Scalable Integrated Region-based Image Retrieval using IRM and Statistical Clustering*

James Z. Wang†
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16801
jwang@ist.psu.edu

Yanping Du‡
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA 16801
ydu@cse.psu.edu

ABSTRACT

Statistical clustering is critical in designing scalable image retrieval systems. In this paper, we present a scalable algorithm for indexing and retrieving images based on region segmentation. The method uses statistical clustering on region features and IRM (Integrated Region Matching), a measure developed to evaluate overall similarity between images that incorporates properties of all the regions in the images by a region-matching scheme. Compared with retrieval based on individual regions, our overall similarity approach (a) reduces the influence of inaccurate segmentation, (b) helps to clarify the semantics of a particular region, and (c) enables a simple querying interface for region-based image retrieval systems. The algorithm has been implemented as a part of our experimental SIMPLIcity image retrieval system and tested on large-scale image databases of both general-purpose images and pathology slides. Experiments have demonstrated that this technique maintains the accuracy and robustness of the original system while reducing the matching time significantly.

Keywords

Content-based image retrieval, wavelets, clustering, segmentation, integrated region matching.

1. INTRODUCTION

As multimedia information bases, such as the Web, become larger and larger in size, scalability of information retrieval system has become increasingly important. According to a report published by Inktomi Corporation and NEC Research in January 2000 [13], there are about 5 million unique Web sites (± 3%) on the Internet. Over one billion web pages (± 35%) can be down-loaded from these Web sites. Approximately one billion images can be found on-line. Searching for information on the Web is a serious problem [16, 17]. Moreover, the current growth rate of the Web is exponential, at an amazing 50% annual rate.

1.1 Image retrieval

Content-based image retrieval is the retrieval of relevant images from an image database based on automatically derived features. The need for efficient content-based image retrieval has increased tremendously in many application areas such as biomedicine, crime prevention, the military, commerce, culture, education, entertainment, and Web image classification and searching.

Content-based image retrieval has been widely studied. Space limitations do not allow us to present a broad survey. Instead we try to emphasize some of the work that is most related to what we propose. The references below are to be taken as examples of related work, not as the complete list of work in the cited area.

In the commercial domain, IBM QBIC [8, 25] is one of the earliest developed systems. Recently, additional systems have been developed at IBM T.J. Watson [34], VIRAGE [10], NEC C&C Research Labs [23], Bell Laboratory [24], Interpix (Yahoo), Excalibur, and Scour.net.

In academia, MIT Photobook [26, 27] is one of the earliest. Berkeley Blobworld [5], Columbia VisualSEEK and WebSEEK [33], CMU Informedia [35], University of Illinois MARS [22], University of California at Santa Barbara NeTra [21], the system developed by University of California at San Diego [14], Stanford WBIIS [36], and Stanford SIMPLIcity [38, 40]) are some of the recent systems.

Many indexing and retrieval methods have been used in these image retrieval systems. Some systems use keywords and full-text descriptions to index images. Others used features such as color histogram, color layout, local texture, wavelet coefficients, and shape to index images. In this paper, we focus on region-based retrieval of images.

*An on-line demonstration is provided at URL: `http://wang.ist.psu.edu`

†Also of Department of Computer Science and Engineering and e-Business Research Center. Research started when the author was with the Departments of Biomedical Informatics and Computer Science at Stanford University.

‡Also of Department of Electrical Engineering, . Now with Cisco Systems, Inc.

JCDL'01, June 24-28, 2001, Roanoke, Virginia, USA.

1.2 Region-based retrieval

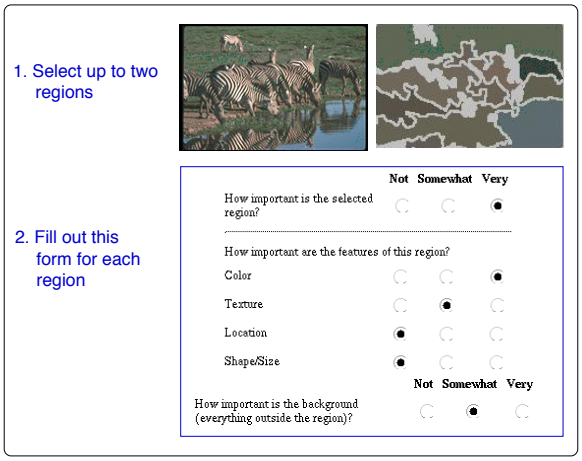


Figure 1: Query procedure of the Blobworld system developed at the University of California, Berkeley.

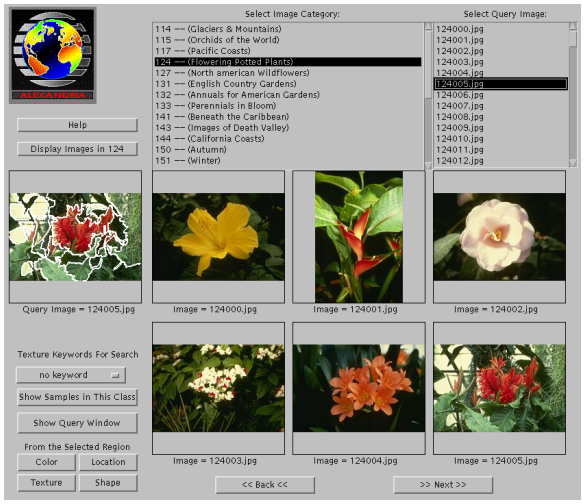


Figure 2: Query interface of the NeTra system developed at the University of California, Santa Barbara.

Before the introduction of region-based systems, content-based image retrieval systems used color histogram and color layout to index the content of images. Region-based approach has recently become a popular research trend. Region-based retrieval systems attempt to overcome the deficiencies of color histogram and color layout search by representing images at the object-level. A region-based retrieval system applies image segmentation to decompose an image into regions, which correspond to objects if the decomposition is ideal. The object-level representation is intended to be close to the perception of the human visual system (HVS).

Many earlier region-based retrieval systems match images based on individual regions. Such systems include the Netra system [21] and the Blobworld system [5]. Figures 1 and 2 show the querying interfaces of Blobworld and Netra. Querying based on a limited number of regions is allowed. The query is performed by merging single-region query results. The motivation is to shift part of the comparison task to the users. To query an image, a user is provided with the segmented regions of the image, and is required to select the regions to be matched and also attributes, e.g., color and texture, of the regions to be used for evaluating similarity. Such querying systems provide more control for the users. However, the query formulation process can be very time consuming.

1.3 Integrated region-based retrieval

Researchers are developing similarity measures that combine information from all of the regions. One effort in this direction is the querying system developed by Smith and Li [34]. Their system decomposes an image into regions with characterizations pre-defined in a finite pattern library. With every pattern labeled by a symbol, images are then represented by region strings. Region strings are converted to composite region template (CRT) descriptor matrices that provide the relative ordering of symbols. Similarity between images is measured by the closeness between the CRT descriptor matrices. This measure is sensitive to object shifting since a CRT matrix is determined solely by the ordering of symbols. Robustness to scaling and rotation is also not considered by the measure. Because the definition of the CRT descriptor matrix relies on the pattern library, the system performance depends critically on the library. The performance degrades if region types in an image are not represented by patterns in the library. The system in [34] uses a CRT library with patterns described only by color. In particular, the patterns are obtained by quantizing color space. If texture and shape features are used to distinguish patterns, the number of patterns in the library will increase dramatically, roughly exponentially in the number of features if patterns are obtained by uniformly quantizing features.

Li et al. of Stanford University recently developed SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) [37]. SIMPLIcity uses semantics type classification and an integrated region matching (IRM) scheme to provide efficient and robust region-based image matching [18]. The IRM measure is a similarity measure of images based on region representations. It incorporates the properties of all the segmented regions so that information about an image can be fully used. With IRM, region-based image-to-image matching can be performed. The overall similarity approach reduces the adverse effect of inaccurate segmentation, helps to clarify the semantics of a particular region, and enables a *simple* querying interface for region-based image retrieval systems. Experiments have shown that IRM is comparatively more effective and more robust than many existing retrieval methods. Like other region-based systems, the SIMPLIcity system is a linear matching system. To perform a query, the system compares the query image with all images in the same semantic class.

1.4 Statistical clustering

There are many efforts made to statistically cluster the high dimensional feature space before the actual searching using various tree structures such as K-D-B-tree [28], quadtree [9] R-tree [11], R^+-tree [31], R^*-tree [1], X-tree [3], SR-tree [15], M-tree [6], TV-tree [19], and hB-tree [20]. As mentioned in [4, 2, 1, 15, 41], the speed and accuracy of these algorithms degrade in very high dimensional spaces.

This is referred to as the *curse of dimensionality*. Besides, many of the clustering and indexing algorithms are designed for general purpose feature spaces such as Euclidean space. We developed our own algorithm for clustering and indexing image databases because we wanted the system to be suitable to our IRM region matching scheme.

1.5 Overview

In this paper, we present an enhancement to the SIMPLIcity system for handling image libraries with million of images. The targeted applications include Web image retrieval and biomedical image retrieval. Region features of images in the same semantic class are clustered automatically using a statistical clustering method. Features in the same cluster are stored in the same file for efficient access during the matching process. IRM (Integrated Region Matching) is used in the query matching process. Tested on large-scale image databases, the system has demonstrated high accuracy, robustness, and scalability.

The remainder of the paper is organized as follows. In Section 2, the similarity matching process based on segmented regions is defined. In Section 3, we describe the experiments we performed and provide results. We discuss limitations of the system in Section 4. We conclude in Section 5.

2. THE SIMILARITY MEASURE

In this section, we describe the similarity matching process we developed. We briefly describe the segmentation process and related notations in Section 2.1. The feature space analysis process is described in Section 2.2. In Section 2.3, we give details of the matching scheme.

2.1 Region segmentation

Semantically-precise image segmentation is extremely difficult and is still an open problem in computer vision [32, 39]. We attempt to develop a robust matching metric that can reduce the adverse effect of inaccurate segmentation. The segmentation process in our system is very efficient because it is essentially a wavelet-based fast statistical clustering process on blocks of pixels.

To segment an image, we partitions the image into blocks with $t \times t$ pixels and extracts a feature vector for each block. 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. We dynamically determine k by starting with $k = 2$ and refine if necessary to $k = 4$, etc. k is dynamically determined based on the complexity of the image. We do not require the clusters to be locationally contiguous because we rely on a robust matching process. The details of the segmentation process is described in [18].

Six features are used for segmentation. Three of them are the average color components in a $t \times t$ block. The other three represent energy in high frequency bands of wavelet transforms [7], that is, the square root of the second order moment of wavelet coefficients in high frequency bands. We use the well-known LUV color space, where L encodes luminance, and U and V encode color information (chrominance). The LUV color space has good perception correlation properties. We chose the block size t to be 4 to compromise between the texture detail and the computation time.

Let N denote the total number of images in the image database. For the i-th image, denoted as R_i, in the database, we obtain a set of n_i feature vectors after the region segmentation process. Each of these n_i d-dimensional feature vectors represents the dominant visual features (including color and texture) of a region, the shape of that region, the rough location in the image, and some statistics of the features obtained in that region.

2.2 Feature space analysis

The new integrated region matching scheme depends on the entire picture library. We must first process and analyze the characteristics of the d-dimensional feature space.

Suppose feature vectors in the d-dimensional feature space are $\{x_i : i = 1, ..., L\}$, where L is the total number of regions in the picture library. Then $L = \sum_{i=1}^{N} n_i$.

The goal of the feature clustering algorithm is to partition the features into k groups with centroids $\hat{x}_1, \hat{x}_2, ..., \hat{x}_k$ such that

$$D(k) = \sum_{i=1}^{L} \min_{1 \le j \le k} (x_i - \hat{x}_j)^2 \quad (1)$$

is minimized. That is, the average distance between a feature vector and the group with the nearest centroid to it is minimized. Two necessary conditions for the k groups are:

1. Each feature vector is partitioned into the cluster with the nearest centroid to it.
2. The centroid of a cluster is the vector minimizing the average distance from it to any feature vector in the cluster. In the special case of the Euclidean distance, the centroid should be the mean vector of all the feature vectors in the cluster.

These requirements of our feature grouping process are the same requirements as those of the Lloyd algorithm [12] to find k cluster means with the following steps:

1. Initialization: choose the initial k cluster centroids.
2. Loop until the stopping criterion is met:
   (a) For each feature vector in the data set, assign it to a class such that the distance from this feature to the centroid of that cluster is minimized.
   (b) For each cluster, recalculate its centroid as the mean of all the feature vectors partitioned to it.

If the Euclidean distance is used, the k-means algorithm results in hyper-planes as cluster boundaries (Figure 3. That is, for the feature space $\mathbb{R}^d$, the cluster boundaries are hyperplanes in the $d-1$ dimensional space $\mathbb{R}^{d-1}$.

Both the initialization process and the stopping criterion are critical in the process. We initialize the algorithm adaptively by choosing the number of clusters k by gradually increasing k and stop when a criterion is met. We start with $k = 2$. The k-means algorithm terminates when no more feature vectors are changing classes. It can be proved that the k-means algorithm is guaranteed to terminate, based on the fact that both steps of k-means (i.e., assigning vectors to nearest centroids and computing cluster centroids) reduce the average class variance. In practice, running to completion may require a large number of iterations. The cost for each iteration is $O(kn)$, for the data size n. Our stopping criterion is to stop after the average class variance is smaller than a threshold or after the reduction of the class variance is smaller than a threshold.

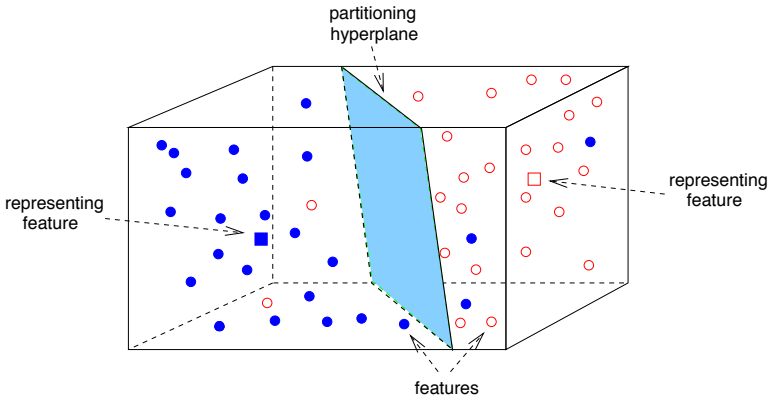


Figure 3: The k-means algorithm partitions the feature space using hyper-planes.

2.3 Image matching

To retrieve similar images for a query image, we first locate the clusters of the feature space to which the regions of the query image belong. Let's assume that the centroids of the set of k clusters are $\{c_1, c_2, ..., c_k\}$. We assume the query image is represented by region sets $R_1 = \{r_1, r_2, ..., r_m\}$, where r_i is the descriptor of region i. For each region feature r_i, we find j such that

$$d(r_i, c_j) = \min_{1 \le l \le k} d(r_i, c_l)$$

where $d(r_1, r_2)$ is the region-to-region distance defined for the system. This distance can be a non-Euclidean distance. We create a list of clusters, denoted as $\{c_{r_1}, c_{r_2}, ..., c_{r_k}\}$. The matching algorithm will further investigate only these 'suspect' clusters to answer the query.

With the list of 'suspect' clusters, we create a list of 'suspect' images. An image in the database is a 'suspect' image to the query if the image contains at least one region feature in these 'suspect' clusters. This step can be accomplished by merging the cluster image IDs non-repeatedly.

To define the similarity measure between two sets of regions, we assume that the image R_1 and image R_2 are represented by region sets $R_1 = \{r_1, r_2, ..., r_m\}$ and $R_2 = \{r'_1, r'_2, ..., r'_n\}$, where r_i or r'_i is the descriptor of region i. Denote the distance between region r_i and r'_j as $d(r_i, r'_j)$, which is written as $d_{i,j}$ in short. To compute the similarity measure between region sets R_1 and R_2, $d(R_1, R_2)$, we first compute all pair-wise region-to-region distances in the two images. Our matching scheme aims at building correspondence between regions that is consistent with our perception. To increase robustness against segmentation errors, we allow a region to be matched to several regions in another image. A matching between r_i and r'_j is assigned with a significance credit $s_{i,j}$, $s_{i,j} \ge 0$. The significance credit indicates the importance of the matching for determining similarity between images. The matrix $S = \{s_{i,j}\}$, $1 \le i \le n$, $1 \le j \le m$, is referred to as the significance matrix.

The distance between the two region sets is the summation of all the weighted matching strength, i.e.,

$$d_{IRM}(R_1, R_2) = \sum_{i,j} s_{i,j} d_{i,j} \ .$$

This distance is the integrated region matching (IRM) distance defined by Li et al. in [18].

To choose the significance matrix S, a natural issue to raise is what constraints should be put on $s_{i,j}$ so that the admissible matching yields good similarity measure. In other words, what properties do we expect an admissible matching to possess? The first property we want to enforce is the fulfillment of significance. Assume that the significance of r_i in Image 1 is p_i, and r'_j in Image 2 is p'_j, we require that

$$\sum_{j=1}^{n} s_{i,j} = p_i, \ i = 1, ..., m$$

$$\sum_{i=1}^{m} s_{i,j} = p'_j, \ j = 1, ..., n \ .$$

A greedy scheme is developed to speed up the determination of the matrix $S = \{s_{i,j}\}$. Details of the algorithm can be found in [18].

2.4 The RF*IPF weighting

For applications such as biomedical image retrieval, local feature is critically important in distinguishing the semantics between two images. In this section, we present the Region Frequency and Inverse Picture Frequency (RF*IPF) weighting, a relatively simple weighting measure developed to further enhance the discriminating efficiency of IRM based on the characteristics of the entire picture library. This weighting can be used to emphasize uncommon features.

The definition of RF*IPF is in some way close to the definition of the Term Frequency and Inverse Document Frequency (TF*IDF) weighting [30], a highly effective techniques in document retrieval. The combination of RF*IPF and IRM is more effective than the IRM itself in a variety of image retrieval applications. Additionally, this weighting measure provides a better unification of content-based image retrieval and text-based image retrieval.

The RF*IPF weighting consists of two parameters: the Region Frequency (RF) and the Inverse Picture Frequency (IPF).

For each region feature vector x_i of the image R_j, we find the closest group centroid from the list of k centroids computed in the feature analysis step. That is, we find c_0 such that

$$\| x_i - \hat{x}_{c_0} \| = \min_{1 \le c \le k} \| x_i - \hat{x}_c \| \ . \qquad (2)$$

Let's denote N_{c_0} as the number of pictures in the database

with at least one region feature closest to the centroid $\hat{x}_{c_0}$ of the image group c_0. Then we define

$$IPF_i = log\left(\frac{N}{N_{c_0}}\right) + 1 \quad (3)$$

where IPF_i is the Inverse Picture Frequency of the feature x_i.

Now let's denote M_j as the total number of pixels in the image R_j. For images in a size-normalized picture library, M_j are constants for all j. Denote $P_{i,j}$ as the area percentage of the region i in the image R_j. Then, we define

$$RF_{i,j} = log(P_{i,j}M_j) + 1 \quad (4)$$

as the Region Frequency of the i-th region in picture j. Then RF measures how frequently a region feature occurs in a picture.

We can now assign a weight for each region feature in each picture. The RF*IPF weight for the i-th region in the j-th image R_j is defined as

$$W_{i,j} = RF_{i,j} * IPF_i \ . \quad (5)$$

Clearly, the definition is close to that of the TF*IDF (Term Frequency times Inverse Document Frequency) weighting in text retrieval.

After computing the RF*IPF weights for all the L regions in all the N images in the image database, we store these weights for the image matching process.

We now combine the IRM distance with the RF*IPF weighting in the process of choosing the significance matrix S. A natural issue to raise is what constraints should be put on $s_{i,j}$ so that the admissible matching yields good similarity measure. In other words, what properties do we expect an admissible matching to possess? The first property we want to enforce is the fulfillment of significance. We computed the significance W_{i,R_1} of r_i in image R_1 and r'_j in image R_2 is W_{j,R_2}, we require that

$$\sum_{j=1}^{n} s_{i,j} = p_i = \frac{W_{i,R_1}}{\sum_{l=1}^{m} W_{l,R_1}},\ i = 1, ..., m$$

$$\sum_{i=1}^{m} s_{i,j} = q_j = \frac{W_{j,R_2}}{\sum_{l=1}^{n} W_{l,R_2}},\ j = 1, ..., n \ .$$

3. EXPERIMENTS

This algorithm has been implemented and compared with the first version of our experimental SIMPLIcity image retrieval system. We tested the system on a general-purpose image database (from COREL) including about 200,000 pictures, which are stored in JPEG format with size 384×256 or 256×384. To conduct a fair comparison, we use only picture features in the retrieval process.

3.1 Speed

On a Pentium III 800MHz PC using the Linux operating system, it requires approximately 60 hours to compute the feature vectors for the 200,000 color images of size 384×256 in our general-purpose image database. On average, one second is needed to segment an image and to compute the features of all regions. Fast indexing has provided us with the capability of handling outside queries and sketch queries in real-time.

The feature clustering process is performed only once for each database. The Lloyd algorithm takes about 30 minutes CPU time and results in clusters with an average of 1100 images. Our image segmentation process generates an average of 4.6 regions per image. That is, on average a 'suspect' list for a query image contains at most $1100 \times 4.6 = 5060$ images.

Category	IRM	fast IRM	EMD2	EMD 1
1. Africa	0.475	0.472	0.288	0.132
2. Beach	0.325	0.323	0.286	0.134
3. Buildings	0.330	0.307	0.233	0.160
4. Buses	0.363	0.389	0.267	0.108
5. Dinosaurs	0.981	0.635	0.914	0.143
6. Elephants	0.400	0.390	0.384	0.169
7. Flowers	0.402	0.447	0.416	0.113
8. Horses	0.719	0.669	0.386	0.096
9. Mountains	0.342	0.335	0.218	0.198
10. Food	0.340	0.340	0.207	0.114

Table 1: The average performance for each image category evaluated by average precision (p).

The matching speed is fast. When the query image is in the database, it takes about 0.15 seconds of CPU time on average to sort all the images in the 200,000-image database using our similarity measure. This is a significant speed-up over the original system which runs at 1.5 second per query. If the query is not in the database, one extra second of CPU time is spent to process the query.

Figures 4 and 5 show the results of sample queries. Due to the limitation of space, we show only two rows of images with the top 11 matches to each query. In the next section, we provide numerical evaluation results by systematically comparing several systems.

Because of the fast indexing and retrieval speed, we allow the user to submit any images on the Internet as a query image to the system by entering the URL of an image (Figure 6). 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 our system on-the-fly. The high efficiency of our image segmentation and matching algorithms made this feature possible[1]. To our knowledge, this feature of our system is unique in the sense that no other commercial or academic systems allow such queries.

3.2 Accuracy on image categorization

We conducted extensive evaluation of the system. One experiment was based on a subset of the COREL database, formed by 10 image categories, each containing 100 pictures. Within this database, it is known whether 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.

For each query, we computed the precision within the first 100 retrieved images. The recall within the first 100 retrieved images was not computed because it is proportional to the precision in this special case. The total number of se-

[1]It takes some other region-based CBIR system [5] approximately 8 minutes CPU time to segment an image.

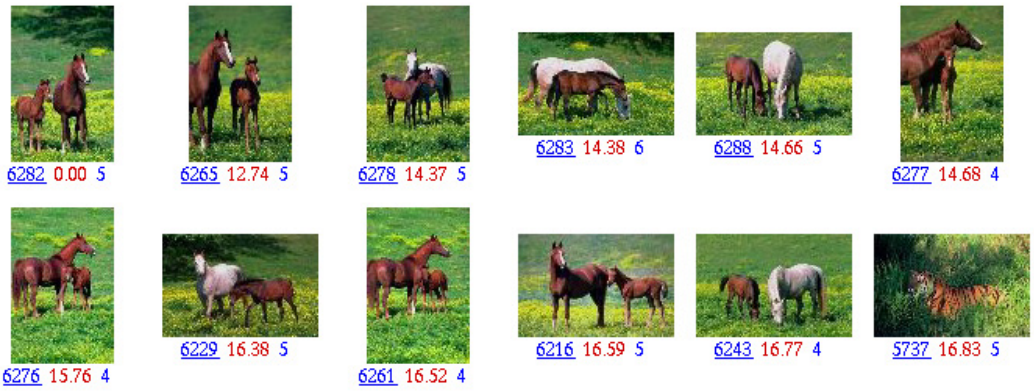
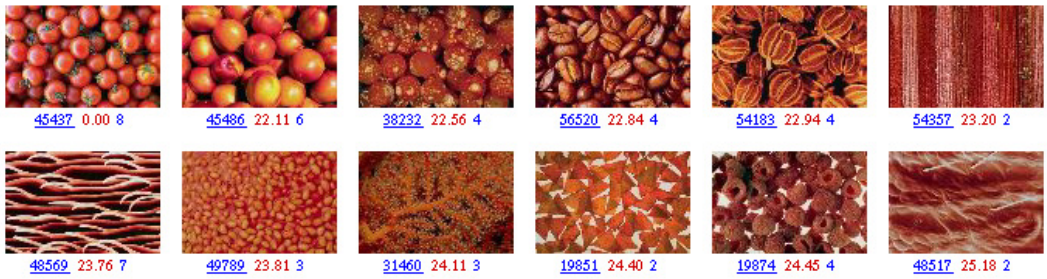


Figure 4: Best 11 matches of a sample query. The database contains 200,000 images from the COREL image library. The upper left corner is the query image. The second image in the first row is the best match.

(a)



(b)

Figure 5: Two other query examples.

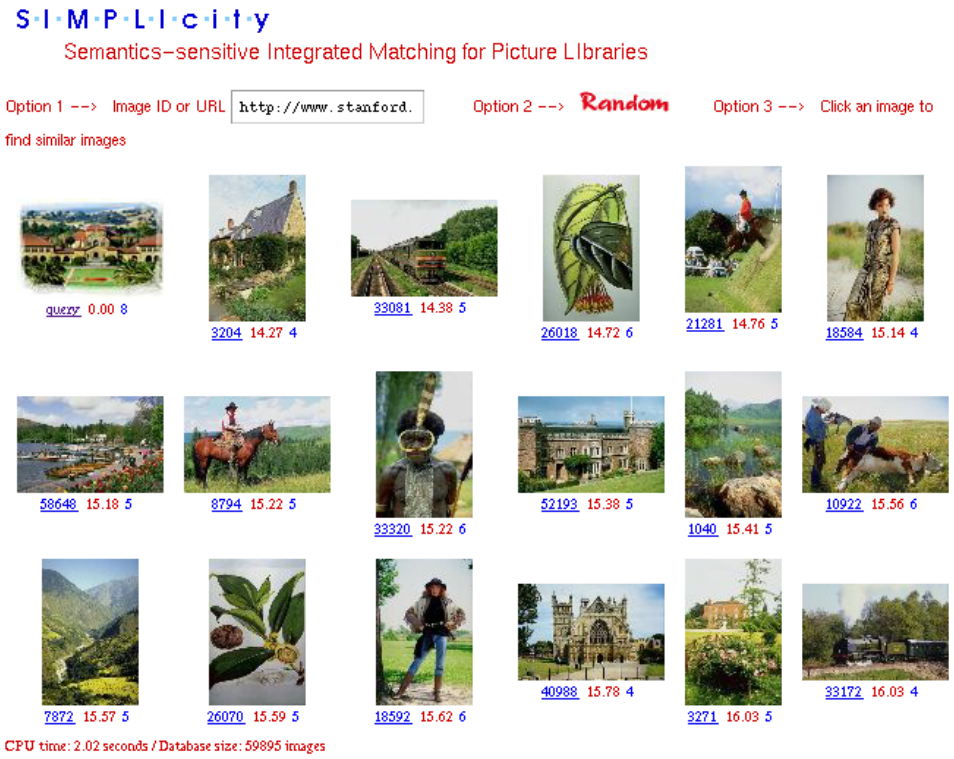


Figure 6: The external query interface. The best 17 matches are presented for a query image selected by the user from the Stanford top-level Web page. The user enters the URL of the query image (shown in the upper-left corner, `http://www.stanford.edu/home/pics/h-quad.jpg`) to form a query.

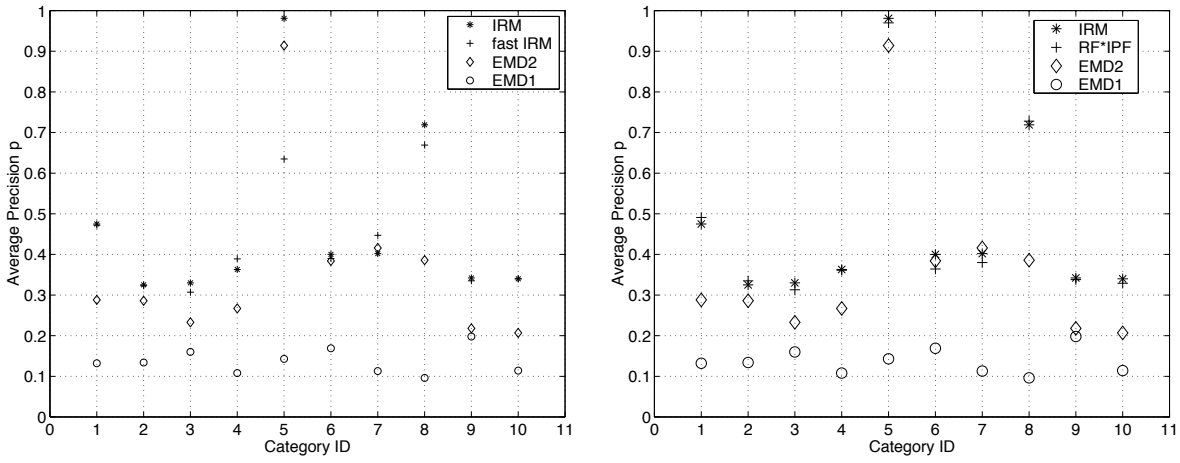


Figure 7: Comparing with color histogram methods on average precision p. Color Histogram 1 gives an average of 13.1 filled color bins per image, while Color Histogram 2 gives an average of 42.6 filled color bins per image. SIMPLIcity partitions an image into an average of only 4.3 regions.

mantically related images for each query is fixed to be 100. The average performance for each image category in terms of the average precision is listed in Table 1, where p denotes precision. For a system that ranks images randomly, the average p is about 0.1.

We carried out similar evaluation tests for color histogram match. We used LUV color space and a matching metric similar to the EMD described in [29] 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, were evaluated. We call the one with less filled color bins the Color Histogram 1 system and the other the Color Histogram 2 system. Figure 7 shows the performance as compared with the Lloyd-based SIMPLIcity system. Clearly, both of the two color histogram-based matching systems perform much worse than the Lloyd-based system in almost all image categories. The performance of the Color Histogram 2 system is better than that of the Color Histogram 1 system due to more detailed color separation obtained with more filled bins. However, the Color Histogram 2 system is so slow that it is impossible to obtain matches on larger databases. The original SIMPLIcity runs at about twice the speed of the faster Color Histogram 1 system and gives much better searching accuracy than the slower Color Histogram 2 system.

The overall performance of the Lloyd-based system is close to that of the original system which uses IRM and area percentages of the segmented regions as significant constraints. Both the regular IRM and the fast IRM algorithms are much more accurate than the EMD-based color histogram. Experiments on a database of 70,000 pathology slides demonstrated similar comparison results.

3.3 Robustness

Similar to the original SIMPLIcity system [38], the current system is exceptionally robust to image alterations such as intensity variation, sharpness variation, intentional color distortions, intentional shape distortions, cropping, shifting, and rotation.

The system is fairly robust to image alterations such as intensity variation, sharpness variation, intentional color distortions, other intentional distortions, cropping, shifting, and rotation. On average, the system is robust to approximately 10% brightening, 8% darkening, blurring with a 15×15 Gaussian filter, 70% sharpening, 20% more saturation, 10% less saturation, random spread by 30 pixels, and pixelization by 25 pixels. These features are important to biomedical image databases because usually visual features of the query image are not identical to the visual features of those semantically-relevant images in the database because of problems such as occlusion, difference in intensity, and difference in focus.

4. DISCUSSIONS

The system has several limitations. (1) Like other CBIR systems, SIMPLIcity assumes that images with similar semantics share some similar features. This assumption may not always hold. (2) The shape matching process is not ideal. When an object is segmented into many regions, the IRM distance should be computed after merging the matched regions. (3) The querying interfaces are not powerful enough to allow users to formulate their queries freely. For different user domains (e.g., biomedicine, Web image retrieval), the query interfaces should ideally provide different sets of functions.

In our current system, the set of features for a particular image category is determined empirically based on the perception of the developers. For example, shape-related features are not used for textured images. Automatic derivation of optimal features is a challenging and important issue in its own right. A major difficulty in feature selection is the lack of information about whether any two images in the database match with each other. The only reliable way to obtain this information is through manual assessment, which is formidable for a database of even moderate size. Furthermore, human evaluation is hard to be kept consistent from person to person. To explore feature selection, primitive studies can be carried with relatively small databases. A database can be formed from several distinctive groups of images, among which only images from the same group are considered matched. A search algorithm can be developed to select a subset of candidate features that provides optimal retrieval according to an objective performance measure. Although such studies are likely to be seriously biased, insights regarding which features are most useful for a certain image category may be obtained.

The main limitation of our current evaluation results is that they are based mainly on precision or variations of precision. In practice, a system with a high overall precision may have a low overall recall. Precision and recall often trade off against each other. It is extremely time-consuming to manually create detailed descriptions for all the images in our database in order to obtain numerical comparisons on recall. The COREL database provides us rough semantic labels on the images. Typically, an image is associated with one keyword about the main subject of the image. For example, a group of images may be labeled as "flower" and another group of images may be labeled as "Kyoto, Japan". If we use the descriptions such as "flower" and "Kyoto, Japan" as definitions of relevance to evaluate CBIR systems, it is unlikely that we can obtained a consistent performance evaluation. A system may perform very well on one query (such as the flower query), but very poorly on another (such as the Kyoto query). Until this limitation is thoroughly investigated, the evaluation results reported in the comparisons should be interpreted cautiously.

5. CONCLUSIONS AND FUTURE WORK

We have developed a scalable integrated region-based image retrieval system. The system uses the IRM measure and the Lloyd algorithm. The algorithm has been implemented as part of the the IRM metric in our experimental SIMPLIcity image retrieval system. Tested on a database of about 200,000 general-purpose images, the technique has demonstrated high efficiency and robustness. The main difference between this system and the previous SIMPLIcity system is the statistical clustering process which significantly reduces the computational complexity of the IRM measure.

The clustering efficiency can be improved by using a better statistical clustering algorithm. Better statistical modeling and matching scheme is likely to improve the matching accuracy of the system. We are also planning to apply the methods to special image databases (e.g., biomedical), and very large multimedia document databases (e.g., WWW, video).

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation's Digital Library II initiative under Grant No. IIS-9817511 and an endowment from the PNC Foundation. The authors wish to thank the help of Jia Li, Xiaoming Huo, Gio Wiederhold, Oscar Firschein, and anonymous reviewers.

7. REFERENCES

[1] N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," *Proc. ACM SIGMOD*, pp. 322-331, Atlantic City, NJ, 23-25 May 1990.

[2] J. Bentley, J. Friedman, "Data structures for range searching," *ACM Computing Surveys*, vol. 11, no. 4, pp. 397-409, December 1979.

[3] S. Berchtold, D. Keim, H.-P. Kriegel, "The X-tree: An index structure for high-dimensional data," *Proc. Int. Conf. on Very Large Databases*, pp. 28-39, 1996.

[4] S. Berchtold, C. Bohm, B. Braunmuller, D. Keim, H.- P. Kriegel, "Fast parallel similarity search in multimedia databases," *Proc. ACM SIGMOD*, pp. 1-12, Tucson, AZ, 1997.

[5] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, J. Malik, "Blobworld: a system for region-based image indexing and retrieval," *Proc. Int. Conf. on Visual Information Systems*, D. P. Huijsmans, A. W.M. Smeulders (eds.), Springer, Amsterdam, The Netherlands, June 2-4, 1999.

[6] P. Ciaccia, M. Patella, P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," *Proc. Int. Conf. on Very Large Databases*, Athens, Greece, 1997.

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

[8] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, W. Equitz, "Efficient and effective querying by image content," *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies*, vol. 3, no. 3-4, pp. 231-62, July 1994.

[9] R. Finkel, J. Bentley, "Quad-trees: A data structure retrieval on composite keys," *ACTA Informatica*, vol. 4, no. 1, pp. 1-9, 1974.

[10] A. Gupta, R. Jain, "Visual information retrieval," *Communications of the ACM*, vol. 40, no. 5, pp. 70-79, May 1997.

[11] A. Guttman, "R-trees: A dynamic index structure for spatial searching," *Proc. ACM SIGMOD*, pp. 47-57, Boston, MA, June 1984.

[12] J. A. Hartigan, M. A. Wong, "Algorithm AS136: a k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100-108, 1979.

[13] "Web surpasses one billion documents," *Inktomi Corporation Press Release*, January 18, 2000.

[14] R. Jain, S. N. J. Murthy, P. L.-J. Chen, S. Chatterjee "Similarity measures for image databases", *Proc. SPIE*, vol. 2420, pp. 58-65, San Jose, CA, Feb. 9-10, 1995.

[15] N. Katayama, S. Satoh, "The SR-tree: An index structure for high-dimensional nearest neighbor queries," *Proc. ACM SIGMOD* pp. 369-380, Tucson, AZ, 1997.

[16] S. Lawrence, C.L. Giles, "Searching the World Wide Web," *Science*, vol. 280, pp. 98, 1998.

[17] S. Lawrence, C.L. Giles, "Accessibility of information on the Web," *Nature*, vol. 400, pp. 107-109, 1999.

[18] J. Li, J. Z. Wang, G. Wiederhold, "IRM: Integrated region matching for image retrieval," *Proc. ACM Multimedia Conference*, pp. 147-156, Los Angeles, ACM, October, 2000.

[19] K.-I. Lin, H. Jagadish, C. Faloutsos, "The TV-tree: An index structure for high-dimensional data," *The VLDB Journal*, vol. 3, no. 4, pp. 517-549, October 1994.

[20] D. Lomet, "The hB-tree: A multiattribute indexing method with good guaranteed performance," *ACM Transactions on Database Systems*, vol. 15, no. 4, pp. 625-658, December 1990.

[21] W. Y. Ma, B. Manjunath, "NaTra: A toolbox for navigating large image databases," *Proc. IEEE Int. Conf. Image Processing*, pp. 568-71, 1997.

[22] S. Mehrotra, Y. Rui, M. Ortega-Binderberger, T.S. Huang, "Supporting content-based queries over images in MARS," *Proc. IEEE International Conference on Multimedia Computing and Systems*, pp. 632-3, Ottawa, Ont., Canada 3-6 June 1997.

[23] S. Mukherjea, K. Hirata, Y. Hara, "AMORE: a World Wide Web image retrieval engine," *World Wide Web*, vol. 2, no. 3, pp. 115-32, Baltzer, 1999.

[24] A. Natsev, R. Rastogi, K. Shim, "WALRUS: A similarity retrieval algorithm for image databases," *Proc. ACM SIGMOD*, Philadelphia, PA, 1999.

[25] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," *Proc. SPIE*, vol. 1908, pp. 173-87, San Jose, February, 1993.

[26] A. Pentland, R. W. Picard, S. Sclaroff, "Photobook: tools for content-based manipulation of image databases," *Proc. SPIE*, vol. 2185, pp. 34-47, San Jose, February 7-8, 1994.

[27] R. W. Picard, T. Kabir, "Finding similar patterns in large image databases," *Proc. IEEE ICASSP*, Minneapolis, vol. V, pp. 161-64, 1993.

[28] J. Robinson, "The k-d-b-tree: A search structure for large multidimensional dynamic indexes," *Proc. ACM SIGMOD* pp. 10-18, 1981.

[29] Y. Rubner, L. J. Guibas, C. Tomasi, "The earth mover's distance, Shimulti-dimensional scaling, and color-based image retrieval," *Proc. ARPA Image Understanding Workshop*, pp. 661-668, New Orleans, LA, May 1997.

[30] G. Salton, M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, NY, 1983.

[31] T. Sellis, N. Roussopoulos, C. Faloustos. "The R+-tree: A dynamic index for multi-dimensional objects," *Proc. Int. Conf. on Very Large Databases*, pp. 507-518, Brighton, England, 1987.

[32] J. Shi, J. Malik, "Normalized cuts and image segmentation," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 731-7, San Juan, Puerto Rico, June, 1997.

[33] J. R. Smith, S.-F. Chang, "An image and video search engine for the World-Wide Web," *Proc. SPIE*, vol. 3022, pp. 84-95, 1997.

[34] J. R. Smith, C. S. Li, "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, 2000.

[35] S. Stevens, M. Christel, H. Wactlar, "Informedia: improving access to digital video," *Interactions*, vol. 1, no. 4, pp. 67-71, 1994.

[36] J. Z. Wang, G. Wiederhold, O. Firschein, X. W. Sha, "Content-based image indexing and searching using Daubechies' wavelets," *International Journal of Digital Libraries*, vol. 1, no. 4, pp. 311-328, 1998.

[37] J. Z. Wang, J. Li, D. , G. Wiederhold, "Semantics-sensitive retrieval for digital picture libraries," *D-LIB Magazine*, vol. 5, no. 11, DOI:10.10 45/november99-wang, November, 1999. http://www.dlib.org

[38] J. Z. Wang, J. Li, G. Wiederhold, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, 2001. to appear.

[39] J. Z. Wang, J. Li, R. M. Gray, G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 85-90, 2001.

[40] J. Z. Wang, *Integrated Region-Based Image Retrieval*, Kluwer Academic Publishers, 190 pp., 2001.

[41] R. Weber, Hans-J. Schek, Stephen Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," *Proc. Int. Conf. on Very Large Databases*, New York, 1998.