SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries *

Jia Li†, James Ze Wang‡, and Gio Wiederhold§

Department of Computer Science, Stanford University, Stanford, CA 94305, USA

Abstract

Content-based image database retrieval has become an active research field in recent years due to the rapid growth of digital image and video storage. We present here SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), an image database retrieval system, using high-level semantics classification and integrated region matching based upon image segmentation. The SIMPLIcity system represents an image by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. Based on segmented regions, the system classifies images into categories which are intended to distinguish semantically meaningful differences. These high-level categories, such as textured-nontextured, indoor-outdoor, objectionable-benign, graph-photograph, enhance retrieval by narrowing down the searching range in a database and permitting semantically adaptive searching methods. A measure for the overall similarity between images is defined by a region-matching scheme that integrates properties of all the regions in the images. Armed with this global similarity measure, the system provides users a simple querying interface. The integrated region matching (IRM) similarity measure is insensitive to inaccurate segmentation. The application of SIMPLIcity to a database of about 60,000 general-purpose images shows robustness to cropping, scaling, shifting, and rotation. Compared with the WBIIS algorithm, SIMPLIcity in general achieves more accurate retrieval at higher speed.

Keywords: Content-based Image Retrieval – Very Large Image Databases – Classification – Image Segmentation – Integrated Matching – K-means

*This work was supported in part by the National Science Foundation. For an on-line demo, see the URL: `http://WWW-DB.Stanford.EDU/~wangz/project/imsearch/SIMPLIcity/`

†Also of Department of Electrical Engineering. Email: `jiali@db.stanford.edu`

‡Also of Department of Medical Informatics. Email: `wangz@cs.stanford.edu`

§Also of Department of Electrical Engineering and Department of Medical Informatics. Email: `gio@cs.stanford.edu`

1 Introduction

With the steady growth of computer power, rapidly declining cost of storage devices, and ever-increasing access to the Internet, digital acquisition of information has become increasingly popular in recent years. Digital information is preferable because of its easy sharing and distribution properties. This trend has motivated research in image databases, which was nearly ignored by traditional computer systems because of the large amount of data required to represent images and the difficulty of automatically analyzing images. Currently, storage is less of an issue since huge storage capacity is available at low cost. However, the efficient indexing and searching of large-scale image databases remains as a challenge for computer systems. The automatic derivation of semantics from the content of an image is the focus of interest for research on image databases.

1.1 Related Work in Content-based Image Retrieval

Many content-based image database retrieval systems have been developed, such as the IBM QBIC System [3, 4] developed at the IBM Almaden Research Center, the Photobook System developed by the MIT Media Lab [16, 15], the WBIIS System [23] developed at Stanford University, and the Blobworld System [1] developed at U.C. Berkeley. The common ground for content-based image retrieval systems is to extract a signature for every image based on its pixel values, and to define a rule for comparing images. This signature serves as an image representation in the 'view' of a retrieval system. The components of the signature are usually called *features*. One advantage of using a signature instead of the original pixel values is the significant simplification of image representation. However, a more important reason for using the signature is the improved correlation with image semantics. Actually, the main task of designing a signature is to bridge the gap between image semantics and the pixel representation, that is, to create better correlation with image semantics. The human vision system (HVS), which may be regarded as a perfect image analyzer, after looking at an image may represent the image as "some brown horses on grass." With the same image, an example signature stored in a database might be "90% of the image is green and 10% is brown." Content-based image database retrieval systems roughly fall into three categories depending on the signature extraction approach used: histogram, color layout, and region-based search. We will briefly review these methods later in this section. There are also systems that combine retrieval results from individual algorithms by a weighted sum matching metric [6, 4], or other merging schemes [18].

After extracting signatures, the next step is to determine a comparison rule, including a querying scheme and the definition of a similarity measure between images. Most image retrieval systems perform a query by having the user specify an image; the system then searches for images similar to the specified one. We refer to this as global search, since similarity is based on the overall properties of images. In contrast to global search, there are also systems that retrieve based on a particular region in an image, such as the NeTra system [11] and the Blobworld system [1]. This querying scheme is referred to as partial search.

1.1.1 Histogram Search

For histogram search [14, 4, 17], an image is characterized by its color histogram. The drawback of histogram representation is over-summarization. Information about object location, shape, and texture is discarded.

1.1.2 Color Layout Search

The "color layout" approach attacks the problems with histogram search to a certain extent. For traditional color layout indexing [14], images are partitioned into blocks and the average color of each block is stored. Thus, the color layout is essentially a low resolution representation of the original image. More advanced systems [23] use significant wavelet coefficients instead of averaging. By adjusting block sizes or the levels of wavelet transforms, the coarseness of a color layout representation can be tuned. The finest color layout using a single pixel block is the original pixel representation. We can hence view a color layout representation as an opposite extreme of a histogram, which naturally retains shape, location, and texture information if at proper resolutions. However, as with pixel representation, although information such as shape is preserved in the color layout representation, the retrieval system cannot "see" it explicitly. Color layout search is sensitive to shifting, cropping, scaling, and rotation because images are characterized by a set of local properties. One approach taken by the WALRUS system [13] to reduce the shifting and scaling sensitivity for color layout search is to exhaustively reproduce many subimages based on an original image. The subimages are formed by sliding windows of various sizes and a color layout signature is computed for every subimage. The similarity between images is then determined by comparing the signatures of subimages. An obvious drawback of the system is the sharply increased computational complexity due to exhaustive generation of subimages. Furthermore, texture and shape information is discarded in the signatures because every subimage is partitioned into four blocks and only average colors of the blocks are used as features. This system is also limited to intensity-level image representations.

1.1.3 Region-based Search

Region-based retrieval systems attempt to overcome the issues with 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 close to the perception of the human visual system. Since the retrieval system has identified what objects are in the image, it is easier for the system to recognize similar objects at different locations and with different orientations and sizes. Region-based retrieval systems include the Netra system [11], the Blobworld system [1], and the query system with color region templates [19].

The NeTra system [11] and the Blobworld system [1] compare images based on individual regions. Although 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 region to be matched and also attributes, e.g., color and texture, of the region to be used for evaluating similarity. Such querying systems provide more control to the users. However, the key pitfall is that the user's semantic understanding of an image is at a higher level than the region representation. When a user submits a query image of a horse on grass, the intent is most likely to retrieve images with horses. But since the concept of horses is not explicitly given in region representations, the user has to convert the concept into shape, color, texture, location, or combinations of them. For objects without distinctive attributes, such as special texture, it is not obvious for the user how to select a query from the large variety of choices. Thus, such a querying scheme may add burdens on users without any reward. On the other hand, because of the great difficulty of achieving accurate segmentation, systems in [11, 1] tend to partition one object into several regions with none of them being representative for the object, especially for images without distinctive objects and scenes. Queries based on such regions often yield images that are indirectly related to the query image.

Not much attention has been paid to developing similarity measures that combine information from all of the regions. One work in this direction is the querying system developed by Smith and Li [19]. 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 regions in an image are not represented in the library. The system in [19] 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.

1.2 Related Work in Image Semantic Classification

Although region-based systems attempt to decompose images into constituent objects, a representation composed of pictorial properties of regions is indirectly related to its semantics. There is no clear mapping from a set of pictorial properties to semantics. An approximately round brown region might be a flower, an apple, a face, or a part of sunset sky. Moreover, pictorial properties such as color, shape, and texture of an object vary dramatically in different images. If a system understood the semantics of images, it would be capable of fast and accurate search. However, due to the great difficulty of understanding images, not much success has been achieved in identifying high-level semantics for the purpose of image retrieval. Therefore,

most systems are confined to matching images with low-level pictorial properties.

Despite the fact that it is currently impossible to reliably recognize objects in general-purpose images, there are methods to distinguish certain semantic types of images. Any information about semantic types is helpful since a system can constrict the search to images with a particular semantic type. The system can also improve retrieval by using various matching schemes tuned to the semantic class of the query image. One example of semantics classification is the identification of natural photographs and artificial graphs generated by computer tools [10, 24]. Other examples include the system to detect objectionable images developed by Wang et al. [24] and the system to classify indoor and outdoor scenes developed by Szummer and Picard [21]. Wang and Fischler [25] have shown that rough but accurate semantic understanding can be very helpful in computer vision tasks such as image stereo matching. Most of these systems use statistical classification methods based on training data.

1.3 Overview of the SIMPLIcity Retrieval System

We now present the major contributions of our proposed SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) system.

1.3.1 Novel Image Retrieval Architecture

The architecture of the SIMPLIcity system is described by Figure 1, the indexing process, and Figure 2, the querying process. During indexing, the system partitions an image into 4×4 pixel blocks and extracts a feature vector for each block. The k-means clustering [8] algorithm is then used to segment the image. The segmentation result is fed into a classifier that decides the semantic type of the image. An image is classified as one of the n pre-defined mutually exclusive and collectively exhaustive semantic classes. As indicated previously, examples of semantic types are indoor-outdoor, objectionable-benign, and graph-photograph images. Features including color, texture, shape, and location information are then extracted for each region in the image. The features selected depend on the semantic type of the image. The signature of an image is the collection of features for all of its regions. Signatures of images with various semantic types are stored in separate databases.

In the querying process, if the query image is not in the database, it is first passed through the same feature extraction process as was used during indexing. For an image in the database, its semantic type is first checked and then its signature is extracted from the corresponding database. Once the signature of the query image is obtained, similarity scores between the query image and images in the database with the same semantic type are computed and sorted to provide the list of images that appear to have the closest semantics.

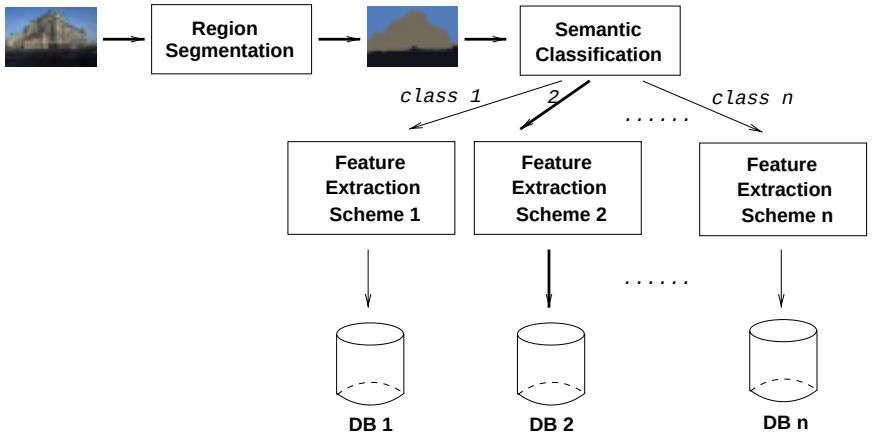


Figure 1: The architecture of feature indexing module. The heavy lines show a sample indexing path of an image.

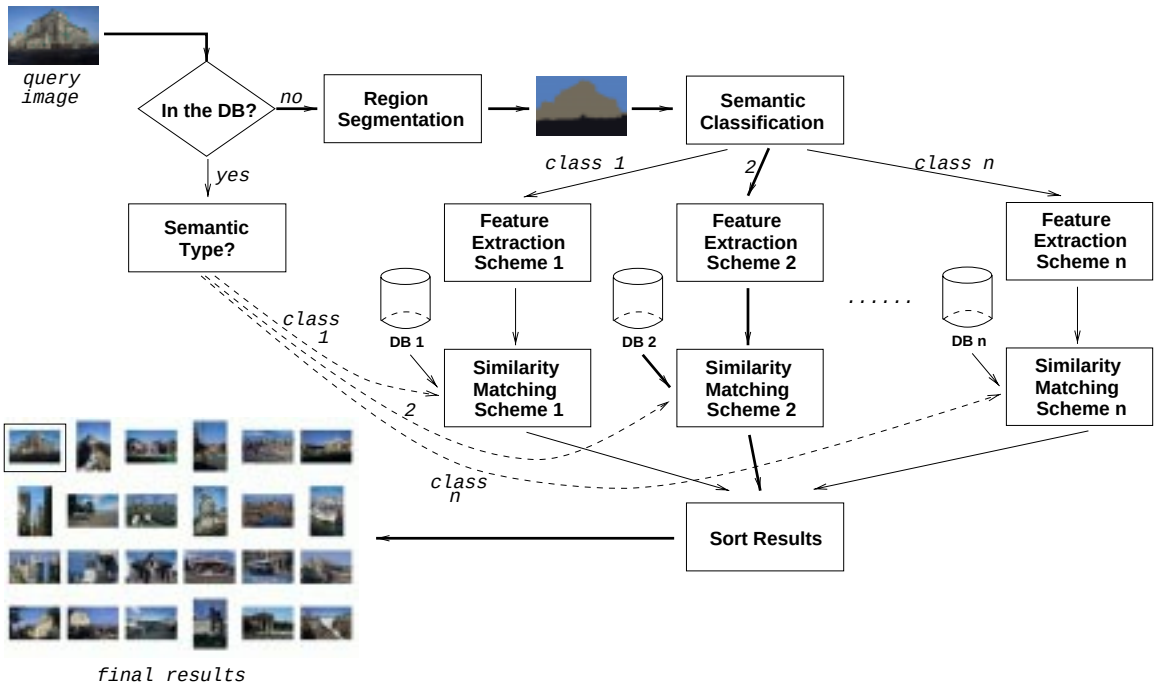


Figure 2: The architecture of query processing module. The heavy lines show a sample querying path of an image.

1.3.2 Textured and non-textured Image Classification

For the current implementation of the SIMPLIcity system, we are particularly interested in classifying images into the classes *textured* and *non-textured*. By textured images, we refer to images that are composed of repeated patterns and appear like a unique texture surface, as shown in Figure 3. As textured images do not contain clustered objects, the perception of such images focuses on color and texture, but not shape, which is critical for understanding non-textured images. Thus an efficient retrieval system should use different features to depict the two types of images. To our knowledge, the problem of distinguishing textured images and non-textured images has not been explored in the image retrieval literature. In this paper, we describe an algorithm to detect texture images based on segmentation results.

Figure 3: Sample textured images.

1.3.3 Integrated Region Matching (IRM) Similarity Measure

Besides using high-level semantics classification, another strategy of SIMPLIcity to shorten the distance between the region representation of an image and its semantics is to define a similarity measure between images based on the properties of all the segmented regions so that information about an image can be fully used. In many cases, knowing that one object usually appears with another object helps to clarify the semantics of a particular region. For example, flowers often appear with green leaves, and boats usually appear with water.

By defining an overall similarity measure, the SIMPLIcity system provides users with a *simple* querying interface. To complete a query, a user only needs to specify the query image. Compared with retrieval based on individual regions, the overall similarity approach also reduces the influence of inaccurate segmentation.

Mathematically, defining the similarity measure is equivalent to defining a distance between sets of points in a high dimensional space, i.e., the feature space. Every point in the space corresponds to the feature vector, or the descriptor, of a region. Although distance between two points in feature space can be easily defined by the Euclidean distance, it is not obvious how to define a distance between sets of points that corresponds to a person's concept of semantic "closeness" of two images. We expect a good distance to take all the points in a set into consideration and to be tolerant to inaccurate image segmentation. To define the similarity measure, we first attempt to match regions in two images. Being aware that segmentation cannot be perfect, we "soften" the matching by allowing one region to be matched to several regions with significance scores. The principle of matching is that the closest region pair is matched first. We call this matching scheme

integrated region matching (IRM) to stress the incorporation of regions in the retrieval process. After regions are matched, the similarity measure is computed as a weighted sum of the similarity between region pairs, with weights determined by the matching scheme.

1.4 Outline of the Paper

The outline of the remainder of the paper is as follows. In Section 2, the image segmentation algorithm used by the SIMPLIcity system is described. We then present the classification algorithm for detecting textured images in Section 3. The similarity measure based on segmentation is defined in Section 4. In Section 5, we describe experiments and provide results. We conclude and suggest future research in Section 6.

2 Image Segmentation

This section describes our image segmentation procedure based on color and frequency features using the k-means algorithm [8]. For general-purpose images such as the images in a photo library or the images on the World-Wide Web (WWW), automatic image segmentation is almost as difficult as automatic image semantic understanding. Currently there is no existing non-stereo image segmentation algorithm that can perform at the level of the HVS. The segmentation accuracy of our system is not crucial because we use a more robust integrated region-matching (IRM) scheme which is insensitive to inaccurate segmentation.

To segment an image, SIMPLIcity partitions the image into blocks with 4×4 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. The k-means algorithm does not specify how many clusters to choose. We adaptively choose the number of clusters k by gradually increasing k and stop when a criterion is met. We start with $k = 2$ and stop increasing k if one of the following conditions is satisfied.

1. The distortion $D(k)$ is below a threshold.
2. The first derivative of distortion with respect to k, $D(k)-D(k-1)$, is below a threshold with comparison to the average derivative at $k = 2, 3$.
3. The number k exceeds an upper bound.

Six features are used for segmentation. Three of them are the average color components in a 4×4 block. The other three represent energy in high frequency bands of wavelet transforms [2, 12], 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, U and V encode color information (chrominance). To obtain the other three features, the Haar wavelet transform is applied to the L component of the image. After a one-level wavelet transform, a 4×4 block is decomposed into four frequency bands as shown in Figure 4.

Each band contains 2×2 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 then computed as

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

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 [22]. 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.

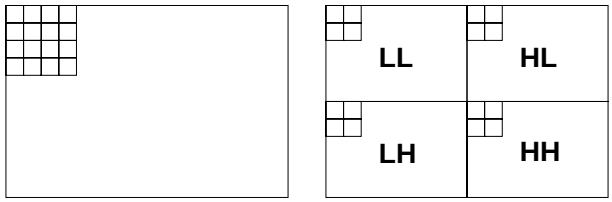


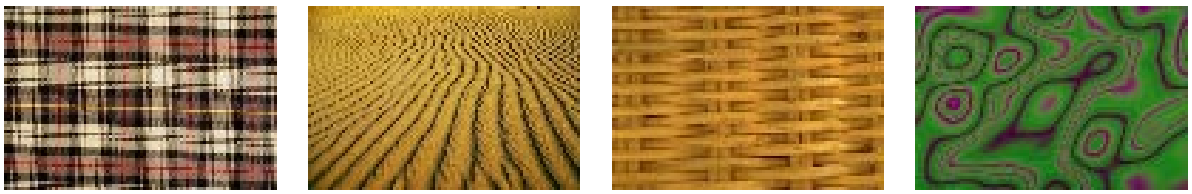
Figure 4: Decomposition of images into frequency bands by wavelet transforms.

Examples of segmentation results for both texture and non-textured images are shown in Figure 5. Segmented regions are shown in their representative colors. It takes about one second on average to segment a 384×256 image on a Pentium Pro 430MHz PC using the Linux operating system. We do not apply post-processing techniques to smooth region boundaries or to delete small isolated regions because these errors are often less significant. Since our retrieval system is designed to tolerate inaccurate segmentation, cleaning the segmentation results by post-processing (at the cost of speed) is unnecessary.

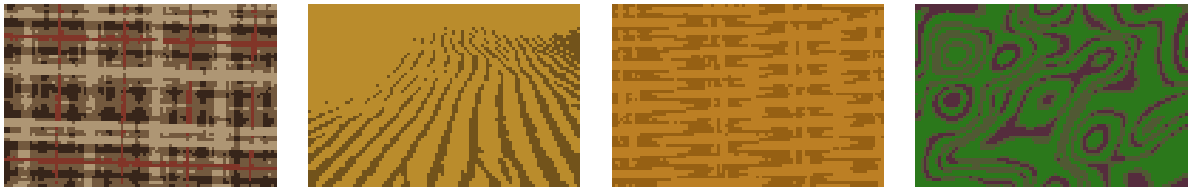
3 Classification of Textured and non-textured Images

In this section we describe the algorithm to classify images into the semantic classes *textured* or *non-textured*. For textured images, color and texture are much more important perceptually than shape, since there are no clustered objects. As shown by the segmentation results in Figure 5, regions in textured images tend to scatter in the entire image, whereas non-textured images are usually partitioned into clumped regions. A mathematical description of how evenly a region scatters in an image is the goodness of match between the distribution of the region and a uniform distribution. The goodness of fit is measured by the χ^2 statistics [20].

We partition an image evenly into 16 zones, $\{Z_1, Z_2, ..., Z_{16}\}$. Suppose the image is segmented into



(a)



#regions=4
$\bar{\chi}^2 = 0.125$

#regions=2
$\bar{\chi}^2 = 0.207$

#regions=2
$\bar{\chi}^2 = 0.009$

#regions=3
$\bar{\chi}^2 = 0.066$

(b)

(c)

#regions=2
$\bar{\chi}^2 = 0.694$

#regions=4
$\bar{\chi}^2 = 1.613$

#regions=4
$\bar{\chi}^2 = 1.447$

#regions=12
$\bar{\chi}^2 = 1.249$

(d)

Figure 5: Segmentation results by the k-means clustering algorithm: (a) Original texture images, (b) Regions of the texture images, (c) Original non-textured images, (d) Regions of the non-textured images.

regions $\{r_i : i = 1, ..., m\}$. For each region r_i, its percentage in zone Z_j is $p_{i,j}$, $\sum_{j=1}^{16} p_{i,j} = 1$, $i = 1, ..., m$. The uniform distribution over the zones should have probability mass function $q_j = 1/16$, $j = 1, ..., 16$. The χ^2 statistics for region i, χ_i^2, is computed by

$$\chi_i^2 = \sum_{j=1}^{16} \frac{(p_{i,j} - q_j)^2}{q_j} = \sum_{j=1}^{16} 16(p_{i,j} - \frac{1}{16})^2$$

The classification of textured or non-textured image is performed by thresholding the average χ^2 statistics for all the regions in the image, $\bar{\chi}^2 = \frac{1}{m}\sum_{i=1}^{m} \chi_i^2$. If $\bar{\chi}^2 < 0.32$, the image is labeled as textured; otherwise, non-textured. We randomly chose 100 textured images and 100 non-textured images and computed $\bar{\chi}^2$ for them. The histograms of $\bar{\chi}^2$ for the two types of images are shown in Figure 6. It can be seen that the two histograms separate significantly around the decision threshold 0.32.

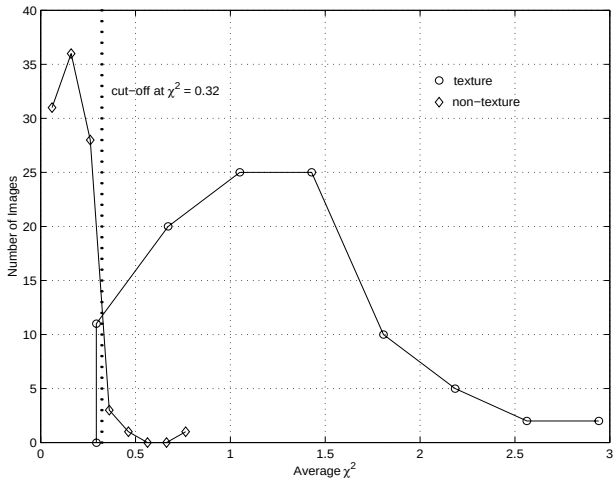


Figure 6: The histograms of average χ^2's over 100 textured images and 100 non-textured images.

4 The Similarity Measure

4.1 Integrated Region Matching

In this section, we define the similarity measure between two sets of regions. Assume that Image 1 and 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. Details about features included in r_i and the definition of $d(r_i, r'_j)$ will be discussed later. To compute the similarity measure between region sets R_1 and R_2, $d(R_1, R_2)$, we first match all regions in the two images. The matching scheme attempts to mimic the image comparison process of the HVS. For example, when the HVS judges the similarity of two animal photographs, it will first compare the animals in the images,

then compare the background areas in the images. The overall similarity of the two images depends on the closeness in the two aspects. The correspondence between objects in the images is crucial for the HVS's judgment of similarity. It would be meaningless to compare the animal in one image with the background in another. Our matching scheme aims at building correspondence between regions that is consistent with human 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 $s_{i,j}$, $s_{i,j} \geq 0$. The significance credit indicates the importance of the matching for determining similarity between images. The matrix

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \dots & s_{1,n} \\ s_{2,1} & s_{2,2} & \dots & s_{2,n} \\ \dots & \dots & \dots & \dots \\ s_{m,1} & s_{m,2} & \dots & s_{m,n} \end{pmatrix} ,$$

is referred to as the significance matrix.

A graphical explanation of the integrated matching scheme is provided in Figure 7. The figure shows that matching between images can be represented by an edge weighted graph in which every vertex in the graph corresponds to a region. If two vertices are connected, the two regions are matched with a significance credit being the weight on the edge. To distinguish from matching two sets of regions, we refer to the matching of two regions as they are *linked*. The length of an edge can be regarded as the distance between the two regions represented. If two vertices are not connected, the corresponding regions are either from the same image or the significance credit of matching them is zero. Every matching between images is characterized by links between regions and their significance credits. The matching used to compute the distance between two images is referred to as the *admissible matching*. The admissible matching is specified by conditions on the significance matrix. If a graph represents an admissible matching, the distance between the two region sets is the summation of all the weighted edge lengths, i.e.,

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

We call this distance the integrated region matching (IRM) distance.

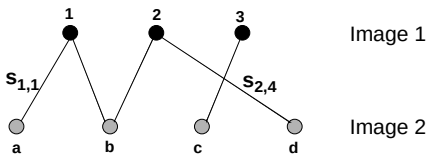


Figure 7: Integrated region matching (IRM).

The problem of defining distance between region sets is then converted to 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. 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

$$\begin{aligned} \sum_{j=1}^{n} s_{i,j} &= p_i,\ i = 1, ..., m \\ \sum_{i=1}^{m} s_{i,j} &= p'_j,\ j = 1, ..., n\ . \end{aligned}$$

For normalization, we have $\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} p'_j = 1$. The fulfillment of significance ensures that all the regions play a role for measuring similarity. We also require an admissible matching to link the most similar regions at the highest priority. For example, if two images are the same, the admissible matching should link a region in Image 1 only to the same region in Image 2. With this matching, the distance between the two images equals zero, which coincides with our intuition. Following the "most similar highest priority (MSHP)" principle, the IRM algorithm attempts to fulfill the significance credits of regions by assigning as much significance as possible to the region link with minimum distance. Initially, assume that $d_{i',j'}$ is the minimum distance, we set $s_{i',j'} = \min(p_{i'}, p'_{j'})$. Without loss of generality, assume $p_{i'} \leq p'_{j'}$. Then $s_{i',j} = 0$, for $j \neq j'$ since the link between region i' and j' has filled the significance of region i'. The significance credit left for region j' is reduced to $p'_{j'} - p_{i'}$. The updated matching problem is then solving $s_{i,j}$, $i \neq i'$, by the MSHP rule under constraints:

$$\begin{aligned} \sum_{j=1}^{n} s_{i,j} &= p_i \quad 1 \leq i \leq m,\ i \neq i' \\ \sum_{i:1\leq i\leq m, i\neq i'} s_{i,j} &= p'_j \quad 1 \leq j \leq n,\ j \neq j' \\ \sum_{i:1\leq i\leq m, i\neq i'} s_{i,j'} &= p'_{j'} - p_{i'} \\ s_{i,j} &\geq 0 \quad 1 \leq i \leq m,\ i \neq i';\ 1 \leq j \leq n\ . \end{aligned}$$

We apply the previous procedure to the updated problem. The iteration stops when all the significance credits p_i and p'_j have been met. The algorithm is summarized as follows.

1. Set $\mathcal{L} = \{\}$, denote $\mathcal{M} = \{(i, j) : i = 1, ..., m; j = 1, ..., n\}$.
2. Choose the minimum $d_{i,j}$ for $(i, j) \in \mathcal{M} - \mathcal{L}$. Label the corresponding (i, j) as (i', j').
3. $\min(p_{i'}, p'_{j'}) \rightarrow s_{i',j'}$.
4. If $p_{i'} < p'_{j'}$, set $s_{i',j} = 0$, $j \neq j'$; otherwise, set $s_{i,j'} = 0$, $i \neq i'$.

5. $p_{i'} - \min(p_{i'}, p'_{j'}) \rightarrow p_{i'}$.

6. $p'_{j'} - \min(p_{i'}, p'_{j'}) \rightarrow p'_{j'}$.

7. $\mathcal{L} + \{(i', j')\} \rightarrow \mathcal{L}$.

8. If $\sum_{i=1}^{m} p_i > 0$ and $\sum_{j=1}^{n} p'_j > 0$, go to Step 2; otherwise, stop.

We now come to the issue of choosing p_i. The value of p_i is chosen to reflect the significance of region i in the image. If we assume that every region is equally important, then $p_i = 1/m$, where m is the number of regions. In the case that Image 1 and 2 have the same number of regions, a region in Image 1 is matched exclusively to one region in Image 2. Another choice of p_i is the percentage of the image covered by region i based on the view that important objects in an image tend to occupy larger areas. We refer to this assignment of p_i as the *area percentage scheme*. This scheme is less sensitive to inaccurate segmentation than the uniform scheme. If one object is partitioned into several regions, the uniform scheme raises its significance improperly, whereas the area percentage scheme retains its significance. On the other hand, if objects are merged into one region, the area percentage scheme assigns relatively high significance to the region. The SIMPLIcity system uses the area percentage scheme.

The scheme of assigning significance credits can also take region location into consideration. For example, higher significance may be assigned to regions in the center of an image than to those around boundaries. Another way to count location in the similarity measure is to generalize the definition of the IRM distance to

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

The parameter $w_{i,j}$ is chosen to adjust the effect of region i and j on the similarity measure. In the SIMPLIcity system, regions around boundaries are slightly down-weighted by using this generalized IRM distance.

4.2 Distance Between Regions

We now discuss the definition of distance between a region pair, $d(r, r')$. The SIMPLIcity system characterizes a region by color, texture, and shape. The feature extraction process is shown in Figure 8. We have described in Section 2 the features used by the k-means algorithm for segmentation. The mean values of these features in one cluster are used to represent color and texture in the corresponding region. To describe shape, normalized inertia [5] of order 1 to 3 are used. For a region H in k dimensional Euclidean space $\Re^k$, its normalized inertia of order γ is

$$l(H, \gamma) = \frac{\int_H \|x - \hat{x}\|^\gamma dx}{[V(H)]^{1+\gamma/k}}$$

where $\hat{x}$ is the centroid of H and $V(H)$ is the volume of H. Since an image is specified by pixels on a grid, the discrete form of the normalized inertia is used, that is,

$$l(H,\gamma) = \frac{\sum_{x:x\in H} \|x-\hat{x}\|^{\gamma}}{[V(H)]^{1+\gamma/k}}$$

where $V(H)$ is the number of pixels in region H. The normalized inertia is invariant with scaling and rotation. The minimum normalized inertia is achieved by spheres. Denote the γth order normalized inertia of spheres as L_γ. We define shape features as $l(H,\gamma)$ normalized by L_γ:

$$f_7 = l(H,1)/L_1 \ , \quad f_8 = l(H,2)/L_2 \ , \quad f_9 = l(H,3)/L_3 \ .$$

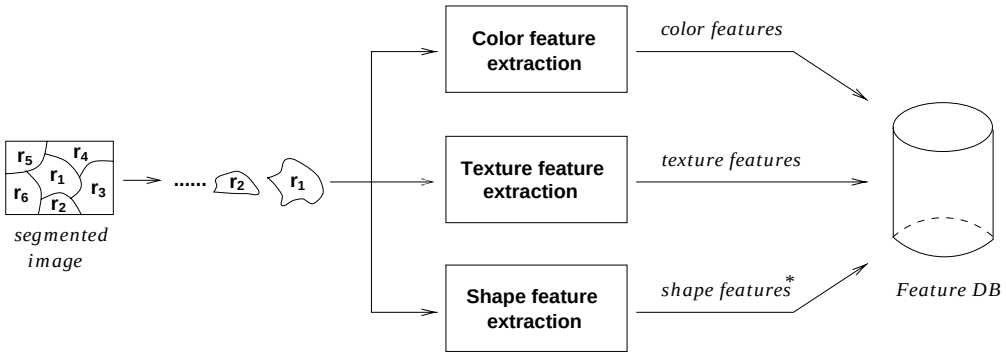


Figure 8: Feature extraction in the SIMPLIcity system. * The computation of shape features is skipped for textured images.

The computation of shape features is skipped for textured images because in this case region shape is not important perceptually. The region distance $d(r,r')$ is defined as

$$d(r,r') = \sum_{i=1}^{6} w_i (f_i - f_i')^2 \ .$$

For non-textured images, $d(r,r')$ is defined as

$$d(r,r') = g(d_s(r,r')) \cdot d_t(r,r') \ ,$$

where $d_s(r,r')$ is the shape distance computed by

$$d_s(r,r') = \sum_{i=7}^{9} w_i (f_i - f_i')^2 \ ,$$

and $d_t(r,r')$ is the color and texture distance defined the same as the distance between textured image

regions, i.e.,

$$d_t(r, r') = \sum_{i=1}^{6} w_i (f_i - f_i')^2 \ .$$

The function $g(d_s(r, r'))$ is a converting function to ensure a proper influence of the shape distance on the total distance. In our system, it is defined as

$$g(d) = \begin{cases} 1 & d \geq 0.5 \\ 0.85 & 0.2 < d \leq 0.5 \\ 0.5 & d < 0.2 \ . \end{cases}$$

It is observed that when $d_s(r, r') \geq 0.5$, the two regions bear little resemblance. It is then not meaningful to distinguish the extent of similarity by $d_s(r, r')$ because perceptually the two regions simply appear different. We thus set $g(d) = 1$ for d greater than a threshold. When $d_s(r, r')$ is very small, we intend to keep the influence of color and texture. Therefore $g(d)$ is bounded away from zero. We set $g(d)$ to be a piece-wise constant function instead of a smooth function for simplicity. Because rather simple shape features are used in our system, we emphasize color and texture more than shape. As can be seen from the definition of $d(r, r')$, the shape distance serves as a "bonus." If two regions match very well in shape, their color and texture distance is attenuated by a smaller weight to provide the final distance.

5 Experiments

The SIMPLIcity system has been implemented with a general-purpose image database including about $60,000$ pictures, which are stored in JPEG format with size 384×256 or 256×384. These images were segmented and classified into textured and non-textured types. For each image, the features, locations, and areas of all its regions are stored. Textured images and non-textured images are stored in separate databases. According to SIMPLIcity, there are 3772 texture images in the database, about 6% of the total collection. Because the WBIIS system is the only other system we have access to, we compare the accuracy of the SIMPLIcity system to that of the WBIIS system using the same database. Moreover, it is difficult to design a fair comparison with existing region-based searching algorithms such as the Blobworld system which depends on manually defined complicated queries. An on-line demo is provided at URL: `http://WWW-DB.Stanford.EDU/~wangz/project/imsearch/SIMPLIcity/` .

5.1 Accuracy Comparison with WBIIS

We compare the SIMPLIcity system with the WBIIS (Wavelet-Based Image Indexing and Searching) system [23] with the same image database. As WBIIS form image signatures using wavelet coefficients in the

lower frequency bands, it performs well with relatively smooth images, such as most landscape images. For images with details crucial to semantics, such as pictures with people, the performance of WBIIS degrades. In general, SIMPLIcity performs as well as WBIIS for smooth landscape images. One example is shown in Figure 9. The query image is the image at the upper-left corner. The underlined numbers below the pictures are the ID numbers of the images in the database. To view the images better or to see more matched images, users can visit the demo web site and use the query image ID to repeat the retrieval.

SIMPLIcity also performs well for images composed of fine details. Retrieval results with a photo of a hamburger as the query are shown in Figure 10. The SIMPLIcity system retrieves 10 images with food out of the first 11 matched images. The WBIIS system, however, does not retrieve any image with food in the first 11 matches. The top match made by SIMPLIcity is also a photo of hamburger, which is perceptually very close to the query image. WBIIS misses this image because the query image contains important fine details, which are smoothed out by the multi-level wavelet transform in the system. The smoothing also causes a textured image (the third match) to be matched. Such errors are observed with many other images abundant of details although they are perceptually very different from textured images. The SIMPLIcity system, however, prevents textured images to be matched to non-textured images by classifying them before searching.

Another three query examples are compared in Figure 11, 12, and 13. The query images in Figure 11 and 12 are difficult to match because objects in the images are not distinctive from the background. Moreover, the color contrast for both images is small. It can be seen that the SIMPLIcity system achieves much better retrieval. For the query in Figure 11, only the third matched image is not a picture of a person. A few images, the 1st, 4th, 7th, and 8th matches, depict a similar topic as well, probably about life in Africa. The query in Figure 13 also shows the advantages of SIMPLIcity. The system finds photos of similar flowers with different sizes and orientations. Only the 9th match does not have flowers in it.

For textured images, SIMPLIcity and WBIIS often perform equally well. However, SIMPLIcity captures high frequency texture information better. An example of textured image search is shown in Figure 14. The granular surface in the query image is matched more accurately by the SIMPLIcity system.

5.2 Robustness to Scaling, Shifting, and Rotation

To show the robustness of the SIMPLIcity system to cropping and scaling, querying examples are provided in Figure 15. As we can see, one query image is a cropped and scaled version of the other. Using either of them as query, SIMPLIcity retrieves the other one as the top match. Retrieval results based on both of the queries are good. However, the retrieval performed by WBIIS using one of the images misses the other one.

To test the robustness to shifting, we shifted two example images and used the shifted images as query images. Results are shown in Figure 16. The original images are both retrieved as the top match. In both cases, SIMPLIcity also finds many other semantically related images. This is expected since the shifted

images are segmented into regions nearly the same as those of the original images. In general, if shifting does not affect region segmentation significantly, the system will be able to retrieve the original images with a high rank.

Another example is provided in Figure 17 to show the effect of rotation. SIMPLIcity retrieves the original image as the top match. All the other images matched are also food pictures. For an image without strong orientational texture, such as the query image in Figure 17, its rotation will be segmented into regions with similar features. Therefore, SIMPLIcity will be able to match images similar to those retrieved by the original image.

5.3 Speed

The algorithm has been implemented on a Pentium Pro 430MHz PC using the Linux operating system. To compute the feature vectors for the $60,000$ color images of size 384×256 in our general-purpose image database requires approximately 17 hours. On average, one second is needed to segment an image and to compute the features of all regions.

The matching speed is very fast. When the query image is in the database, it takes about 1.5 seconds of CPU time on average to sort all the images in the database using our similarity measure. If the query is not in the database, one extra second of CPU time is spent to process the query.

SIMPLIcity

WBIIS

Figure 9: Comparison of SIMPLIcity and WBIIS. The query image is a landscape image on the upper-left corner of each block of images.

SIMPLIcity

WBIIS

Figure 10: Comparison of SIMPLIcity and WBIIS. The query image is a photo of food.

SIMPLIcity

WBIIS

Figure 11: Comparison of SIMPLIcity and WBIIS. The query image is a portrait image.

SIMPLIcity

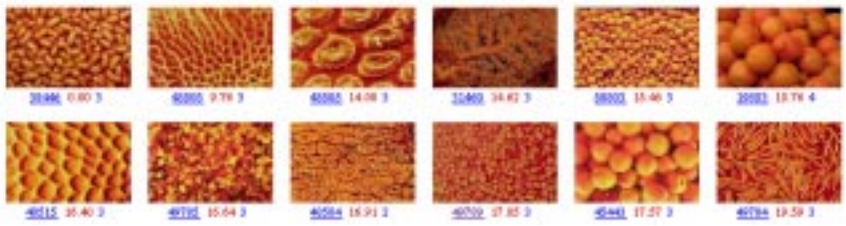
WBIIS

Figure 12: Comparison of SIMPLIcity and WBIIS. The query image is a portrait image.

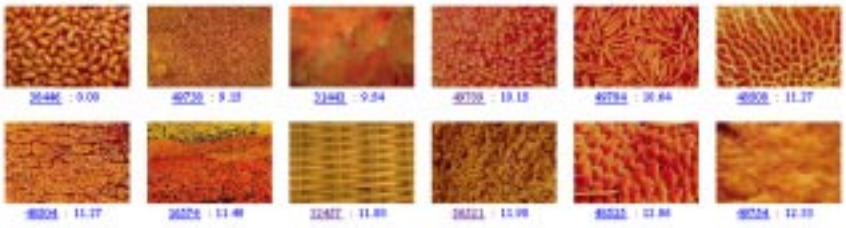
SIMPLIcity

WBIIS

Figure 13: Comparison of SIMPLIcity and WBIIS. The query image is a photo of flowers.



SIMPLIcity



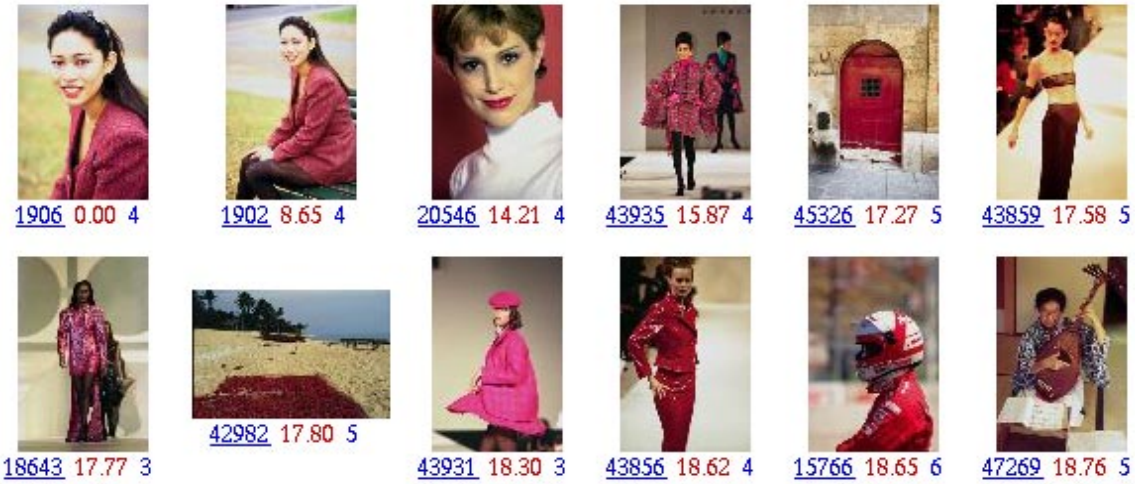
WBIIS

Figure 14: Comparison of SIMPLIcity and WBIIS. The query image is a textured image.

6 Conclusions and Future Work

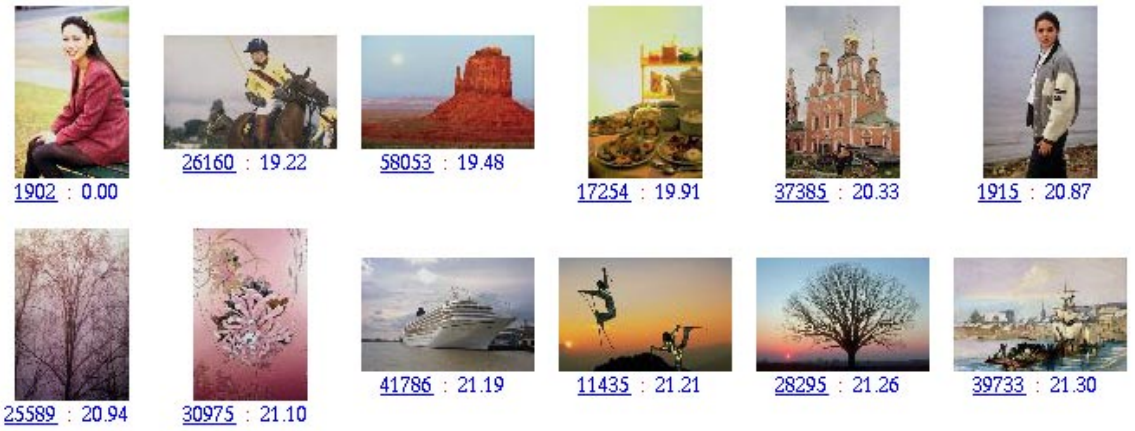
An important contribution of this paper is the idea that images can be classified into global semantic classes, such as textured or nontextured, indoor or outdoor, objectionable or benign, graph or photograph, and that much can be gained if the feature extraction scheme is tailored to best suit each class. We have implemented this idea in SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), an image database retrieval system that uses high-level semantics classification and integrated region matching (IRM) based upon image segmentation. A method for classifying textured or non-textured images using statistical testing has been developed. A measure for the overall similarity between images, defined by a region-matching scheme that integrates properties of all the regions in the images, makes it possible to provide a simple querying interface. The application of SIMPLIcity to a database of about 60,000 general-purpose images shows more accurate and faster retrieval compared with the WBIIS algorithm. Additionally, SIMPLIcity is robust to cropping, scaling, shifting, and rotation.

We are working on integrating more semantic classification algorithms to SIMPLIcity. In addition, it is possible to improve the accuracy by developing a more robust region-matching scheme. The speed can be improved significantly by adopting a feature clustering scheme or using a parallel query processing scheme. We are also working on a simple but capable interface for partial query processing. Experiments with our system on a WWW image database or a video database could be another interesting study.



SIMPLIcity

SIMPLIcity



WBIIS

Figure 15: The robustness of the SIMPLIcity system to image cropping and scaling.

(a)

(b)

Figure 16: The retrieval results made by the SIMPLIcity system with shifted query images.

Figure 17: The retrieval results made by the SIMPLIcity system with a rotated query image.

7 Acknowledgments

We would like to thank Oscar Firschein of Stanford University, Dragutin Petkovic and Wayne Niblack of the IBM Almaden Research Center, Kyoji Hirata and Yoshinori Hara of NEC C&C Research Laboratories, and Martin A. Fischler and Quang-Tuan Luong of the SRI International for valuable discussions on content-based image retrieval, image understanding and photography. The work is funded in part by the Digital Libraries Initiative of the National Science Foundation.

References

[1] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik, "Blobworld: A system for region-based image indexing and retrieval," *Third Int. Conf. on Visual Information Systems*, June 1999.

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

[3] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and 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.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," *Computer*, vol. 28, no. 9, pp. 23-32, Sept. 1995.

[5] A. Gersho, "Asymptotically Optimum Block Quantization," *IEEE Trans. Inform. Theory*, vol. IT-25, no. 4, pp. 373-380, July 1979.

[6] A. Gupta and R. Jain, "Visual information retrieval," *Comm. Assoc. Comp. Mach.*, vol. 40, no. 5, pp. 70-79, May 1997.

[7] D. Harman, "Relevance feedback and other query modification techniques," *Information retrieval: Data structures & algorithms*, Prentice Hall, 1992.

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

[9] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, McGraw-Hill Publishing Company, 1990.

[10] J. Li and R. M. Gray, "Context based multiscale classification of images," *Int. Conf. Image Processing*, Chicago, Oct. 1998.

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

[12] Y. Meyer, *Wavelets Algorithms and Applications*, SIAM, Philadelphia, 1993.

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

[14] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, "The QBIC project: querying images by content using color, texture, and shape," *Proc. SPIE - Int. Soc. Opt. Eng., in Storage and Retrieval for Image and Video Database*, vol. 1908, pp. 173-87, 1993.

[15] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *SPIE Storage and Retrieval Image and Video Databases II*, San Jose, 1995.

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

[17] Y. Rubner, L. J. Guibas, and C. Tomasi, "The Earth Mover's Distance, Multi-Dimensional Scaling, and Color-Based Image Retrieval," *Proceedings of the ARPA Image Understanding Workshop*, pp. 661-668, New Orleans, LA, May 1997.

[18] G. Sheikholeslami, W. Chang, and A. Zhang, "Semantic clustering and querying on heterogeneous features for visual data," *ACM Multimedia*, pp. 3-12, Bristol, UK, 1998.

[19] J. R. Smith and C. S. Li, "Image classification and querying using composite region templates," *Journal of Computer Vision and Image Understanding*, 1999, to appear.

[20] G. W. Snedecor and W. G. Cochran, *Statistical Methods,* Iowa State University Press, Ames, Iowa, 1989.

[21] M. Szummer and R. W. Picard, "Indoor-outdoor image classification," *Int. Workshop on Content-based Access of Image and Video Databases*, pp. 42-51, Jan. 1998.

[22] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing,* vol. 4, no. 11, pp. 1549-1560, Nov. 1995.

[23] J. Z. Wang, G. Wiederhold, O. Firschein, and 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.

[24] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, "System for screening objectionable images," *Computer Communications Journal*, vol. 21, no. 15, pp. 1355-60, Elsevier Science, 1998.

[25] J. Z. Wang, M. A. Fischler, "Visual similarity, judgmental certainty and stereo correspondence," *Proceedings of DARPA Image Understanding Workshop*, Morgan Kauffman, Monterey, 1998.