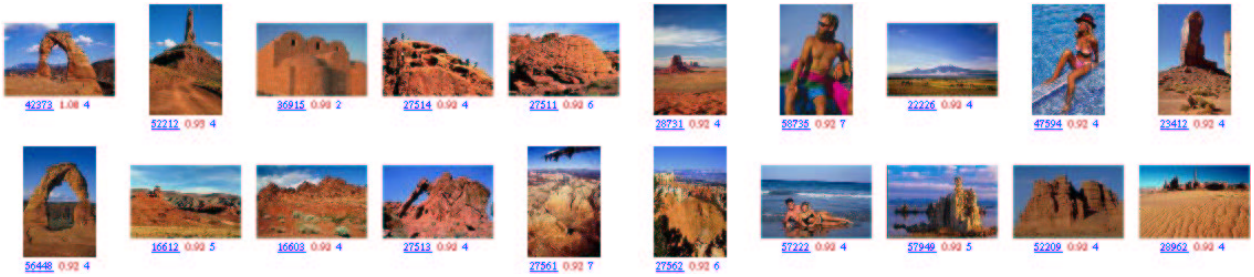
The robustness of the UFM scheme to image alterations, such as intensity variation, sharpness variation, color distortion, cropping, shifting, rotation, and other intentional distortions, is also tested. Figure 3 shows some query results using the 60,000-image COREL database. The query image is the left image for each group of images. In this example, the first retrieved image is exactly the unaltered version of the query image for all tested image alterations except sharpening, in which case, the unaltered version appears in the second place.

### 5.2 Systematic Evaluation

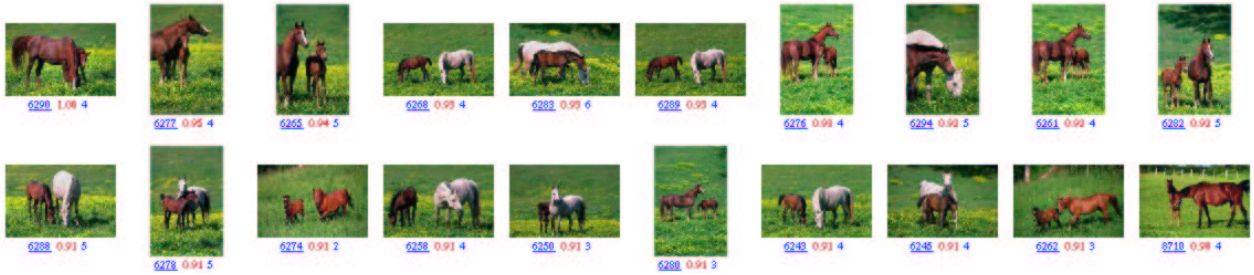
The UFM scheme is quantitatively evaluated focusing on the accuracy, the robustness to image segmentation, and the robustness to image alterations. Comparisons with the EMD-based color histogram system [23] and the region-based IRM system [16] are also provided. However, it is hard to make objective comparisons with some other region-based searching algorithms such as the Blobworld and the NeTra systems which require additional information provided by the user during the retrieval process.

#### 5.2.1 Experiment Setup

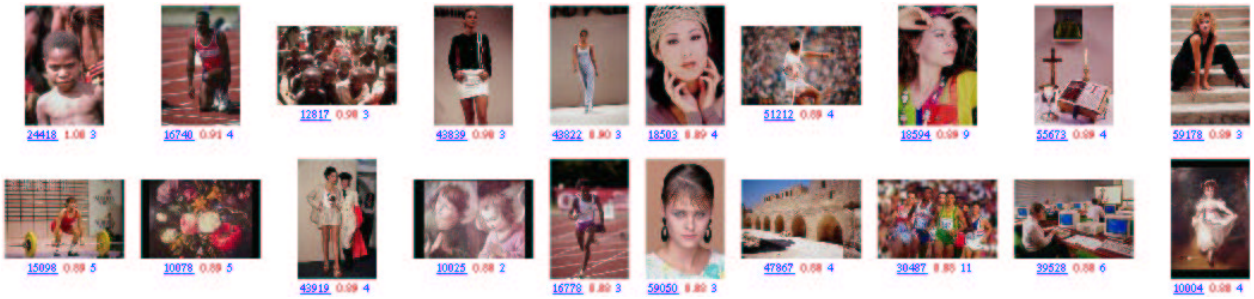
To provide more objective comparisons, the UFM scheme is evaluated based on a subset of the COREL database, formed by 10 image categories, each containing 100 pictures. The categories are Africa, Beach, Buildings, Buses, Dinosaurs, Elephants, Flowers, Horses, Mountains, and Food with corresponding Category ID's denoted by integers from 1 to 10, respectively. Within this database, it is known whether any two images are of the same category. In particular, a



(a) Natural out-door scene; 15 matches out of 19; 23 matches out of 31

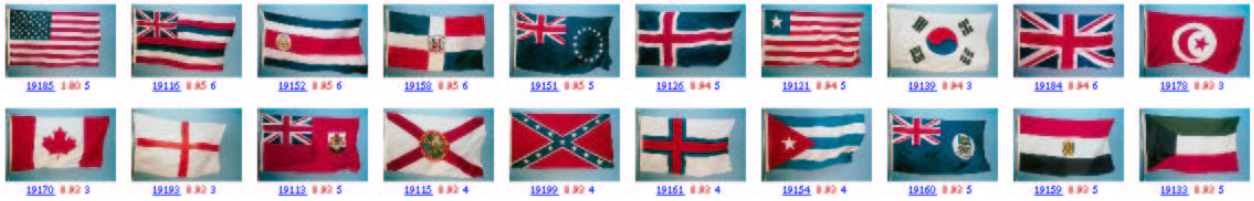


(b) Horses; 19 matches out of 19; 28 matches out of 31



(c) People; 15 matches out of 19; 23 matches out of 31

(d) Vehicle; 17 matches out of 19; 24 matches out of 31



(e) Flag; 19 matches out of 19; 25 matches out of 31

Fig. 2. The accuracy of the UFM scheme. For each block of images, the query image is on the upper-left corner. There are three numbers below each image. From left to right they are: the ID of the image in the database, the value of the UFM measure between the query image and the matched image, and the number of regions in the image.

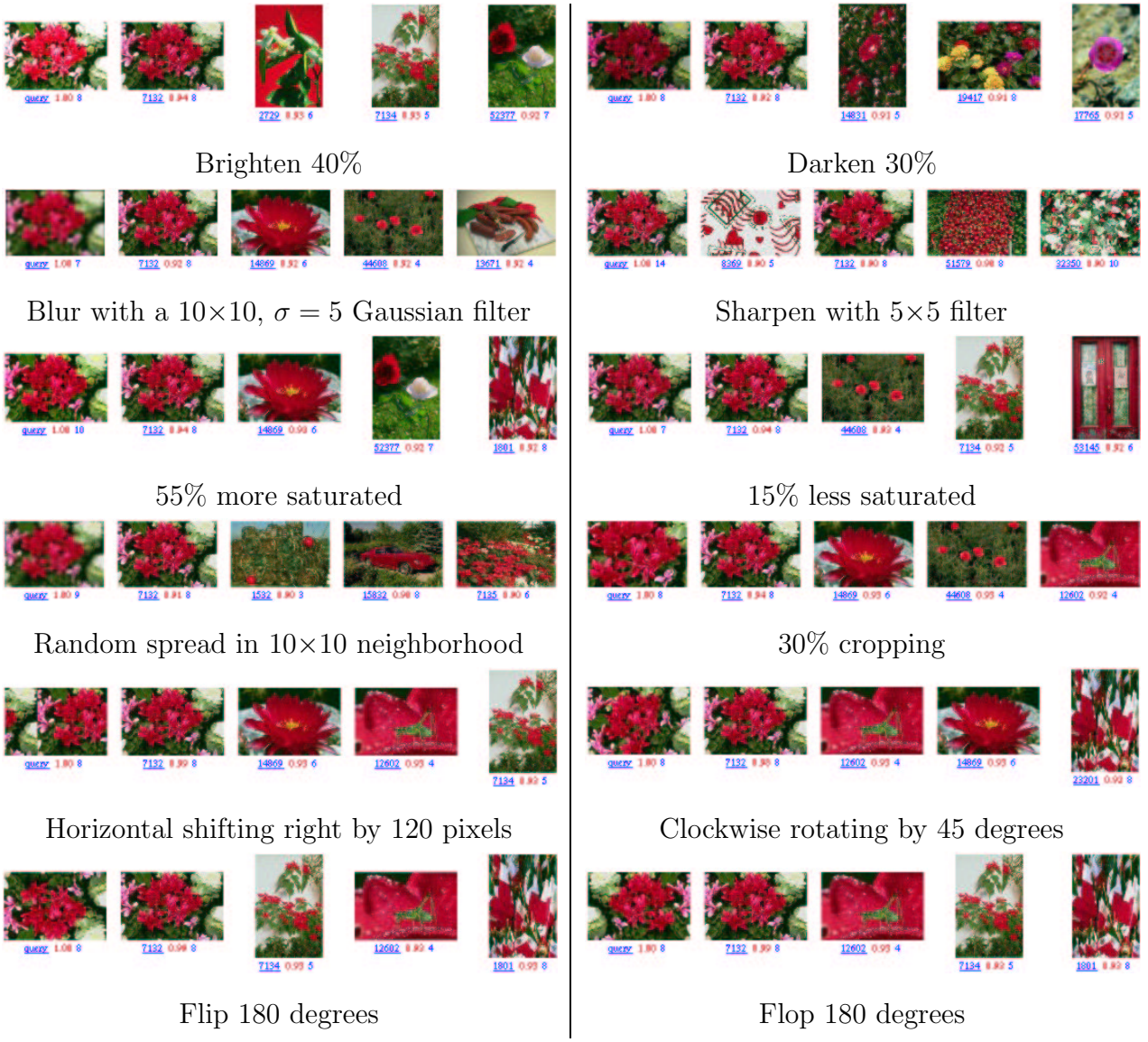



Fig. 3. The robustness of the UFM scheme against image alterations.

retrieved image is considered a correct match if and only if it is in the same category as the query. This assumption is reasonable since the 10 categories were chosen so that each depicts a distinct semantic topic. Every image in the sub-database is tested as a query, and the positions of all the retrieval images are recorded.

The following are some notations used in the performance evaluation. $ID(i)$ denotes the *Category ID* of image $i$ ($1 \leq i \leq 1000$ since there are totally 1000 images in the sub-database). It is clear that $ID(i)$ is an integer between 1 and 10 for any $1 \leq i \leq 1000$. For a query image $i$, $r(i,j)$ is the *rank* of image $j$ (position of image $j$ in the retrieved images for query image $i$,

it is an integer between 1 and 1000). The *precision* for query image $i$, $p(i)$, is defined by

$$p(i) = \frac{1}{100} \sum_{1 \le j \le 1000,\, r(i,j) \le 100,\, ID(j)=ID(i)} 1,$$

which is the percentile of images belonging to the category of image $i$ in the first 100 retrieved images. Another two statistics are also computed for query image $i$. They are the *mean rank* $r(i)$ of all the matched images and the *standard deviation* $\sigma(i)$ of the matched images, which are defined by

$$\begin{aligned} r(i) &= \frac{1}{100} \sum_{1 \le j \le 1000,\, ID(j)=ID(i)} r(i,j), \\ \sigma(i) &= \sqrt{\frac{1}{100} \sum_{1 \le j \le 1000,\, ID(j)=ID(i)} [r(i,j) - r(i)]^2}. \end{aligned}$$

Based on above definitions, we define the *average precision* $p_t$, *average mean rank* $r_t$, and *average standard deviation* $\sigma_t$ for Category $t$ ($1 \le t \le 10$) as

$$p_t = \frac{1}{100} \sum_{1 \le i \le 1000,\, ID(i)=t} p(i), \tag{14}$$

$$r_t = \frac{1}{100} \sum_{1 \le i \le 1000,\, ID(i)=t} r(i), \tag{15}$$

$$\sigma_t = \frac{1}{100} \sum_{1 \le i \le 1000,\, ID(i)=t} \sigma(i). \tag{16}$$

Similarly, the *overall average precision* $p$, *overall average mean rank* $r$, and *overall average standard deviation* $\sigma$ for all images in the sub-database are defined by

$$p = \frac{1}{1000} \sum_{i=1}^{1000} p(i), \tag{17}$$

$$r = \frac{1}{1000} \sum_{i=1}^{1000} r(i), \tag{18}$$

$$\sigma = \frac{1}{1000} \sum_{i=1}^{1000} \sigma(i). \tag{19}$$

Finally, we use entropy to characterize the segmentation-related uncertainties in an image. For image $i$ with $C$ segmented regions, its entropy, $E(i)$, is defined as

$$E(i) = -\sum_{j=1}^{C} P(\mathbf{R}_j^i) \log[P(\mathbf{R}_j^i)], \tag{20}$$

where $P(\mathbf{R}_j^i)$ is the percentage of image $i$ covered by region $\mathbf{R}_j^i$. The larger the value of entropy, the higher the uncertainty level. Accordingly, the *overall average entropy* $E$ for all images in the sub-database are define by

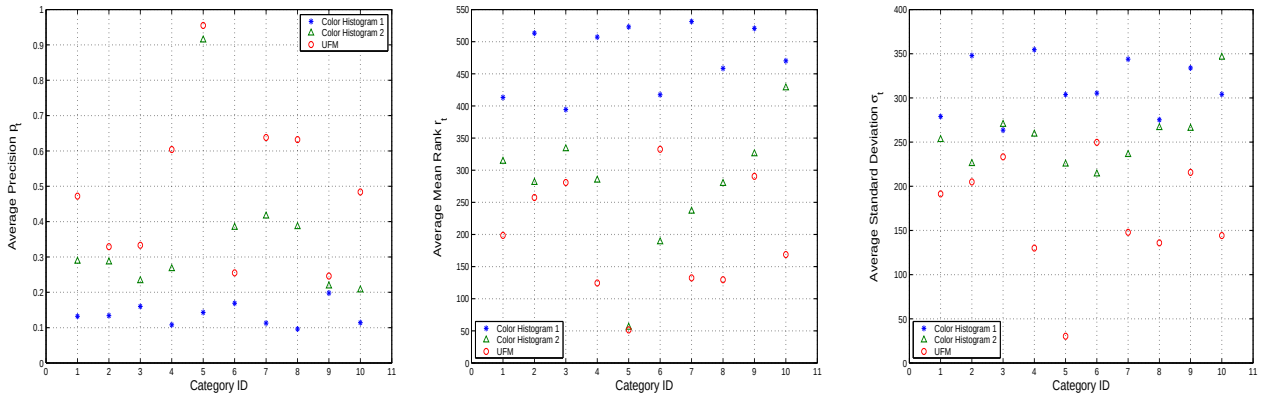
$$E = \frac{1}{1000} \sum_{i=1}^{1000} E(i). \tag{21}$$

### 5.2.2 Performance on Image Categorization

For image categorization, good performance is achieved when images belonging to the category of the query image are retrieved with low ranks. To that end, the average precision $p_t$ and the average mean rank $r_t$ should be maximized and minimized, respectively. The best performance, $p_t = 1$ and $r_t = 50.5$, occurs when the first 100 retrieved images belong to Category $t$ for any query image from Category $t$ (since the total number of semantically related images for each query is fixed to be 100). The worst performance, $p_t = 0$ and $r_t = 950.5$, happens when no image in the first 900 retrieved images belongs to Category $t$ for any query image from Category $t$. For a system that ranks images randomly, $p_t$ is about 0.1, and $r_t$ is about 500 for any Category $t$. Consequently, the overall average precision $p$ is about 0.1, and the overall average mean rank $r$ is about 500. In the experiments, the recall within the first 100 retrieved images was not computed because it is proportional to the precision in this special case.

The UFM scheme is compared with the EMD-based color histogram matching approach. We use the LUV color space and a matching metric similar to the EMD described in [23] to extract color histogram features and match in the categorized image database. Two different color bin sizes, with an average of 13.1 and 42.6 filled color bins per image, are evaluated. we call the one with less filled color bins the Color Histogram 1 system and the other the Color Histogram 2 system. Comparisons of average precision $p_t$, average mean rank $r_t$, and average standard deviation $\sigma_t$ are given in Figure 4. $p_t$, $r_t$, and $\sigma_t$ are computed according to equations (14), (15), and (16), respectively.

It is clear that the UFM scheme performs much better than both of the two color histogram-based approaches in almost all image categories. The performance of the Color Histogram 2 system is better that that of the Color Histogram 1 system due to more detailed color separation obtained with more filled bins. However, the price paid for the performance improvement is the decrease in speed. The UFM runs at about twice the speed of the relatively fast Color Histogram 1 system and still provides much better retrieval accuracy than the extremely slow Color Histogram 2 system.




Average precision $p_t$ Average mean rank $r_t$ Average standard deviation $\sigma_t$

Fig. 4. Comparing the UFM scheme with the EMD-based color histogram approaches on average precision $p_t$, average mean rank $r_t$, and average standard deviation $\sigma_t$. For $p_t$, the larger numbers indicate better results. For $r_t$ and $\sigma_t$, the lower numbers denote better results.

The UFM scheme is also compared with the IRM approach [16] using the same image segmentation algorithm with the average number of regions per image for all images in the sub-database being 8.64. Experiment results show that the UFM scheme outperforms the IRM approach by a 6.2% increase in overall average precision, a 6.7% decrease in the overall average mean rank, and a 4.0% decrease in the overall average standard deviation.

### 5.2.3 Robustness to Segmentation-Related Uncertainties

Because image segmentation cannot be perfect, being robust to segmentation-related uncertainties becomes a critical performance index for a region-based image retrieval system. In this section, we compare the performance of the UFM and IRM approaches with respect to the coarseness of image segmentation. We use the entropy, defined by equation (20), to measure the segmentation-related uncertainty levels. As we will see, the overall average entropy $E$, given by (21), increases with the increase of the *average number of regions* $C$ for all images in the sub-database. Thus, we can adjust the average uncertainty level through changing the value of $C$. The control of $C$ is achieved by modifying the stop criteria of the $k$-means algorithm. Figure 5 shows two images, beach scene and bird, and the segmentation results with different number of regions. Segmented regions are shown in their representative colors. Segmentation results for all images in the database can be found on the demonstration web site `http://wang.ist.psu.edu/IMAGE/`.

To give a fair comparison between UFM and IRM at different uncertainty levels, we perform the same experiments for different values of $C$ (4.31, 6.32, 8.64, 11.62, and 12.25). Based

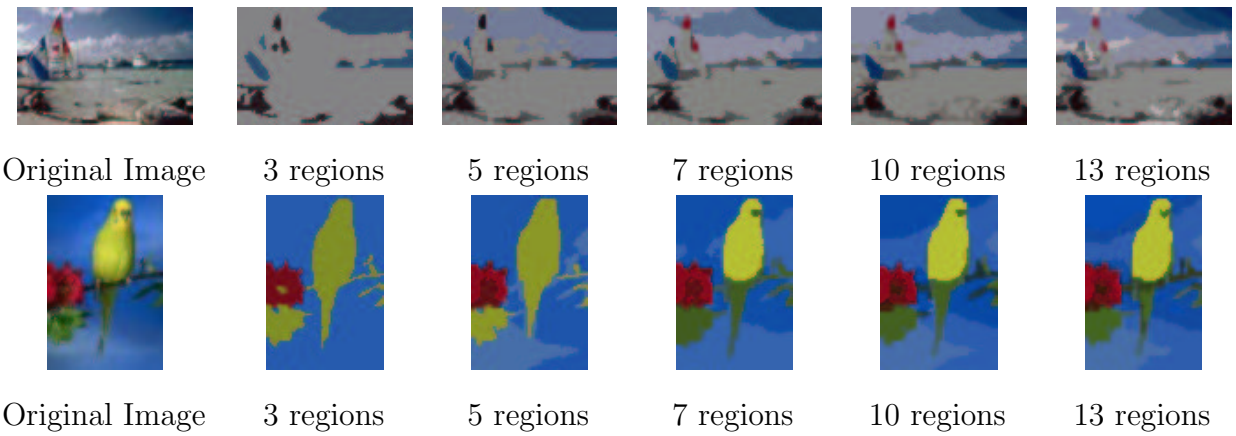



Fig. 5. Segmentation results by the $k$-means clustering algorithm. Original images are in the first column.

on equations (17), (18), and (19), the performance in terms of overall average precision $p$, overall average mean rank $r$, and overall average standard deviation $\sigma$ are evaluated for both approaches. The results are given in Figure 6. As we can see, the overall average entropy

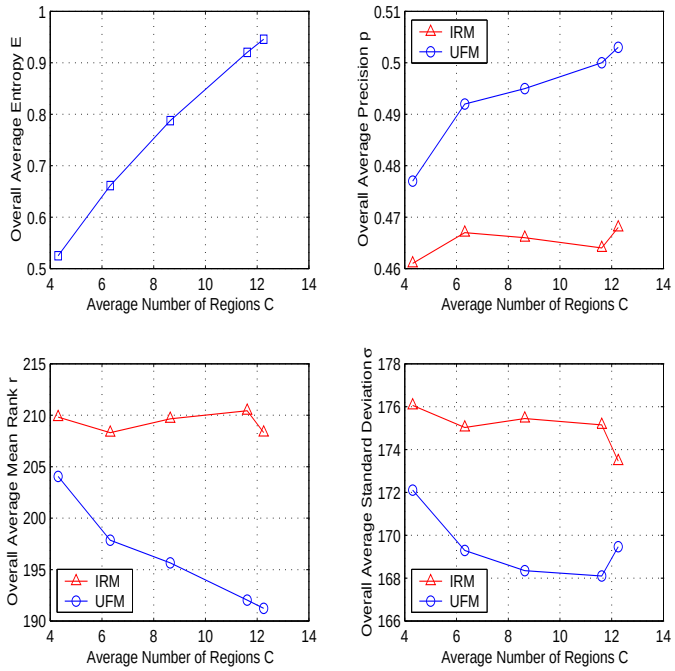



Fig. 6. Comparing the UFM scheme with the IRM method on the robustness to image segmentation: overall average entropy $E$, overall average precision $p$, overall average mean rank $r$, and overall average standard deviation $\sigma$.

$E$ increases when images are, on average, segmented into more regions. In other words, the uncertainty level increases when segmentation becomes finer. At all uncertainty levels, the UFM scheme performs better than the IRM method in all three statistics, namely $p$, $r$, and $\sigma$. In addition, there is a significant increase in $p$ and a decrease in $r$ for the UFM scheme as

the average number of regions increases. While for the IRM method, $p$ and $r$ almost remain unchanged for all values of $C$. This can be explained as follows. When segmentation becomes finer, although the uncertainty level increases, more details (or information) about the original image are also preserved (as shown in Figure 5). Compared with the IRM method, the UFM scheme is more robust to segmentation-related uncertainties and thus benefits more from the increasing of the average amount of information per image.

### 5.2.4 Robustness to Image Alterations

The UFM approach has been tested for the robustness to image alterations including intensity variation, color distortion, sharpness variation, shape distortion, cropping, and shifting. The goal is to demonstrate the ability of the system to recognize an image when its altered version is submitted as the query. We apply image alteration to an image (called target image $i$) in the sub-database. The resulting image $i'$ is then used as the query image, and the rank of the retrieved target image $i$, $r(i', i)$, is recorded. Repeating the process for all images in the sub-database, the average rank $r'$ for target images and the standard deviation $\sigma'$ of the rank are computed as

$$r' = \frac{1}{1000}\sum_{i=1}^{1000} r(i', i) \tag{22}$$

$$\sigma' = \sqrt{\frac{1}{1000}\sum_{i=1}^{1000}[r(i', i) - r']^2}. \tag{23}$$

Clearly, smaller numbers for $r'$ and $\sigma'$ indicate more robust performance.

For each type of image alteration, curves for $r'$ and $\sigma'$ with respect to the intensity of image alteration are plotted in Figure 7. If we call a system being robust to image alterations when the target image appear in the first 10 retrieved images, then, on average, the UFM scheme is robust to approximately 22% brightening, 20% darkening, 56% more saturation, 30% less saturation, $5 \times 5$ Gaussian filter, random spread pixels in a $14 \times 14$ neighborhood, and cropping 45%. The UFM scheme is extremely robust to horizontal and vertical image shifting.

## 5.3 Speed

The algorithm has been implemented on a Pentium III 700MHz PC running Linux operating system. Computing the feature vectors for $60,000$ color images of size $384 \times 256$ requires around 17 hours. On average, one second is needed to segment and compute the fuzzy features for an

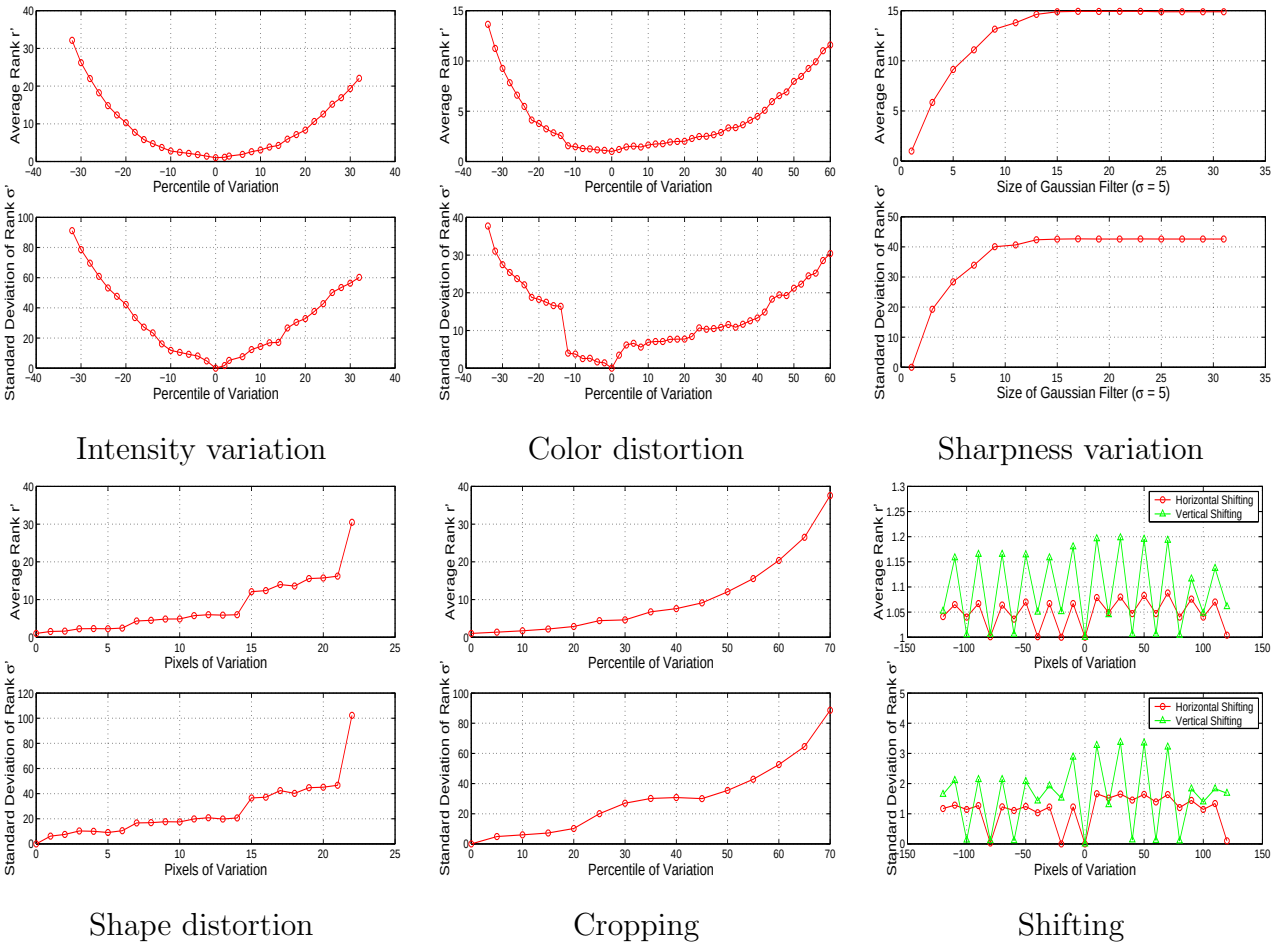



Fig. 7. The robustness of the UFM scheme to image alterations. Average rank $r'$ and standard deviation of rank $\sigma'$ are plotted against the intensity of image alterations.

image, which is the same as the speed of IRM. It is much faster than the Blobworld system [4], which, on average, takes about 5 minutes to segment a $128 \times 192$ image [3]. Fast segmentation speed provides us the ability of handling outside queries in real-time.

The time for matching images and sorting results in UFM is $O(C^2N + N \log N)$, where $N$ is the number of images in the database, $C$ is the average number of regions of an image. For our current database ($N = 60,000$ and $C = 4.3$), when the query image is in the database, it takes about 0.7 seconds of CPU time on average to compute and sort the similarities for all images in the database. If the query is not in the database, one extra second of CPU time is spent to process the query.

Based on 100 random runs, a quantitative comparison of the speed of UFM, IRM, and Blobworld systems is summarized in Table I where $t_s$ is the average CPU time for image

[3]The segmentation algorithm (in Matlab code) is tested on a 400MHz UltraSPARC IIi with the code obtained from `http://elib.cs.berkeley.edu/src/blobworld/`.

segmentation, $t_i$ is the average CPU time for computing similarity measures and indexing [4]. The UFM and IRM use the same database of $60,000$ images. The Blobworld system is tested on a database of $35,000$ images. Unlike IRM and UFM, the Blobworld system doesn't support outside queries. For inside queries, which do not require online image segmentation, UFM is 0.43 times faster than IRM, and 6.57 times faster than Blobworld.

TABLE I

Comparison of UFM, IRM, and Blobworld systems on average segmentation time $t_s$ and average indexing time $t_i$.

| **CBIR Systems** | **UFM** | **IRM** | **Blobworld** |
|---|---|---|---|
| $t_s$(second) | 1.1 | 1.1 | 292.5 |
| $t_i$(second) | 0.7 | 1.0 | 5.3 |
| Database Size | $60,000$ | $60,000$ | $35,000$ |

### 5.4 Comparison of Membership Functions

The UFM scheme is tested against different membership functions, namely the cone, exponential, and Cauchy functions. To make comparisons consistent, for a given region, we require the fuzzy features with different membership functions have identical 0.5-cuts. The 0.5-cut of a fuzzy feature is the set of feature vectors that have degrees of membership greater than or equal to 0.5. For a Cauchy function $\mathcal{C}(\vec{x}) = \frac{d^\alpha}{d^\alpha + \|\vec{x}-\vec{v}\|^\alpha}$, the above requirement can be easily satisfied by choosing the cone function as $\mathcal{T}(\vec{x}) = \max(1 - \frac{\|\vec{x}-\vec{v}\|^\alpha}{(2d)^\alpha}, 0)$ and the exponential function as $\mathcal{E}(\vec{x}) = e^{-\frac{\|\vec{x}-\vec{v}\|^\alpha}{(1.443d)^\alpha}}$.

Under an experiment setup identical to that of Section 5.2.2, the performance on image categorization is tested for three membership functions with parameter $\alpha$ varying from 0.1 to 2.0. The overall average precision $p$ is calculated according to (17). As shown in the upper plot in Figure 8, the highest $p$ for Cauchy and exponential membership functions, which is 0.477, occurs at $\alpha = 1.0$. The best $\alpha$ for the cone membership function is 0.8 with $p = 0.478$. So three membership functions generate almost the same maximum overall average precision. However, the computational complexities of three membership functions with corresponding

[4]Approximate execution times are obtained by issuing queries to the demonstration web sites `http://wang.ist.psu.edu/IMAGE/` (UFM and IRM) and `http://elib.cs.berkeley.edu/photos/blobworld/` (Blobworld). The web server for UFM and IRM is a 700MHz Pentium III PC, while the web server for Blobworld is unknown.

optimal $\alpha$ values are quite different. For any given $\|\vec{x} - \vec{v}\|$, the cone membership function needs to compute a power term $\left(\frac{\|\vec{x}-\vec{v}\|}{2d}\right)^{0.8}$. The exponential membership function needs to evaluate an exponential term $e^{-\frac{\|\vec{x}-\vec{v}\|}{1.443d}}$. Only two floating point operations are required by the Cauchy membership function. Based on the 60,000-image database, $t_i$ for three membership functions are plotted in the lower part of Figure 8. As expected, $t_i$ enlarges linearly with the increase of the number of regions in the query image and the Cauchy membership function produces the smallest $t_i$.

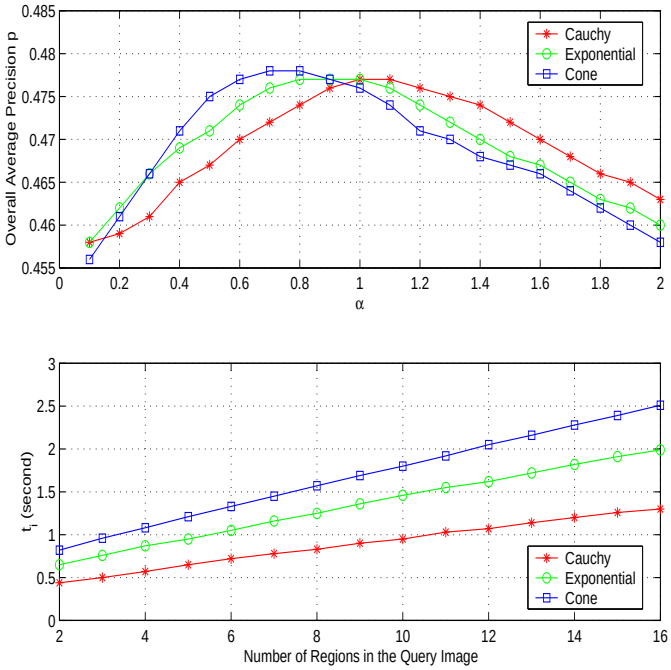



Fig. 8. Comparing the Cauchy, exponential, and cone membership functions on overall average precision $p$ and average CPU time $t_i$ for inside queries.

We also test the robustness to image alterations with respect to the type of membership function being used. For all six image alterations described in Section 5.2.4, the performances of exponential ($\alpha = 1.0$) and cone ($\alpha = 0.8$) membership functions are almost identical to that of the Cauchy ($\alpha = 1.0$) membership function in terms of $r'$ and $\sigma'$ defined by (22) and (23), respectively. The Cauchy membership function requires the least computational cost.

## 6 Conclusions and Future Work

We have developed the UFM, a novel region-based fuzzy feature matching approach for CBIR. In the UFM scheme, an image is first segmented into regions. Each region is then represented by a fuzzy feature that is determined by center location (a feature vector) and width (grade of

fuzziness). Compared with the conventional region representation using a single feature vector, each region is represented by a set of feature vectors each with a value denoting its degree of membership to the region. Consequently, the membership functions of fuzzy sets naturally characterize the gradual transition between regions within an image. That is, they characterize the blurring boundaries due to imprecise segmentation.

A direct consequence of fuzzy feature representation is the region-level similarity. Instead of using the Euclidean distance between two feature vectors, a fuzzy similarity measure, which is defined as the maximum value of the membership function of the intersection of two fuzzy features, is used to describe the resemblance of two regions. This value is always within $[0, 1]$ with a larger value indicating a higher degree of similarity between two fuzzy features. The value depends on both the Euclidean distance between the center locations and the grades of fuzziness of two fuzzy features. Intuitively, even though two fuzzy features are close to each other, if they are not "fuzzy" (i.e., the boundary between two regions is distinctive), then their similarity could be low. In the case that two fuzzy features are far away from each other, but they are very "fuzzy" (i.e., the boundary between two regions is very blurring), the similarity could be high. These correspond reasonably to the viewpoint of the human perception.

Trying to provide a comprehensive and robust "view" of similarity between images, the region-level similarities are combined into an image-level similarity vector pair, and then the entries of the similarity vectors are weighted and added up to produce the UFM similarity measure which depicts the overall resemblance of images in color, texture, and shape properties. The comprehensiveness and robustness of UFM measure can be examined from two perspectives namely the contents of similarity vectors and the way of combining them. Each entry of similarity vectors 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 a region is allowed to be matched with several regions in case of inaccurate image segmentation which in practice occurs quite often. By weighted summation, every fuzzy feature in both images contributes a portion to the overall similarity measure. This further reduces the sensitivity of UFM measure. The application of the UFM method to a database of about $60,000$ general-purpose images has demonstrated good accuracy and excellent robustness to image segmentation and image alterations.

A major limitation of the UFM scheme, which is inherent to the current fuzzy feature representation, is that the specificity is sacrificed to the robustness. The current system works well

for the testing image database that consists of 60,000 photographic pictures. However, experiments on a different image database (also available at the demonstration web site) of about 140,000 clip art pictures show that the IRM outperforms the UFM a little in accuracy. This is because, unlike photographs, segmentation of a clip art picture tends to be very accurate. Fuzzy features blur the boundaries of the originally clear-cut regions, which makes accurately recognizing and matching similar regions even harder.

The system may be improved in the following ways:

- The image segmentation algorithm may be improved. The current segmentation algorithm is very simple and efficient, and is able to cluster several similar objects in an image into a single region. But objects totally different in semantic may be clustered into the same region (as shown in Figure 5, when the image of bird is segmented into 3 regions, the bird, branch, and leaves are grouped into a single region).
- One of the advantages of region-based image retrieval methods is that the size, shape, and absolute and relative location of the regions can provide additional help. The current system utilizes the shape and size information, but the location is not fully exploited (only the relative locations of regions have influence on the border favored scheme of choosing weights).
- Under the current implementation, all fuzzy features within one image have the same shape. In reality, however, the grades of fuzziness of regions can be different even within an image. The UFM can be improved by allowing different shapes for fuzzy features in same image. Another potential enhancement to UFM is to use dynamic fuzzy features. That is, we can make the fuzzy features of the query image self-adaptable to the uncertainty level (e.g., entropy) of target images. This may provide more flexibility in dealing with semantically different images.

Currently, we are working on statistical modeling based image comparison, statistical feature-space selection, feature clustering scheme for region-based retrieval, and biomedical applications.

## Acknowledgments

This work was supported in part by the US National Science Foundation, the PNC Foundation, and SUN Microsystems. The authors would like to thank Jia Li, Department of Statistics, The Pennsylvania State University, for valuable discussions. They would also like to thank anonymous reviewers and the associate editor for their comments that led to improvements of the paper.

# References


[1] H. Bandemer and W. Nather, *Fuzzy Data Analysis*, Kluwer Academic Publishers, 1992.

[2] S. Berretti, A. Del Bimbo, and P. Pala, "Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing," *IEEE Trans. Multimedia*, vol. 2, no. 4, pp. 225–239, 2000.

[3] A. Del Bimbo and P. Pala, "Visual Image Retrieval by Elastic Matching of User Sketches," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 19, no. 2, pp. 121–132, 1997.

[4] C. Carson, M. Thomas, S. Belongie, J.M. Hellerstein, and J. Malik, "Blobworld: A System for Region-Based Image Indexing and Retrieval," *Proc. Visual Information Systems*, pp. 509–516, June 1999.

[5] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathomas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 20–37, 2000.

[6] I. Daubechies, *Ten Lectures on Wavelets*, Capital City Press, 1992.

[7] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz, "Efficient and Effective Querying by Image Content," *J. Intell. Inform. Syst.*, vol. 3, no. 3-4, pp. 231–262, 1994.

[8] A. Gersho, "Asymptotically Optimum Block Quantization," *IEEE Trans. Information Theory*, vol. 25, no. 4, pp. 373–380, 1979.

[9] T. Gevers and A. W. M. Smeulders, "PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 102–119, 2000.

[10] A. Gupta and R. Jain, "Visual Information Retrieval," *Commun. ACM*, vol. 40, no. 5, pp. 70–79, 1997.

[11] J.A. Hartigan and M.A. Wong, "Algorithm AS136: A k-means Clustering Algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[12] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods For Classification, Data Analysis and Image Recognition*, John Wiley & Sons, LTD, 1999.

[13] L. Jia and L. Kitchen, "Object-Based Image Similarity Computation Using Inductive Learning of Contour-Segment Relations," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 80–87, 2000.

[14] S. Kulkarni, B. Verma, P. Sharma, and H. Selvaraj, "Content Based Image Retrieval Using a Neuro-Fuzzy Technique," *Proc. IEEE Int'l Joint Conf. on Neural Networks*, pp. 846–850, July 1999.

[15] J. Li, J.Z. Wang, and G. Wiederhold, "Classification of Textured and Non-Textured Images Using Region Segmentation," *Proc. 7th Int'l Conf. on Image Processing*, pp. 754–757, September 2000.

[16] J. Li, J.Z. Wang, and G. Wiederhold, "IRM: Integrated Region Matching for Image Retrieval," *Proc. 8th ACM Int'l Conf. on Multimedia*, pp. 147–156, October 2000.

[17] W.Y. Ma and B.S. Manjunath, "NeTra: A Toolbox for Navigating Large Image Databases," *Proc. IEEE Int'l Conf. Image Processing*, pp. 568–571, 1997.

[18] W.Y. Ma and B.S. Manjunath, "EdgeFlow: A Technique for Boundary Detection and Image Segmentation," *IEEE Trans. Image Processing*, vol. 9, no. 8, pp. 1375–1388, 2000.

[19] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, and T.S. Huang, "Supporting Content-Based Queries over Images in MARS," *Proc. IEEE Int'l Conf. on Multimedia Computing and Systems*, pp. 632–633, June 1997.

[20] T.P. Minka and R.W. Picard, "Interactive Learning with a 'Society of Models'," *Pattern Recognition*, vol. 30, no. 4, pp. 565–581, 1997.

[21] A. Mojsilovic, J. Kovacevic, J. Hu, R.J. Safranek, and S.K. Ganapathy, "Matching and Retrieval Based on the Vocabulary and Grammar of Color Patterns," *IEEE Trans. Image Processing*, vol. 9, no. 1, pp. 38–54, 2000.

[22] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation for Image Databases," *Int'l J. Comput. Vis.*, vol. 18, pp. 233–254, 1996.

[23] Y. Rubner, L.J. Guibas, and C. Tomasi, "The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval," *Proc. DARPA Image Understanding Workshop*, pp. 661–668, May 1997.

[24] C. Schmid and R. Mohr, "Local Grayvalue Invariants for Image Retrieval," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 19, no. 5, pp. 530–535, 1997.

[25] G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases," *Proc. 24th Int'l Conf. on Very Large Data Bases*, pp. 428–439, August 1998.

[26] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[27] J.R. Smith and S.-F. Chang, "VisualSEEK: A Fully Automated Content-Based Query System," *Proc. ACM Multimedia*, pp. 87–98, 1996.

[28] J.R. Smith and C.S. Li, "Image Classification and Querying Using Composite Region Templates," *J. Comput. Vis. Image Understanding*, vol. 75, no. 1-2, pp. 165–174, 1999.

[29] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 18, no. 8, pp. 831–837, 1996.

[30] M. Unser, "Texture Classification and Segmentation Using Wavelet Frames," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1549–1560, 1995.

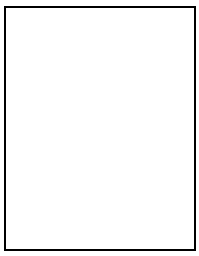
[31] C. Vertan and N. Boujemaa, "Embedding Fuzzy Logic in Content Based Image Retrieval," *Proc. 19th Int'l Meeting of the North American Fuzzy Information Processing Society NAFIPS 2000*, pp. 85–89, July 2000.

[32] J.Z. Wang, G. Wiederhold, O. Firschein, and X.W. Sha, "Content-Based Image Indexing and Searching Using Daubechies' wavelets," *Int'l J. Digital Libraries*, vol. 1, no. 4, pp. 311–328, 1998.

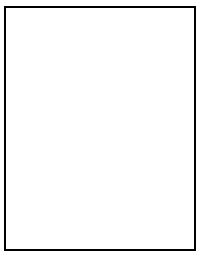
[33] J.Z. Wang, J. Li, R.M. Gray, and G. Wiederhold, "Unsupervised Multiresolution Segmentation for Images with Low Depth of Field," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 23, no. 1, pp. 85–91, 2001.

[34] J.Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 23, no. 9, pp. 947–963, 2001.

[35] S.C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Anal. and Machine Intell.*, vol. 18, no. 9, pp. 884–900, 1996.



**Yixin Chen** received the B.S. degree from the Department of Automation, Beijing Polytechnic University, China, in 1995, the M.S. degree in control theory and application from Tsinghua University, China, in 1998, and the M.S. and Ph.D. degrees in electrical engineering from the University of Wyoming, Laramie, WY, in 1999 and 2001, respectively. Since August 2000, he has been a Ph.D student in the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA. His research interests include machine learning, content-based image retrieval, computer vision, precision and fault tolerant robotic control, and soft computing. He is a student member of the IEEE, the IEEE Computer Society, and the IEEE Robotics and Automation Society.



**James Z. Wang** received the Summa Cum Laude bachelor's degree in mathematics and computer science from University of Minnesota (1994), the MSc degree in mathematics and the MSc degree in computer science, both from Stanford University (1997), and the PhD degree in medical information sciences from Stanford University Biomedical Informatics Program and Computer Science Database Group (2000). Since 2000, he has been the holder of the PNC Technologies Career Development Endowed Professorship and an assistant professor at the School of Information Sciences and Technology and the Department of Computer Science and Engineering at The Pennsylvania State University. He has been a visiting scholar at Uppsala University in Sweden, SRI International, IBM Almaden Research Center, and NEC Computer and Communications Research Lab. He is a member of the IEEE.