Looking Beyond Region Boundaries: A Robust Image Similarity Measure Using Fuzzified Region Features

Yixin Chen* and James Z. Wang†‡
*†Dept. of Computer Science and Engineering, ‡School of Information Sciences and Technology
The Pennsylvania State University, University Park, PA 16802
Email: *yixchen@cse.psu.edu, †jwang@ist.psu.edu

Abstract— **The performance of most region-based image retrieval systems depend critically on the accuracy of object segmentation. We propose a region matching approach, unified feature matching (UFM), which greatly increases the robustness of the retrieval system against segmentation related uncertainties. In our retrieval system, an image is represented by a set of segmented regions each of which is characterized by a fuzzy feature reflecting color, texture, and shape properties. The resemblance between two images is then defined as the overall similarity between two families of fuzzy features, and quantified by the UFM measure. The system has been tested on a database of about 60,000 general-purpose images. Experimental results demonstrate improved accuracy and robustness.**

I. Introduction

In many application fields such as biomedicine, the military, education, commerce, entertainment, crime prevention and World Wide Web searching, large volume of data appear in image form. As a result, it happens quite often that a customer or operator need to find relevant information from an image database based on the contents of the images. This problem has been known as content-based image retrieval (CBIR) for more than a decade. The difficulties and complexities in quantifying "meanings" of images via feature sets and designing efficient matching algorithms make CBIR a very challenging problem. There is a rich resource of prior work on this subject. The work that are most related to ours include IBM QBIC [4], MIT Photobook [12], Virage System [6], Columbia VisualSEEK [13], Stanford WBIIS [15], UIUC MARS [9], UCSB NeTra system [10], Berkeley Blobworld system [1], PicHunter system [3], PicToSeek system developed by Gevers *et al.* [5], and the work by Vertan and Boujemaa [14].

A central task in CBIR systems is similarity comparison. 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. A major drawback of the global histogram search lies in its sensitivity to intensity variations, color distortions, and cropping. The color layout indexing method is proposed to alleviate this drawback. The color layout of an image is essentially a low resolution representation of the image obtained by partitioning the image into blocks and taking the average or dominating color of each block. In general color layout searching is sensitive to shifting, cropping, scaling, and rotation [16].

Recognizing these deficiencies and the fact that human beings are capable of this technically complex performance, researchers try to relate human perception to CBIR systems. 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. Motivated by this intrinsic attribute of human visual perception, a region-based retrieval system applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal. Since the retrieval system has identified objects in the image, it is relatively easy for the system to recognize similar objects at different locations and with different orientations and sizes.

For general-purpose images such as the images in a photo library or the images on the World Wide Web, precise object segmentation is still an open problem in computer vision. Li *et al.* [7] propose an integrated region matching (IRM) scheme which allows for matching a region of one image to several regions of another image and thus decreases the impact of inaccurate segmentation by smoothing over the imprecision. The scheme is implemented in the SIMPLIcity system [16]. Nevertheless, the inaccuracies are not explicitly expressed in the IRM measure. Semantically precise image segmentation by an algorithm is very difficult [11][17]. 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 blurry. Based upon these observations, we hypothesize that, by softening the boundaries between regions, the robustness of a region-based image retrieval system against segmentation-related uncertainties can be improved.

II. The UFM Similarity Measure

A. Image Segmentation and Fuzzification

To segment an image, the system partitions the image into blocks with 4×4 pixels and extracts a feature vector for each block. We choose this block size to optimize between

texture effectiveness and segmentation coarseness. 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. The number of clusters (regions) is selected adaptively. Because clustering is performed in the feature space, blocks in each cluster does not have to be neighboring blocks in the images. This way, we preserve the natural clustering of objects and allow classification of textured images [8].

Six features are used for segmentation, as presented in [7]. Three of them are the average color components in a 4×4 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, that is, the square root of the second order moment of wavelet coefficients in high frequency bands obtained from a Daubechies-4 wavelet transform on the L component of the image block.

Here are some notations that will be used in the paper. $\mathbb{R}$ and $\mathbb{N}$ denote the sets of real numbers and positive integers, respectively. $\mathbb{R}^k$ represents a k dimensional Euclidean space. $\vec{f}_i \in \mathbb{R}^6$ is the feature vector (used for segmentation) of the ith block of an image. $\mathbf{F} = \{\vec{f}_i \in \mathbb{R}^6 : 1 \le i \le B, i \in \mathbb{N}\}$ is the set of all block feature vectors of an image, where B is the number of blocks in an image. Feature set $\mathbf{F}_j \subset \mathbf{F}$ contains all feature vectors in the jth cluster, where $1 \le j \le C, j \in \mathbb{N}$, $C \ge 2$ is the number of clusters. We also define the center of $\mathbf{F}_j$ (or jth cluster) as $\widehat{\vec{f}}_j = \frac{\sum_{\vec{f}:\vec{f}\in\mathbf{F}_j} \vec{f}}{\mathrm{V}(\mathbf{F}_j)}$ where $\mathrm{V}(\mathbf{F}_j)$ is the volume of $\mathbf{F}_j$. In general, $\widehat{\vec{f}}_j$ may not be an element of $\mathbf{F}_j$. $\mathbf{R}_j \subset \mathbb{N}^2$ is the region (set of pixels) corresponding to feature set $\mathbf{F}_j$.

To describe shape properties, three extra features are calculated for each region. They are normalized inertia 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 γ is given as $I_{(\mathbf{R}_j,\gamma)} = \frac{\sum_{(x,y):(x,y)\in\mathbf{R}_j}[(x-\hat{x})^2+(y-\hat{y})^2]^{\frac{\gamma}{2}}}{\mathrm{V}(\mathbf{R}_j)^{1+\frac{\gamma}{2}}}$, where $(\hat{x},\hat{y})$ is the centroid of $\mathbf{R}_j$. The normalized inertia is invariant to scaling and rotation. The minimum normalized inertia is achieved by spheres. Denote the γth 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., $\vec{h}_j = \left[\frac{I_{(\mathbf{R}_j,1)}}{\mathbf{I}_1}, \frac{I_{(\mathbf{R}_j,2)}}{\mathbf{I}_2}, \frac{I_{(\mathbf{R}_j,3)}}{\mathbf{I}_3}\right]^T$.

B. Fuzzy Feature Sets

After segmentation, an image can be viewed as a collection of regions. Equivalently, in the feature space, a segmented image is characterized by a collection of feature sets. These feature sets form a partition of $\mathbf{F}$, i.e., for any feature vector in $\mathbf{F}$, it belongs to exactly one feature set. However, segmentation can not be perfect. As a result, for many feature vectors, a unique decision between in and not in the feature set is impossible. Only a degree (between 0 and 1) of membership that it belongs to some feature set should be given, and it could belong to several feature sets with some possibly different degrees. Fuzzy set is a good description for this phenomenon.

Some commonly used prototype membership functions are cone, trapezoidal, B-splines, exponential, Cauchy, and paired sigmoid functions. We have 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. The exponential and Cauchy functions are comparable in performance. Considering the computational complexity, we pick the Cauchy function because it requires much less computations.

Thus we define the *membership function for the feature set* $\mathbf{F}_j$, $\mathcal{M}_{\mathbf{F}_j} : \mathbb{R}^6 \to [0,1]$, as

$$\mathcal{M}_{\mathbf{F}_j}(\vec{f}) = \frac{1}{1+\left(\frac{\|\vec{f}-\widehat{\vec{f}}_j\|}{\sigma_f}\right)^\alpha} \tag{1}$$

where $\sigma_f = \frac{2}{C(C-1)}\sum_{i=1}^{C-1}\sum_{k=i+1}^{C}\|\widehat{\vec{f}}_i - \widehat{\vec{f}}_k\|$ is the average distance between cluster centers. Similarly, the *membership function for the shape feature* $\vec{h}_j$, $\mathcal{M}_{\vec{h}_j} : \mathbb{R}^3 \to [0,1]$, is

$$\mathcal{M}_{\vec{h}_j}(\vec{h}) = \frac{1}{1+\left(\frac{\|\vec{h}-\vec{h}_j\|}{\sigma_h}\right)^\alpha} \tag{2}$$

where $\sigma_h = \frac{2}{C(C-1)}\sum_{i=1}^{C-1}\sum_{k=i+1}^{C}\|\vec{h}_i - \vec{h}_k\|$ is the average distance between shape features. The experiments show that the performance changes insignificantly when α is in the interval $[0.7, 1.5]$, but degrades rapidly outside the interval. So we set $\alpha = 1$ in both (1) and (2) to simplify the computation.

C. Unified Feature Matching

A direct consequence of fuzzy feature representation is the region-level similarity measure. Instead of using the Euclidean distance between two feature vectors, a fuzzy similarity measure is used to describe the resemblance of two regions. It is essentially the similarity between two fuzzy sets. Here we adopt the following definition.

Let $\mathbf{A}$ and $\mathbf{B}$ be fuzzy sets defined on $\mathbb{R}^k$ with corresponding membership functions $\mathcal{M}_\mathbf{A} : \mathbb{R}^k \to [0,1]$ and $\mathcal{M}_\mathbf{B} : \mathbb{R}^k \to [0,1]$, respectively. The intersection of $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A}\cap\mathbf{B}$, is a fuzzy set on $\mathbb{R}^k$ with membership function, $\mathcal{M}_{\mathbf{A}\cap\mathbf{B}} : \mathbb{R}^k \to [0,1]$, defined as

$$\mathcal{M}_{\mathbf{A}\cap\mathbf{B}}(\vec{x}) = \min[\mathcal{M}_\mathbf{A}(\vec{x}), \mathcal{M}_\mathbf{B}(\vec{x})]. \tag{3}$$

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

$$\mathcal{M}_{\mathbf{A}\cup\mathbf{B}}(\vec{x}) = \max[\mathcal{M}_\mathbf{A}(\vec{x}), \mathcal{M}_\mathbf{B}(\vec{x})]. \tag{4}$$

The similarity between $\mathbf{A}$ and $\mathbf{B}$, $\mathcal{S}(\mathbf{A},\mathbf{B}) \in [0,1]$, is given by

$$\mathcal{S}(\mathbf{A},\mathbf{B}) = \sup_{\vec{x}:\vec{x}\in\mathbb{R}^k} \mathcal{M}_{\mathbf{A}\cap\mathbf{B}}(\vec{x}). \quad (5)$$

For the fuzzy sets defined by Cauchy functions, calculating similarity according to (5) is relatively simple. This is because Cauchy function is symmetric and unimodal, and thus the maximum of (3) can only occur on the line segments connecting the center points of two functions. It is not hard to show that for fuzzy sets $\mathbf{A}$ and $\mathbf{B}$ on $\mathbb{R}^k$ with Cauchy membership functions $\mathcal{M}_{\mathbf{A}}(\vec{x}) = \frac{1}{1+\left(\frac{\|\vec{x}-\vec{u}\|}{\sigma_a}\right)^{\alpha}}$ and $\mathcal{M}_{\mathbf{B}}(\vec{x}) = \frac{1}{1+\left(\frac{\|\vec{x}-\vec{v}\|}{\sigma_b}\right)^{\alpha}}$, the similarity between $\mathbf{A}$ and $\mathbf{B}$ is

$$\mathcal{S}(\mathbf{A},\mathbf{B}) = \frac{(\sigma_a+\sigma_b)^{\alpha}}{(\sigma_a+\sigma_b)^{\alpha}+\|\vec{u}-\vec{v}\|^{\alpha}}. \quad (6)$$

The similarity between a region and an image can be computed from the region-level similarities. Let $\mathcal{F}_q = \{\mathbf{F}_j^q : 1 \le j \le C_q, j \in \mathbb{N}\}$ denote the collection of fuzzy sets for a query image segmented into C_q regions, and $\mathcal{F}_t = \{\mathbf{F}_j^t : 1 \le j \le C_t, j \in \mathbb{N}\}$ denote the collection of fuzzy sets for a target image segmented into C_t regions. First, for every $\mathbf{F}_j^q \in \mathcal{F}_q$, we define the similarity between it and $\mathcal{F}_t$ as

$$l_j^{(q,t)} = \mathcal{S}(\mathbf{F}_j^q, \bigcup_{i=1}^{C_t} \mathbf{F}_i^t). \quad (7)$$

Combining $l_j^{(q,t)}$'s together, we get a vector $\vec{l}^{(q,t)} = [l_1^{(q,t)}, l_2^{(q,t)}, \cdots, l_{C_q}^{(q,t)}]^T$. Similarly, the similarity between any $\mathbf{F}_j^t \in \mathcal{F}_t$ and $\mathcal{F}_q$ is

$$l_j^{(t,q)} = \mathcal{S}(\mathbf{F}_j^t, \bigcup_{i=1}^{C_q} \mathbf{F}_i^q), \quad (8)$$

and $\vec{l}^{(t,q)} = [l_1^{(t,q)}, l_2^{(t,q)}, \cdots, l_{C_t}^{(t,q)}]^T$. Finally, we define the similarity vector between $\mathcal{F}_q$ and $\mathcal{F}_t$, denoted by $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$, as

$$\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)} = \begin{bmatrix} \vec{l}^{(q,t)} \\ \vec{l}^{(t,q)} \end{bmatrix},$$

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

It can be shown that if two images are the same, $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ contains all 1's. If a fuzzy feature of one image is very different from all the fuzzy features of the other image, in the sense that the distances between their centers are large, the corresponding entry in $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ would be close to 0. Equations (7) and (8) can be equivalently computed by

$$l_j^{(q,t)} = \max_{i=1,\cdots,C_t} \mathcal{S}(\mathbf{F}_j^q, \mathbf{F}_i^t),\ l_j^{(t,q)} = \max_{i=1,\cdots,C_q} \mathcal{S}(\mathbf{F}_j^t, \mathbf{F}_i^q). \quad (9)$$

From (9), it is clear that the matching is in a *Winner Takes All* fashion. $\mathbf{F}_j^q$ ($\mathbf{F}_j^t$) is matched with all $\mathbf{F}_i^t$'s ($\mathbf{F}_i^q$'s), and the winner is the one with the maximum degree of membership.

Although $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ describes the similarity between the query and target images, it is certainly not handy for image retrieval. So we define the UFM similarity measure as the summation of all the weighted entries of $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ because each entry of $\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ quantifies the similarity between a fuzzy feature of the query (target) image and all fuzzy features of the target (query) image.

There are many ways of choosing weights. If we assume every region being equally important, then all weights equal to $\frac{1}{C_q+C_t}$ (the image with more regions is favored). If only the regions within the same image are equally important, then the weights for $\vec{l}^{(q,t)}$ and $\vec{l}^{(t,q)}$ are $\frac{1}{2C_q}$ and $\frac{1}{2C_t}$, respectively (the image with less regions is favored). We can also take the location of the regions into consideration, 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. In our current system, both area percentage and border favored schemes are used.

The whole UFM scheme used in our system is summarized as follows.

1. Image segmentation, fuzzification, and classification.
The query image is represented as $\mathcal{F}_q$ and $\mathcal{H}_q$. The target image is represented as $\mathcal{F}_t$ and $\mathcal{H}_t$. $\mathcal{F}_q$ and $\mathcal{F}_t$ are the classes of fuzzy sets which are defined according to (1). $\mathcal{H}_q$ and $\mathcal{H}_t$ are collections of fuzzy sets (containing shape properties) whose membership functions are consistent with (2)[1]. The query image is classified as either a textured or a non-textured image [8].
2. Fuzzy Region Matching.
$\vec{L}_{(\mathcal{F}_q,\mathcal{F}_t)}$ is calculated using equations (1,6,9).
3. Fuzzy Shape Matching.
$\vec{L}_{(\mathcal{H}_q,\mathcal{H}_t)}$ is calculated using equations (2,6,9). This step is needed only if the query image is classified as a non-textured image.
4. Similarity Measure calculation.
If the query image is textured, the similarity measure for two images, $m_{(q,t)}$, is

$$m_{(q,t)} = \frac{1}{2}\vec{L}^T_{(\mathcal{F}_q,\mathcal{F}_t)}[(1-\lambda)\vec{w} + \lambda\vec{\mu}].$$

[1]In our system, all images in the database have been segmented offline. A query image needs to go through this step only if it is not in the database.

Otherwise,

$$m_{(q,t)} = \frac{1-\delta}{2}\vec{L}^T_{(\mathcal{F}_q,\mathcal{F}_t)}[(1-\lambda)\vec{w} + \lambda\vec{\mu}] + \frac{\delta}{2}\vec{L}^T_{(\mathcal{H}_q,\mathcal{H}_t)}\vec{w}.$$

Here $\vec{w}$ is a vector containing the area percentages of the query and target images, $\vec{\mu}$ contains normalized weights which favor regions near the image boundary. The summation of all entries of $\vec{w}$ or $\vec{\mu}$ equals 2.

In this matching scheme, for non-textured query image, the overall measure reflects three types of similarities. $\vec{L}^T_{(\mathcal{F}_q,\mathcal{F}_t)}\vec{w}$ favors the matching of regions with larger areas. $\vec{L}^T_{(\mathcal{F}_q,\mathcal{F}_t)}\vec{\mu}$ characterizes the similarity of the backgrounds of two images. And $\vec{L}^T_{(\mathcal{H}_q,\mathcal{H}_t)}\vec{w}$ describes the similarity of the shape of the objects in both images. All of them are weighted by λ and δ which are in the real interval $[0,1]$. For texture images, the matching according to shape information is skipped since region shape is not perceptually important for such images. In our system, both λ and δ are chosen to be 0.1. When they are within $[0.05, 0.3]$, no major system performance deterioration is observed in our experiments. $m_{(q,t)}$ is always in the real interval $[0,1]$, and equals 1 if only if two images are same. In general, there is little resemblance between images if $m_{(q,t)} \leq 0.5$. So the UFM measure is very intuitive for query users.

III. Experimental Results

The system is implemented on a Pentium III 700 MHz PC running Linux operating system with a general-purpose image database including about 60,000 images. These images are stored in JPEG format with size 256×384 or 384×256. Each image is associated with a keyword describing the main subject of the image. About 20 research groups around the world have downloaded our image database [2] and use as a testbed for benchmarking purposes. In our current system all 60,000 images in the database are automatically classified into two semantic types: textured photograph, and non-textured photograph [8]. Feature extraction and image segmentation for all images in the database are performed off-line, which takes about 18 hours. On average, it takes about one second to segment and compute the fuzzy features for an image, and another 0.7 seconds to calculate and sort the UFM measures.

Compared with other region-based retrieval systems, such as [1], the query interface, which is shown in Figure 1, for our system is very simple. It provides a CGI-based Web access interface and a CGI-based outside query interface. The Web access interface is designed for query images which are in the database. The system provides a *Random* option that will give a user a random set of images from the database to start with. Users can also enter the ID of an image as the query. If the user moves the mouse on top of a thumbnail shown in the window, the thumbnail will

[2]Available at URL: `http://wang.ist.psu.edu/IMAGE`

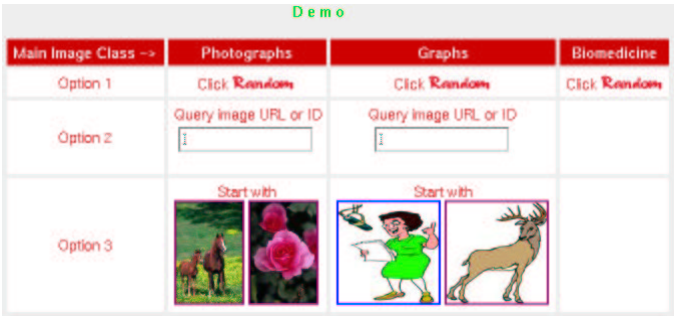


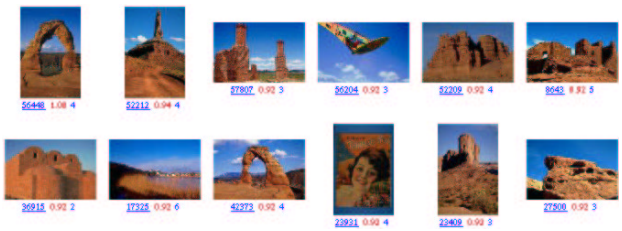
Fig. 1. The query interface.

be automatically switch to its region segmentation with each region painted with corresponding color. This feature is important for partial region matching since the user may choose a subset of the regions of an image to form a query rather than using the entire query image. The outside query interface allows the user to submit any image on the Internet as a query image to the system by entering the URL of the image. Our system is capable of handling any image format from anywhere on the Internet and reachable by our server via the HTTP protocol. The image is downloaded and processed by out system in real-time. The high efficiency of our image segmentation and matching algorithm made this feature possible.

To qualitatively illustrate 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, vehicle, cards, and people. For each query example, we manually exam the precision of the query results depending on the relevance of the image semantics. Due to space limitation, only top 11 matches to each query are shown in Figure 2. We also provide the number of relevant images among top 31 matches. 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 and the matched images, and the number of regions in the image.

The robustness of system is also tested with respect to image alterations such as intensity variation, sharpness variation, color distortion, shifting, and cropping. 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.

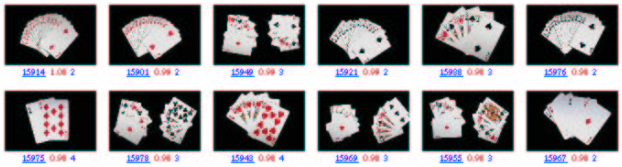
To provide more objective comparisons, the system performance is evaluated based on a subset of the COREL database, formed by 10 image categories, each containing 100 pictures. The categories are Africa, Beach, Buildings, Dinosaurs, Elephants, Flowers, Horses, Mountains, and Food. Within this database, it is known whether



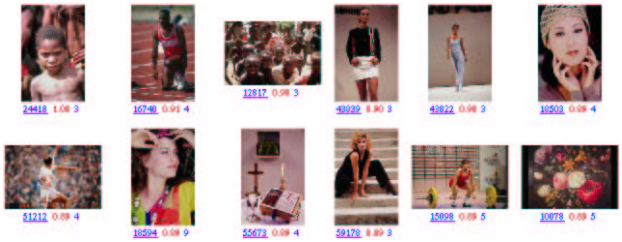
(a) Out-door scene; 9 matches out of 11; 21 matches out of 31

(b) Horses; 11 matches out of 11; 28 matches out of 31

(c) Vehicle; 10 matches out of 11; 24 matches out of 31



(d) Cards; 11 matches out of 11; 25 matches out of 31



(e) People; 10 matches out of 11; 23 matches out of 31

Fig. 2. The accuracy of the UFM measure.

(a) Brighten 33%

(b) Darken 25%

(c) 10% more saturated

(d) 9% less saturated

(e) Blur with a 20×20, $\sigma = 16$ Gaussian filter

(f) Sharpen with 5×5 filter

(g) Random spread 6 pixels

(h) 67% cropping

Fig. 3. The robustness of the UFM measure to image alterations. The query image is the first image in each example.

any two images are of the same category. In particular, a retrieved image is considered a 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 was tested as a query, and the retrieval ranks of all the rest images were recorded. Three statistics were computed for each query: the precision within the first 100 retrieval images, the mean rank of all the matched images, and the standard deviation of the ranks of matched images. We use entropy to characterize the segmentation-related uncertainties in an image. For an image with C segmented regions, its entropy is defined as $E_{\{\mathbf{R}_1,\cdots,\mathbf{R}_C\}} = -\sum_{j=1}^{C} P(\mathbf{R}_j) \log[P(\mathbf{R}_j)]$ where $P(\mathbf{R}_j)$ is the percentage of the image covered by region $\mathbf{R}_j$. The larger the value of entropy, the higher the uncertainty level.

To give a fair comparison between UFM and IRM at different uncertainty levels, we perform the same experiments for different average values of C. The performance in terms of overall average precision p, overall average mean rank r, and overall average standard deviation σ are evaluated for both approaches. The results are given in Figure 4. As we can see, the overall average entropy E increases when images are, on average, segmented into more regions. In

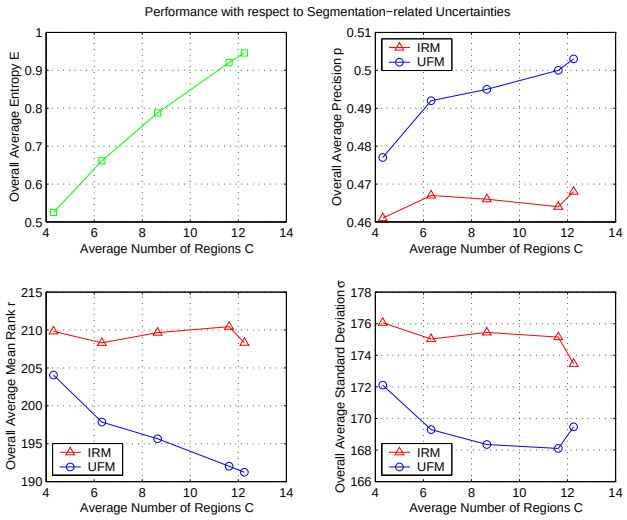


Fig. 4. Comparing the UFM scheme with the IRM method on the robustness to image segmentation: overall average entropy E, overall average precision p, overall average mean rank r, and overall average standard deviation σ.

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 σ. 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. 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.

IV. Conclusions

We have developed UFM, a image similarity measure for region-based image retrieval using fuzzified region representations. The UFM measure has two major advantages:

- Compared with retrieval methods based on individual regions, the UFM approach reduces the adverse effect of inaccurate segmentation, and make the retrieval system more robust to image alterations.
- Compared with overall similarity approaches, such as that proposed in [7], the UFM is better in extracting useful information under the same uncertain conditions.

Experiments have shown the accuracy of the scheme and the robustness to image segmentation and image alterations.

V. Acknowledgments

Some details about the algorithm appeared in [2]. The material about the SIMPLIcity system was based upon work supported by the National Science Foundation under Grant No. IIS-9817511 and in part by Stanford University. Work on the UFM system is supported by the National Science Foundation under Grant No. IIS-0219272, The Pennsylvania State University, the PNC Foundation, and SUN Microsystems under grant EDUD-7824-010456-US. The authors would like to thank Jia Li for valuable discussions.

References

[1] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and its Application to Image Querying," *IEEE Trans. Pattern Anal. Machine Intell.*, 24(8):1026–1038, 2002.

[2] Y. Chen and J.Z. Wang, "A Region-Based Fuzzy Feature Matching Approach to Content-Based Image Retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, 24:(9):1252–1267, 2002.

[3] 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*, 9(1):20–37, 2000.

[4] 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.*, 3(3-4):231–262, 1994.

[5] T. Gevers and A. W. M. Smeulders, "PicToSeek: Combining Color and Shape Invariant Features for Image Retrieval," *IEEE Trans. Image Processing*, 9(1):102–119, 2000.

[6] A. Gupta and R. Jain, "Visual Information Retrieval," *Commun. ACM*, 40(5):70–79, 1997.

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

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

[9] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, and T. S. Huang, "Supporting Content-Based Queries over Images in MARS," *Proc. IEEE Int. Conf. on Multimedia Computing and Systems*, pp. 632–633, 1997.

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

[11] W. Y. Ma and B. S. Manjunath, "EdgeFlow: A Technique for Boundary Detection and Image Segmentation," *IEEE Trans. Image Processing*, 9(8):1375–1388, 2000.

[12] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-Based Manipulation for Image Databases," *Int. J. Comput. Vis.*, 18(3):233–254, 1996.

[13] J. R. Smith and S.-F. Chang, "VisualSEEK: A Fully Automated Content-Based Query System," *Proc. ACM Multimedia'96*, pp. 87–98, 1996.

[14] 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, 2000.

[15] J. Z. Wang, G. Wiederhold, O. Firschein, and X. W. Sha, "Content-Based Image Indexing and Searching Using Daubechies' Wavelets," *Int. J. Digital Libraries*, 1(4):311–328, 1998.

[16] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries," *IEEE Trans. Pattern Anal. Machine Intell.*, 23(9):947–963, 2001.

[17] S. C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, 18(9):884–900, 1996.