SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries

James Z. Wang, *Member*, *IEEE*, Jia Li, *Member*, *IEEE*, and Gio Wiederhold, *Fellow*, *IEEE*

Abstract—The need for efficient content-based image retrieval has increased tremendously in many application areas such as biomedicine, military, commerce, education, and Web image classification and searching. We present here SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries), an image retrieval system, which uses semantics classification methods, a wavelet-based approach for feature extraction, and integrated region matching based upon image segmentation. As in other region-based retrieval systems, an image is represented by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. The system classifies images into semantic categories, such as textured-nontextured, graph-photograph. Potentially, the categorization enhances retrieval by permitting semantically-adaptive searching methods and narrowing down the searching range in a database. A measure for the overall similarity between images is developed using a region-matching scheme that integrates properties of all the regions in the images. Compared with retrieval based on individual regions, the overall similarity approach 1) reduces the adverse effect of inaccurate segmentation, 2) helps to clarify the semantics of a particular region, and 3) enables a *simple* querying interface for region-based image retrieval systems. The application of SIMPLIcity to several databases, including a database of about 200,000 general-purpose images, has demonstrated that our system performs significantly better and faster than existing ones. The system is fairly robust to image alterations.

Index Terms—Content-based image retrieval, image classification, image segmentation, integrated region matching, clustering, robustness.

1 Introduction

With the steady growth of computer power, rapidly declining cost of storage, and ever-increasing access to the Internet, digital acquisition of information has become increasingly popular in recent years. Effective indexing and searching of large-scale image databases remain as challenges for computer systems.

The automatic derivation of semantically-meaningful information from the content of an image is the focus of interest for most research on image databases. The image "semantics," i.e., the meanings of an image, has several levels. From the lowest to the highest, these levels can be roughly categorized as

1. semantic types (e.g., landscape photograph, clip art),
2. object composition (e.g., a bike and a car parked on a beach, a sunset scene),
3. abstract semantics (e.g., people fighting, happy person, objectionable photograph), and
4. detailed semantics (e.g., a detailed description of a given picture).

- *J.Z. Wang is with the School of Information Sciences and Technology and the Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16801. E-mail: wangz@cs.stanford.edu.*
- *J. Li is with the Department of Statistics, The Pennsylvania State University, University Park, PA 16801. E-mail: jiali@stat.psu.edu.*
- *G. Wiederhold is with the Department of Computer Science, Stanford University, Stanford, CA 94305. E-mail: gio@cs.stanford.edu.*

Manuscript received 20 Oct. 1999; revised 8 Aug. 2000; accepted 21 Feb. 2001.
Recommended for acceptance by R. Picard.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110789.

Content-based image retrieval (CBIR) is the set of techniques for retrieving semantically-relevant images from an image database based on automatically-derived image features.

1.1 Related Work in CBIR

CBIR for general-purpose image databases is a highly challenging problem because of the large size of the database, the difficulty of understanding images, both by people and computers, the difficulty of formulating a query, and the issue of evaluating results properly. A number of general-purpose image search engines have been developed. We cannot survey all related work in the allocated space. Instead, we try to emphasize some of the work that is most related to our work. 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 [4] is one of the earliest systems. Recently, additional systems have been developed at IBM T.J. Watson [22], VIRAGE [7], NEC AMORA [13], Bell Laboratory [14], and Interpix. In the academic domain, MIT Photobook [15], [17], [12] is one of the earliest. Berkeley Blobworld [2], Columbia VisualSEEK and WebSEEK [21], CMU Informedia [23], UCSB NeTra [11], UCSD [9], University of Maryland [16], Stanford EMD [18], and Stanford WBIIS [28] are some of the recent systems.

The common ground for CBIR systems is to extract a signature for every image based on its pixel values and to define a rule for comparing images. The *signature* serves as an image representation in the "view" of a CBIR system. The components of the signature are called *features*. One advantage of a signature over the original pixel values is the significant compression of image representation. However,

a more important reason for using the signature is to gain on improved correlation between image representation and 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 a better correlation with image semantics.

Existing general-purpose CBIR systems roughly fall into three categories depending on the approach to extract signatures: histogram, color layout, and region-based search. We will briefly review the three methods in this section. There are also systems that combine retrieval results from individual algorithms by a weighted sum matching metric [7], [4], or other merging schemes [19].

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. For most image retrieval systems, a query is specified by an image to be matched. We refer to this as global search since similarity is based on the overall properties of images. By contrast, there are also "partial search" querying systems that retrieve based on a particular region in an image [11], [2].

1.1.1 Histogram Search

Histogram search algorithms [4], [18] characterize an image by its color distribution or histogram. Many distances have been used to define the similarity of two color histogram representations. Euclidean distance and its variations are the most commonly used [4]. Rubner et al. of Stanford University proposed the earth mover's distance (EMD) [18] using linear programming for matching histograms.

The drawback of a global histogram representation is that information about object location, shape, and texture [10] is discarded. Color histogram search is sensitive to intensity variation, color distortions, and cropping.

1.1.2 Color Layout Search

The "color layout" approach attempts to overcome the drawback of histogram search. In simple color layout indexing [4], 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. A relatively recent system, WBIIS [28], uses significant Daubechies' 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. Hence, we can view a color layout representation as an opposite extreme of a histogram. At proper resolutions, the color layout representation naturally retains shape, location, and texture information. However, as with pixel representation, although information such as shape is preserved in the color layout representation, the retrieval system cannot perceive it directly. Color layout search is sensitive to shifting, cropping, scaling, and rotation because images are described by a set of local properties [28].

The approach taken by the recent WALRUS system [14] 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 and increase of size of the search space 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 deficiencies of color layout search by representing images at the object-level. A region-based retrieval system applies image segmentation [20], [27] 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). However, image segmentation is nearly as difficult as image understanding because the images are 2D projections of 3D objects and computers are not trained in the 3D world the way human beings are.

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 [2], and the query system with color region templates [22].

The NeTra and the Blobworld systems 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 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 to the user. However, the user's semantic understanding of an image is at a higher level than the region representation. For objects without discerning 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 significant reward. On the other hand, because of the great difficulty of achieving accurate segmentation, systems in [11], [2] often partition one object into several regions with none of them being representative for the object, especially for images without distinctive objects and scenes.

Not much attention has been paid to 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 [22]. Their system decomposes an image into regions with characterizations predefined 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. The measure is also lacking

robustness to scaling and rotation. 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 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 also 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 Semantic Classification

The underlying assumption of CBIR is that semantically-relevant images have similar visual characteristics, or features. Consequently, a CBIR system is not necessarily capable of understanding image semantics. Image semantic classification, on the other hand, is a technique for classifying images based on their semantics. While image semantics classification is a limited form of image understanding, the goal of image classification is not to understand images the way human beings do, but merely to assign the image to a semantic class. We argue that image class membership can assist retrieval.

Minka and Picard [12] introduced a learning component in their CBIR system. The system internally generated many segmentations or groupings of each image's regions based on different combinations of features, then it learned which combinations best represented the semantic categories given as exemplars by the user. The system requires the supervised training of various parts of the image.

Although region-based systems aim 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 part of a 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 and could determine which features of an object are significant, it would be capable of fast and accurate search. However, due to the great difficulty of recognizing and classifying 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. More importantly, the semantic classification schemes can improve retrieval by using various matching schemes tuned to the semantic class of the query image.

One example of semantic classification is the identification of natural photographs versus artificial graphs generated by computer tools [29]. The classifier divides an image into blocks and classifies every block into either of the two classes. If the percentage of blocks classified as photograph is higher than a threshold, the image is marked as photograph; otherwise, text.

Other examples include the WIPE system to detect objectionable images developed by Wang et al. [29], motivated by an earlier system by Fleck et al. [5] of the University of California at Berkeley. WIPE uses training images and CBIR to determine if a given image is closer to the set of objectionable training images or the set of benign training images. The system developed by Fleck et al., however, is more deterministic and involves a skin filter and a human figure grouper.

Szummer and Picard [24] have developed a system to classify indoor and outdoor scenes. Classification over low-level image features such as color histogram and DCT coefficients is performed. A 90 percent accuracy rate has been reported over a database of 1,300 images from Kodak.

Other examples of image semantic classification include city versus landscape [26] and face detection [1]. Wang and Fischler [30] have shown that rough, but accurate semantic understanding, can be very helpful in computer vision tasks such as image stereo matching.

1.3 Overview of the SIMPLIcity System

CBIR is a complex and challenging problem spanning diverse disciplines, including computer vision, color perception, image processing, image classification, statistical clustering, psychology, human-computer interaction (HCI), and specific application domain dependent criteria. While we are not claiming to be able to solve all the problems related to CBIR, we have made some advances towards the final goal, close to human-level automatic image understanding and retrieval performance.

In this paper, we discuss issues related to the design and implementation of a semantics-sensitive CBIR system for picture libraries. An experimental system, the SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) system, has been developed to validate the methods. We summarize the main contributions as follows.

1.3.1 Semantics-Sensitive Image Retrieval

The capability of existing CBIR systems is limited in large part by fixing a set of features used for retrieval. Apparently, different image features are suitable for the retrieval of images in different semantic types. For example, a color layout indexing method may be good for outdoor pictures, while a region-based indexing approach is much better for indoor pictures. Similarly, global texture matching is suitable only for textured pictures.

We propose a *semantics-sensitive* approach to the problem of searching general-purpose image databases. Semantic classification methods are used to categorize images so that semantically-adaptive searching methods applicable to each category can be applied. At the same time, the system can narrow down the searching range to a subset of the original database to facilitate fast retrieval. For example, automatic classification methods can be used to categorize a general-purpose picture library into semantic classes including "graph," "photograph," "textured," "nontextured," "benign," "objectionable," "indoor," "outdoor," "city," "landscape," "with people," and "without people." In our experiments, we used textured-nontextured and graph-photograph classification methods. We apply a

suitable feature extraction method and a corresponding matching metric to each of the semantic classes. When more classification methods are utilized, the current semantic classification architecture may need to be improved.

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.

1.3.2 Image Classification

For the purpose of searching picture libraries such as those on the Web or in a patient digital library, we are initially focusing on techniques to classify images into the classes "textured" versus "nontextured," "graph" versus "photograph." Several other classification methods have been previously developed elsewhere, including "city" versus "landscape" [26], and "with people" versus "without people" [1]. In this paper, we report on several classification methods we have developed and their performance.

1.3.3 Integrated Region Matching (IRM) Similarity Measure

Besides using semantics classification, another strategy of SIMPLIcity to better capture the image semantics is to define a robust region-based similarity measure, the Integrated Region Matching (IRM) metric. It incorporates the properties of all the segmented regions so that information about an image can be fully used to gain robustness against inaccurate segmentation. Image segmentation is an extremely difficult process and is still an open problem in computer vision. For example, an image segmentation algorithm may segment an image of a dog into two regions: the dog and the background. The same algorithm may segment another image of a dog into six regions: the body of the dog, the front leg(s) of the dog, the rear leg(s) of the dog, the eye(s), the background grass, and the sky.

Traditionally, region-based matching is performed on individual regions [2], [11]. The IRM metric we have developed has the following major advantages:

1. Compared with retrieval based on individual regions, the overall "soft similarity" approach in IRM reduces the adverse effect of inaccurate segmentation, an important property lacked by previous systems.
2. In many cases, knowing that one object usually appears with another helps to clarify the semantics of a particular region. For example, flowers typically appear with green leaves, and boats usually appear with water.
3. By defining an overall image-to-image 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. If desired, the system can be added with a function allowing users to query based on a specific region or a few regions.

1.4 Outline of the Paper

The remainder of the paper is organized as follows: The semantics-sensitive architecture is further introduced in Section 2. The image segmentation algorithm is described in Section 3. Classification methods are presented in Section 4. The IRM similarity measure based on segmentation is defined in Section 5. In Section 6, experiments and results are described. We conclude and suggest future research in Section 7.

2 Semantics-Sensitive Architecture

The architecture of the SIMPLIcity retrieval system is presented in Fig. 1. During indexing, the system partitions an image into 4×4 pixel blocks and extracts a feature vector for each block. A statistical clustering [8] algorithm is then used to quickly segment the image into regions. The segmentation result is fed into a classifier that decides the semantic type of the image. An image is currently classified as one of the n manually-defined mutually exclusive and collectively exhaustive semantic classes. The system can be extended to one that classifies an image softly into multiple classes with probability assignments. Examples of semantic types are indoor-outdoor, objectionable-benign, textured-nontextured, city-landscape, with-without people, and graph-photograph images. Features reflecting 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 as indicated by the user interface, 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.

3 The Image Segmentation Method

In this section, we describe the image segmentation procedure based on the k-means algorithm [8] using color and spatial variation features. For general-purpose images such as the images in a photo library or on the World Wide Web (WWW), automatic image segmentation is almost as

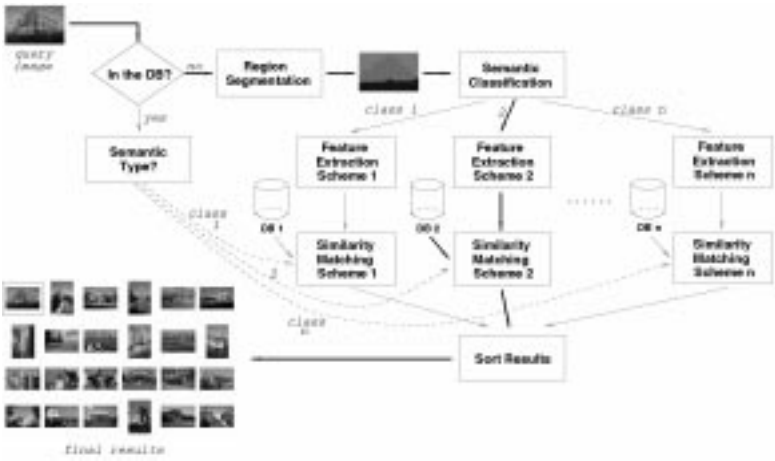


Fig. 1. The architecture of feature indexing process. The heavy lines show a sample indexing path of an image.

difficult as automatic image semantic understanding. The segmentation accuracy of our system is not crucial because an integrated region-matching (IRM) scheme is used to provide robustness against 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. Since the block size is small and boundary blockyness has little effect on retrieval, we choose blockwise segmentation rather than pixelwise segmentation to lower computational cost significantly.

Suppose observations are $\{x_i : i = 1, \ldots, L\}$. The goal of the k-means algorithm is to partition the observations into k groups with means $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k$ such that

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

is minimized. 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. A low $D(k)$ indicates high purity in the clustering process. The threshold is not critical because the IRM measure is not sensitive to k.
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 $k = 2, 3$. A low $D(k) - D(k-1)$ indicates convergence in the clustering process. The threshold determines the overall time to segment images and needs to be set to a near-zero value. It is critical to the speed, but not the quality of the final image segmentation. The threshold can be adjusted according to the experimental runtime.
3. The number k exceeds an upper bound. We allow an image to be segmented into a maximum of 16 segments. That is, we assume an image has less than 16 distinct types of objects. Usually, the segmentation process generates much less number of segments in an image. The threshold is rarely met.

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 [3], 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. The block size is chosen to be 4×4 to compromise between the texture detail and the computation time.

To obtain the other three features, we apply either the Daubechies-4 wavelet transform or the Haar transform to the L component of the image. We use these two wavelet transforms because they have better localization properties and require less computation compared to Daubechies' wavelets with longer filters. After a one-level wavelet transform, a 4×4 block is decomposed into four frequency bands, as shown in Fig. 2. 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 these features is their reflection of texture properties. Moments of wavelet coefficients in various frequency bands have proven effective for discerning texture [25]. The intuition behind this is that coefficients in different frequency bands signal variations in

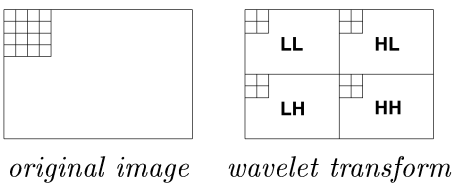


Fig. 2. Decomposition of images into frequency bands by wavelet transforms.

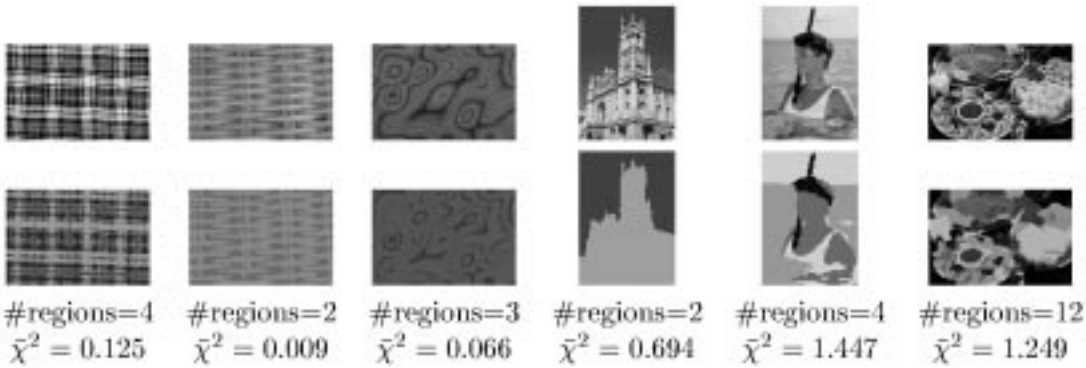


Fig. 3. Segmentation results by the k-means clustering algorithm: First row: original images. Second row: regions of the images. Results for other images in the database can be found online.

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. This texture feature is a good compromise between computational complexity and effectiveness.

Examples of segmentation results for both textured and nontextured images are shown in Fig. 3. 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 450MHz PC using the Linux operating system. We do not apply postprocessing to smooth region boundaries or to delete small isolated regions because these errors rarely cause degradation in the performance of our retrieval system, which is designed to tolerate inaccurate segmentation. Additionally, postprocessing usually costs a large amount of computation.

4 The Image Classification Methods

The image classification methods described in this section have been developed mainly for searching picture libraries such as Web images. We are initially interested in classifying images into the classes textured versus nontextured, graph versus photograph, and objectionable versus benign. Karu et al. provided an overview of texture-related research [10]. Other classification methods such as city versus landscape [26] and with people versus without people [1] were developed elsewhere.

4.1 Textured versus Nontextured Classification

In this section, we describe the algorithm to classify images into the semantic classes *textured* or *nontextured*. A *textured* image is defined as an image of a surface, a pattern of similarly-shaped objects, or an essential element of an object. For example, the structure formed by the threads of a fabric is a textured image. Fig. 4 shows some sample textured images. As textured images do not contain isolated objects or object clusters, the perception of such images focuses on color and texture, but not shape, which is critical for understanding nontextured images. Thus, an efficient retrieval system should use different features to depict these two types of images. To our knowledge, the problem of distinguishing textured images and nontextured images has not been explored in the literature.

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 Fig. 3, regions in textured images tend to scatter in the entire image, whereas nontextured 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.

We partition an image evenly into 16 zones, $\{Z_1, Z_2, \ldots, Z_{16}\}$. Suppose the image is segmented into regions $\{r_i : i = 1, \ldots, m\}$. For each region r_i, its percentage in zone Z_j is $p_{i,j}$, $\sum_{j=1}^{16} p_{i,j} = 1$, $i = 1, \ldots, m$. The uniform distribution over the zones should have probability mass function $q_j = 1/16$, $j = 1, \ldots, 16$. The χ^2 statistics for region i, χ_i^2, is computed by

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

The classification of textured or nontextured 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, nontextured. We randomly chose 100 textured images and 100 nontextured images and computed $\bar{\chi}^2$ for them. The histograms of $\bar{\chi}^2$ for the two types of images are shown in Fig. 5. It is shown that

Fig. 4. Sample textured images. (a) Surface texture. (b) Fabric texture. (c) Artificial texture. (d) Pattern of similarly-shaped objects.

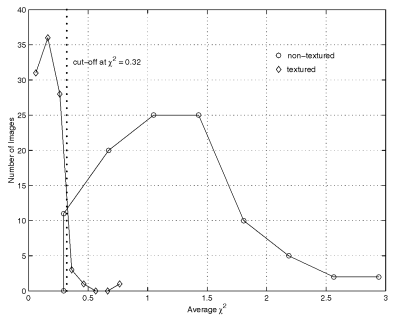


Fig. 5. The histograms of average χ^2's over 100 textured images and 100 nontextured images.

the two histograms differ prominently when $\bar{\chi}^2$ is slightly away from the decision threshold 0.32.

4.2 Graph versus Photograph Classification

An image is a *photograph* if it is a continuous-tone image. A *graph image* is an image containing mainly text, graph, and overlays. We have developed a graph-photograph classification method. This method is important for retrieving general-purpose picture libraries.

The classifier partitions an image into blocks and classifies every block into either of the two classes. If the percentage of blocks classified as photograph is higher than a threshold, the image is marked as photograph; otherwise, text. The algorithm we used to classify image blocks is based on a probability density analysis of wavelet coefficients in high frequency bands. For every block, two feature values, which describe the distribution pattern of the wavelet coefficients in high frequency bands, are evaluated. Then, the block is marked as a corresponding class according to the two feature values.

We tested the classification method on a database of 12,000 photographic images and a database of 300 randomly downloaded graph-based image maps from the Web. We achieved 100 percent sensitivity for photographic images and higher than 95 percent specificity.

5 The IRM Similarity Measure

In this section, the integrated region matching (IRM) measure of image similarity is described. IRM measures the overall similarity between images by integrating properties of all the regions in the images. An advantage of the overall similarity measure is the robustness against poor segmentation (Fig. 6), an important property lacked in previous work [2], [11].

Mathematically, defining a 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 a feature space can be easily defined by various measures, such as the Euclidean distance, it is not obvious how to define a distance between two sets of feature points. The distance should be sufficiently consistent with a person's concept of semantic "closeness" of two images.

We argue that a similarity measure based on region segmentation of images can be tolerant to inaccurate image segmentation if it takes all the regions in an image into consideration. To define the similarity measure, we first attempt to match regions in two images. Being aware that the segmentation process cannot be perfect, we "soften" the matching by allowing one region of an image to be matched to several regions of another image. Here, a region-to-region *match* is obtained when the regions are significantly similar to each other in terms of the features extracted.

The principle of matching is that the most similar 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. Fig. 7 illustrates the concept of IRM in a 3D feature space. The features we extract on the segmented regions are of high dimensions. The problem is more complex in a high-dimensional feature space.

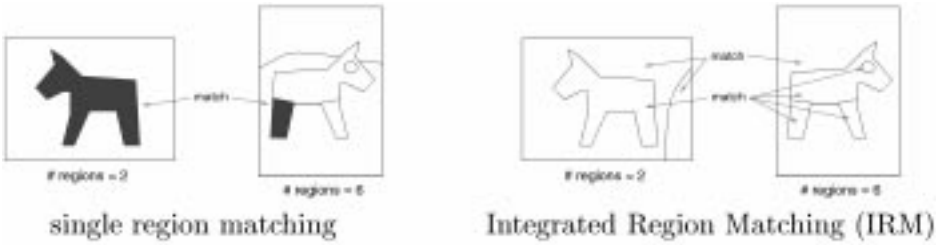


Fig. 6. Integrated Region Matching (IRM) is potentially robust to poor image segmentation.

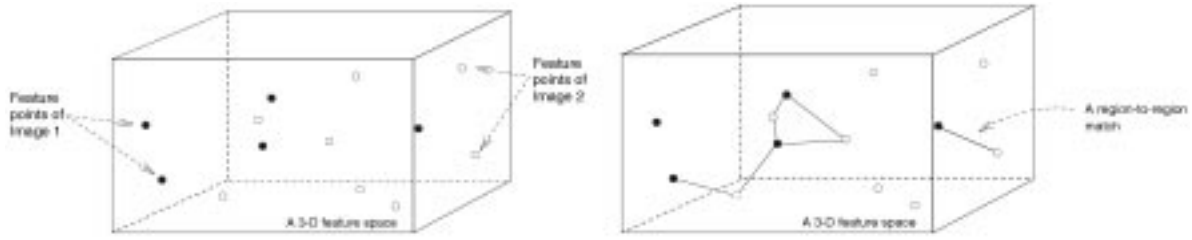


Fig. 7. Region-to-region matching results are incorporated in the Integrated Region Matching (IRM) metric. A 3D feature space is shown to illustrate the concept.

5.1 Integrated Region Matching (IRM)

Assume that Image 1 and 2 are represented by region sets $R_1 = \{r_1, r_2, \ldots, r_m\}$ and $R_2 = \{r'_1, r'_2, \ldots, r'_n\}$, 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. Consider a scenario of judging the similarity of two animal photographs. We usually compare the animals in the images before comparing 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 to evaluating similarity since 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, a region is allowed 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

$$S = \begin{pmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,n} \\ s_{2,1} & s_{2,2} & \cdots & s_{2,n} \\ \cdots & \cdots & \cdots & \cdots \\ s_{m,1} & s_{m,2} & \cdots & s_{m,n} \end{pmatrix}, \tag{3}$$

is referred to as the significance matrix.

A graphical explanation of the integrated matching scheme is provided in Fig. 8. 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 represented by 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 in the same image or the significance credit of matching them is zero. Every match 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.,

$$d(R_1, R_2) = \sum_{i,j} s_{i,j} d_{i,j}. \tag{4}$$

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

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

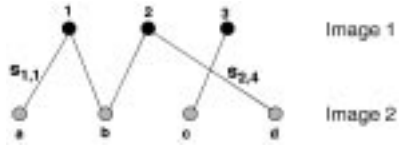


Fig. 8. Integrated region matching (IRM) allows a region in an image to be matched with several regions in another image.

$$\sum_{j=1}^{n} s_{i,j} = p_i, \ i = 1, \ldots, m \tag{5}$$

$$\sum_{i=1}^{m} s_{i,j} = p'_j, \ j = 1, \ldots, n. \tag{6}$$

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. 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. We call this the "most similar highest priority (MSHP)" principle. 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 regions 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:

$$\sum_{j=1}^{n} s_{i,j} = p_i \quad 1 \leq i \leq m, \ i \neq i' \tag{7}$$

$$\sum_{i:1 \leq i \leq m, i \neq i'} s_{i,j} = p'_j \quad 1 \leq j \leq n, \ j \neq j' \tag{8}$$

$$\sum_{i:1 \leq i \leq m, i \neq i'} s_{i,j'} = p'_{j'} - p_{i'} \tag{9}$$

$$s_{i,j} \geq 0 \quad 1 \leq i \leq m, \ i \neq i'; \ 1 \leq j \leq n. \tag{10}$$

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

1. Set $\mathcal{L} = \{\}$, denote
$$\mathcal{M} = \{(i, j) : i = 1, \ldots, m; \ j = 1, \ldots, n\}.$$
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.

Consider an example of applying the integrated region matching algorithm. Assume that $m = 2$ and $n = 3$. The values of p_i and p'_j are: $p_1 = 0.4$, $p_2 = 0.6$, $p'_1 = 0.2$, $p'_2 = 0.3$, $p'_3 = 0.5$.

The region distance matrix $\{d_{i,j}\}$, $i = 1, 2$, $j = 1, 2, 3$, is

$$\begin{pmatrix} 0.5 & 1.2 & 0.1 \\ 1.0 & 1.6 & 2.0 \end{pmatrix}.$$

The sorted $d_{i,j}$ is

$$\begin{array}{lcccccc} (i,j): & (1,3) & (1,1) & (2,1) & (1,2) & (2,2) & (2,3) \\ d_{i,j}: & 0.1 & 0.5 & 1.0 & 1.2 & 1.6 & 2.0. \end{array} \quad (11)$$

The first two regions matched are regions 1 and 3. As the significance of region 1, p_1, is fulfilled by the matching, region 1 in Image 1 is no longer in consideration. The second pair of regions matched is then regions 2 and 1. The region pairs are listed below in the order of being matched:

$$\begin{array}{lcccc} \text{region pairs}: & (1,3) & (2,1) & (2,2) & (2,3) \\ \text{significance}: & 0.4 & 0.2 & 0.3 & 0.1. \end{array} \quad (12)$$

The significance matrix is

$$\begin{pmatrix} 0.0 & 0.0 & 0.4 \\ 0.2 & 0.3 & 0.1 \end{pmatrix}.$$

Now, we 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 Image 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

$$d(R_1, R_2) = \sum_{i,j} s_{i,j} w_{i,j} d_{i,j}. \quad (13)$$

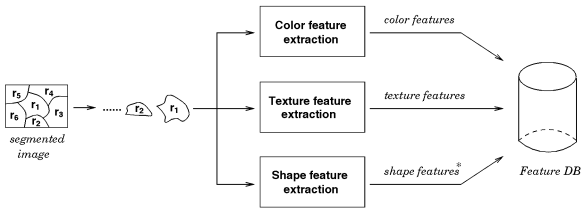


Fig. 9. Feature extraction in the SIMPLIcity system. (* The computation of shape features is omitted for textured images.)

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.

5.2 Distance between Regions

Now, we 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 Fig. 9. We have described 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. These features are denoted as: f_1, f_2, and f_3 for the averages in L, U, V components of color, respectively; f_4, f_5, and f_6 for the square roots of the 2nd-order moment of wavelet coefficients in the HL band, the LH band, and the HH band, respectively.

To describe shape, normalized inertia [6] 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

$$l(H, \gamma) = \frac{\int_H \|x - \hat{x}\|^\gamma dx}{[V(H)]^{1+\gamma/k}}, \quad (14)$$

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,

$$l(H, \gamma) = \frac{\sum_{x:x\in H} \|x - \hat{x}\|^\gamma}{[V(H)]^{1+\gamma/k}}, \quad (15)$$

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:

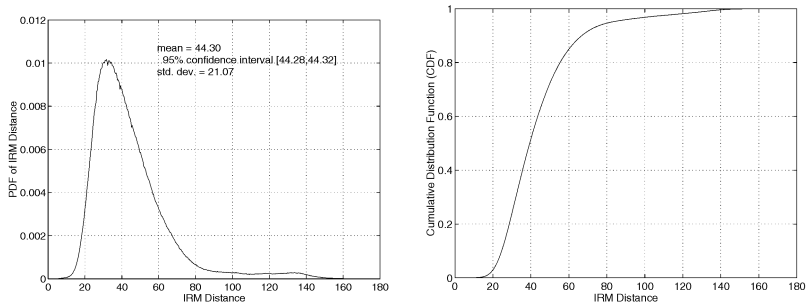
$$f_7 = l(H, 1)/L_1\ , \quad f_8 = l(H, 2)/L_2\ , \quad f_9 = l(H, 3)/L_3. \quad (16)$$

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

$$d(r, r') = \sum_{i=1}^{6} w_i (f_i - f'_i)^2. \quad (17)$$

For nontextured images, $d(r, r')$ is defined as

$$d(r, r') = g(d_s(r, r')) \cdot d_t(r, r'), \quad (18)$$



Empirical pdf Empirical cdf

Fig. 10. The empirical pdf and cdf of the IRM distance.

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

$$d_s(r, r') = \sum_{i=7}^{9} w_i (f_i - f_i')^2 \tag{19}$$

and $d_t(r, r')$ is the color and texture distance defined equally as the distance between textured image regions, i.e.,

$$d_t(r, r') = \sum_{i=1}^{6} w_i (f_i - f_i')^2. \tag{20}$$

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

$$g(d) = \begin{cases} 1 & d \geq 0.5 \\ 0.85 & 0.2 < d \leq 0.5 \\ 0.5 & d < 0.2. \end{cases} \tag{21}$$

It is observed that, when $d_s(r, r') \geq 0.5$, the two regions bear little resemblance and, hence, distinguishing the extent of similarity by $d_s(r, r')$ is not meaningful. Thus, we set $g(d) = 1$ for d greater than the threshold 0.5. 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 define $g(d)$ as a piecewise 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 demonstrated by 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.

5.3 Characteristics of IRM

To study the characteristics of the IRM distance, we performed 100 random queries on our COREL photograph data set. Based on the 5.6 million IRM distances obtained, we estimated the distribution of the IRM distance. The empirical mean of the IRM is 44.30, with a 95 percent confidence interval of $[44.28, 44.32]$. The standard deviation of the IRM is 21.07. Fig. 10 shows the empirical probability distribution function (pdf) and the empirical cumulative distribution function (cdf).

Based on this empirical distribution of the IRM, we may give more intuitive similarity distances to the end user than the distances themselves using the *similarity percentile*. As shown in the empirical cumulative distribution function, an IRM distance of 15 represents approximately 1 percent of the images in the database. We may notify the user that two images are considered to be very close when the IRM distance between the two images is less than 15. Likewise, we may advise the user that two images are considerably different when the IRM distance between the two images is greater than 50.

6 Experiments

The SIMPLIcity system has been implemented with a general-purpose image database including about $200,000$ pictures, which are stored in JPEG format with size 384×256 or 256×384. The system uses no textual information in the matching process because we try to explore the possible advances of CBIR. In a real-world application, however, textual information is often used as a helpful addition to CBIR systems. Two classification methods, graph-photograph and textured-nontextured, have been used in our experiments. Adding more classification methods into the system may introduce problems to the accuracy of the retrieval.

For each image, the features, locations, and areas of all its regions are stored. Images of different semantic classes are stored in separate databases. Because the EMD-based color histogram system [18] and the WBIIS system are the only other systems we have access to, we compare the accuracy of the SIMPLIcity system to these systems using the same COREL database. WBIIS had been compared with the original IBM QBIC system and found to perform better [28]. It is difficult to design a fair comparison with existing region-based searching algorithms such as the Blobworld system and the NeTra system which depends on additional information to be provided by the user during the process. As a future work, we will try to compare our system with other existing systems such as the VisualSeek system developed by Columbia Univerisity.

With the Web, online demonstration has become a popular direction in letting user evaluate CBIR systems. An online demonstration is provided.[1] Readers are encouraged to compare the performance of SIMPLIcity with other systems. A list of online image retrieval demonstration Web sites can be found on our site.

1. URL: http://wang.ist.psu.edu.

The current implementation of the SIMPLIcity system provides several query interfaces: a CGI-based Web access interface, a JAVA-based drawing interface, and a CGI-based Web interface for submitting a query image of any format anywhere on the Internet.

6.1 Accuracy

We evaluated the accuracy of the system in two ways. First, we used a 200,000-image COREL database to compare with existing systems such as EMD-based color histogram and WBIIS. Then, we designed systematic evaluation methods to judge the performance statistically. The SIMPLIcity system has demonstrated much improved accuracy over the other systems.

6.2 Query Comparison

We compare the SIMPLIcity system with the WBIIS (Wavelet-Based Image Indexing and Searching) system [28] with the same image database. In this section, we show the comparison results using query examples. Due to the limitation of space, we show only two rows of images with the top 11 matches to each query. At the same time, we provide the number of related images in the top 29 matches (i.e., the first screenful) for each query. We chose the numbers "11" and "29" before viewing the results. In the next section, we provide numerical evaluation results by systematically comparing several systems.

For each query example, we manually examine the precision of the query results. The relevance of image semantics depends on the point-of-view of the reader. We use our judgments here to determine the relevance of images. In each query, we decide the relevance to the query image before viewing the query results. We admit that our relevance criteria, specified in the caption of Fig. 11, may be very different from the criteria used by a user of the system.

As WBIIS forms 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 Fig. 11a. 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. The other two numbers are the value of the similarity measure between the query image and the matched image, and the number of regions in the image. To view the images better or to see more matched images, users can visit the demonstration Web site and use the query image ID to repeat the retrieval.

SIMPLIcity also gives higher precision within the best 11 or 29 matches for images composed of fine details. Retrieval results with a photo of a hamburger as the query are shown in Fig. 11b. 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. It is often impossible to define the relevance between two given images. For example, the user may be interested in finding other hamburger images and not food images. Returning food images is not likely to be more helpful to the user than returning other images. The top match made by SIMPLIcity is also a photo of hamburger which is also 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 multilevel 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 image queries. The SIMPLIcity system, however, classifies images first and tries to prevent images classified as textured images to be matched to images classified as nontextured images. The method relies on highly accurate classifiers. In practice, a classifier can give wrong classification results, which lead to wrong retrieval.

Another three query examples are compared in Figs. 11c, 11d, and 11e. The query images in Figs. 11c and 11d 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 better retrieval, based on the relevance criteria we have used. For the query in Fig. 11c, only the third matched image is not a picture of a person. A few images, the first, fourth, seventh, and eighth matches, depict a similar topic as well, probably about life in Africa. The query in Fig. 11e also shows the advantages of SIMPLIcity. The system finds photos of similar flowers with different sizes and orientations. Only the ninth 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 Fig. 12. The granular surface in the query image is matched more accurately by the SIMPLIcity system. We performed another test on this query using SIMPLIcity system without the image classification component. As shown in Fig. 12, the degraded system found several nontextured pictures (e.g., sunset scenes) for this textured query picture.

Typical CBIR systems do not perform well when the image databases contain both photographs and graphs. Graphs, such as clip art pictures and image maps, appear frequently on the Web. The semantics of clip art pictures are typically more abstract and significantly different from photos with similar low-level visual features, such as the color histogram. For image maps on the Web, an indexing method based on Optical Character Recognition (OCR) may be more efficient than CBIR systems based on visual features. SIMPLIcity classifies picture libraries into graphs and photographs using image segmentation and statistical hypothesis testing before the feature indexing step. Fig. 13 shows the result of a clip art query. All the best 11 matches of this 200,000-picture database are clip art pictures, many with similar semantics.

6.3 Systematic Evaluation

6.3.1 Performance on Image Queries

To provide numerical results, we tested 27 sample images chosen randomly from nine categories, each containing three of the images. Image matching is performed on the COREL database of 200,000 images. A retrieved image is considered a match if it belongs to the same category of the query image. The categories of images tested are listed in Table 1a. Most categories simply include images containing the specified objects. Images in the "sports and public events" class contain people in a game or public event, such as a festival. Portraits are not included in this category. The

Fig. 11. Comparison of SIMPLIcity and WBIIS. The query image is the upper-left corner image of each block of images. Due to the limitation of space, we show only two rows of images with the top 11 matches to each query. More matches can be viewed from the online demonstration site. (a) Natural out-door scene, (b) food, (c) people, (d) portrait, and (e) flower.

"landscape with buildings" class refers to outdoor scenes featuring man-made constructions such as buildings and sculptures. The "beach" class refers to scenery at coasts or river banks. For the "portrait" class, an image has to show people as the main feature. A scene with human beings as a minor part is not included.

Precision was computed for both SIMPLIcity and WBIIS. Recall was not calculated because the database is large and it is difficult to estimate the total number of images in one category, even approximately. In the future, we will develop a large-scale sharable test database to evaluate the recall.

To account for the ranks of matched images, the average of the precision values within k retrieved images, $k = 1, \ldots, 100$, is computed. That is, $\bar{p} = \frac{1}{100}\sum_{k=1}^{100} \frac{n_k}{k}$ and n_k is the number of matches in the first k retrieved images. This average precision is called the "weighted precision"

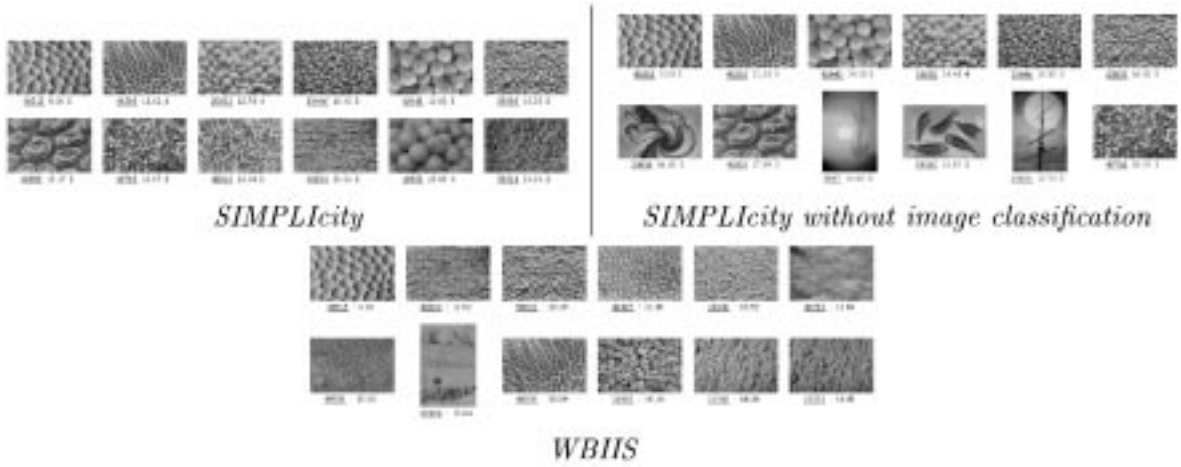


Fig. 12. SIMPLIcity gives better results than the same system without the classification component. The query image is a textured image.

because it is equivalent to a weighted percentage of matched images with a larger weight assigned to an image retrieved at a higher rank. For instance, a relevant image appearing earlier in the list of retrieved images would enhance the weighted precision more significantly than if it appears later in the list.

For each of the nine image categories, the average precision and weighted precision based on the three sample images are plotted in Fig. 14. The image category identification number is indicated in Table 1a. Except for the tools and toys category, in which case the two systems perform about equally well, SIMPLIcity has achieved better results than WBIIS measured in both ways. For the two categories of landscape with buildings and vehicle, the difference between the two systems is quite significant. On average, the precision and the weighted precision of SIMPLIcity are higher than those of WBIIS by 0.227 and 0.273, respectively.

6.3.2 Performance on Image Categorization

The SIMPLIcity system was also evaluated based on a subset of the COREL database, formed by 10 image categories (shown in Table 1b), 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 subdatabase was tested as a query and the retrieval ranks of all the rest images were recorded. Three statistics were computed for each query: 1) the precision within the first 100 retrieved images, 2) the mean rank of all the matched images, and 3) the standard deviation of the ranks of matched images.

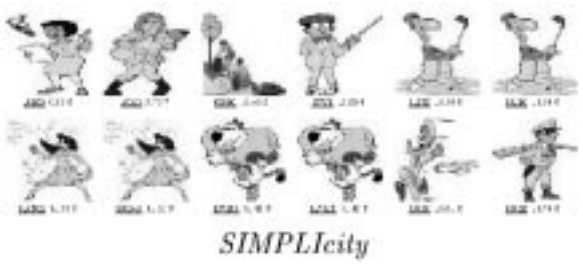


Fig. 13. SIMPLIcity does not mix clip art pictures with photographs. A graph-photograph classification method using image segmentation and statistical hypothesis testing is used. The query image is a clip art picture.

The recall within the first 100 retrieved images is identical to the precision in this special case. The total number of semantically related images for each query is fixed to be 100. The average performance for each image category is computed in terms of the three statistics: p (precision), r (the mean rank of matched images), and σ (the standard deviation of the ranks of matched images). For a system that ranks images randomly, the average p is about 0.1, and the average r is about 500. An ideal CBIR system should demonstrate an average p of 1 and an average r of 50.

Similar evaluation tests were carried out for the state-of-the-art EMD-based color histogram match. We used LUV color space and a matching metric similar to the EMD described in [18] 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. Fig. 15 shows the performance when compared to the SIMPLIcity system. Clearly, both of the two color histogram-based matching systems perform much worse than the SIMPLIcity region-based CBIR 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 practically impossible to obtain matches on databases with more than 50,000 images. For this reason, we cannot evaluate this system using the COREL database of 200,000 images and the 27 sample query images described in the previous section. SIMPLIcity runs at about twice the speed of the relatively fast Color Histogram 1 system and still provides much better searching accuracy than the extremely slow Color Histogram 2 system.

6.4 Robustness

We have performed extensive experiments on the robustness of the system. Figs. 17 and 18 summarize the results. The graphs in the first row show the changes in ranking of the

TABLE 1
COREL Categories of Images Tested

ID	Category Name
1	Sports and public events
2	Beach
3	Food
4	Landscape with buildings
5	Portrait
6	Horses
7	Tools and toys
8	Flowers
9	Vehicle

(a)

ID	Category Name
1	Africa people and villages
2	Beach
3	Buildings
4	Buses
5	Dinosaurs
6	Elephants
7	Flowers
8	Horses
9	Mountains and glaciers
10	Food

(b)

(a) Test 1. (b) Test 2.

target image as we increase the significance of image alterations. The graphs in the second row show the the changes in IRM distance between the altered image and the target image as we increase the significance of image alterations.

The system is fairly robust to image alterations such as intensity variation, sharpness variation, intentional color distortions, other intentional distortions, cropping, shifting, and rotation. Fig. 16 shows some query examples, using the 200,000-image COREL database. On average, the system is robust to approximately 10 percent brightening, 8 percent darkening, blurring with a 15×15 Gaussian filter, 70 percent sharpening, 20 percent more saturation, 10 percent 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.

6.4.1 Speed

The algorithm has been implemented on a Pentium III 450MHz PC using the Linux operating system. To compute the feature vectors for the $200,000$ color images of size 384×256 in our general-purpose image database requires approximately 60 hours. On average, one second is needed to segment an image and to compute the features of all regions. The speed is much faster than other region-based methods. Fast indexing has provided us with the capability of handling external queries and sketch queries in real time.

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 200,000-image database using the IRM similarity measure. If the query image is not already in the database, one extra second of CPU time is spent to extract the feature from the query image.

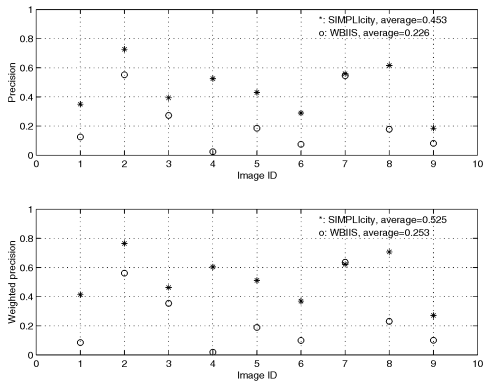


Fig. 14. Comparison of SIMPLIcity and WBIIS: average precision and weighted precision of nine image categories.

7 Conclusions and Future Work

In this work, we experimented with the idea that images can be classified into global semantic classes, such as textured or nontextured, graph or photograph, and that much can be gained if the feature extraction scheme is tailored to best suit each class. For the purpose of searching general-purpose image databases, we have developed a series of statistical image classification methods, including the graph-photograph, textured-nontextured classifiers. We have explored the application of advanced wavelets in feature extraction. We have developed an image region segmentation algorithm using wavelet-based feature extraction and the k-means statistical clustering algorithm. Finally, we have developed a measure for the overall similarity between images, i.e., the Integrated Region Matching (IRM) measure, defined based on a region-matching scheme that integrates properties of all the regions in the images, resulting in a simple querying interface. The advantage of using such a soft matching is the improved robustness against poor segmentation, an important property overlooked in previous work.

The application of SIMPLIcity to a database of about 200,000 general-purpose images shows more accurate and much faster retrieval compared with the existing algorithms. An important feature of the algorithms implemented in SIMPLIcity is that it is fairly robust to intensity variations, sharpness variations, color distortions, other distortions, cropping, scaling, shifting, and rotation. The system is also easier to use than other region-based retrieval systems.

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

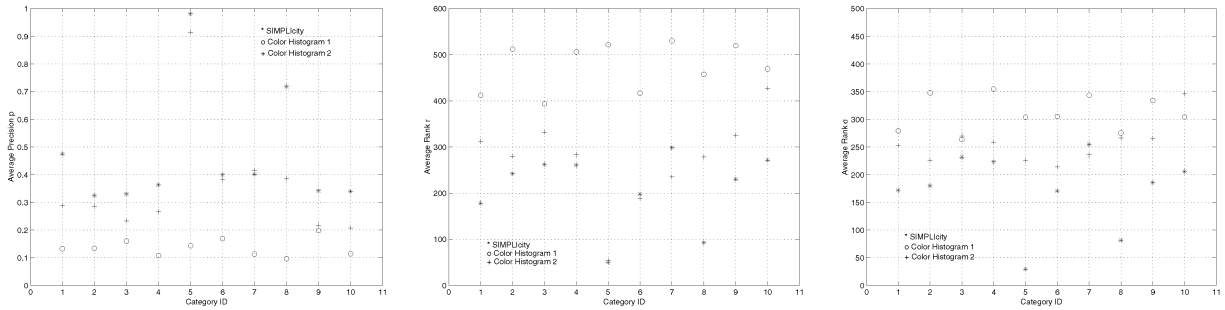


Fig. 15. Comparing SIMPLIcity with color histogram methods on average precision p, average rank of matched images r, and the standard deviation of the ranks of matched images σ. *The lower numbers indicate better results for the last two plots (i.e., the r plot and the σ plot).* Color Histogram 1 gives 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.

IRM distance should be computed after merging the matched regions.

3. The statistical semantic classification methods do not distinguish images in different classes perfectly. Furthermore, an image may fall into several semantic classes simultaneously.
4. The querying interfaces are not powerful enough to allow users to formulate their queries freely. For different user domains, the query interfaces should ideally provide different sets of functions.

A 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

Fig. 16. The robustness of the system to image alterations. Due to space, only the best five matches are shown. The first image in each example is the query image. Database size: 200,000 images.

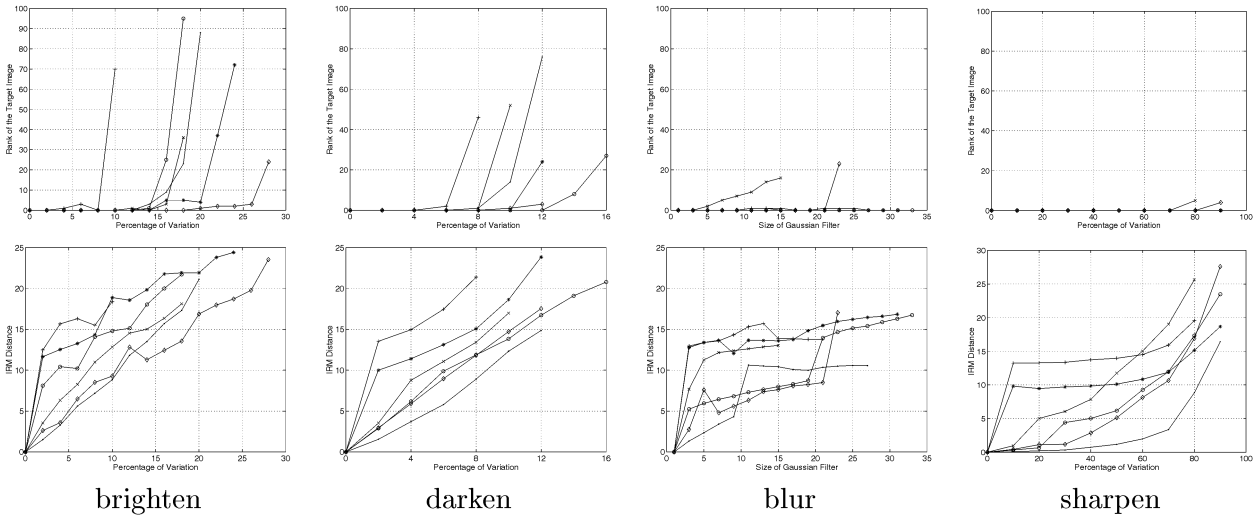


Fig. 17. The robustness of the system compared to image alterations. Six query images were randomly selected from the database. Each curve represents the robustness on one of the six images.

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.

A statistical soft classification architecture can be developed to allow an image to be classified based on its probability of belonging to a certain semantic class. We need to design more high-level classifiers. The speed can be improved significantly by adopting a feature clustering scheme or using a parallel query processing scheme. We need to continue our effort in designing simple but capable graphical user interfaces. We are planning to build a sharable testbed for statistical evaluation of different CBIR systems. Experiments with a WWW image database or a video database could be another interesting study.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under grant IIS-9817511. Research was performed while J.Z. Wang and J. Li were at Stanford University. The authors would like to thank Shih-Fu Chang, Oscar Firschein, Martin A. Fischler, Hector Garcia-Molina, Yoshinori Hara, Kyoji Hirata, Quang-Tuan Luong, Wayne Niblack, and Dragutin Petkovic for valuable discussions on content-based image retrieval, image understanding, and photography. We would also like to acknowledge the comments and constructive suggestions from anonymous reviewers and the associate editor. Finally, we thank Thomas P. Minka for providing us the source codes of the MIT Photobook.

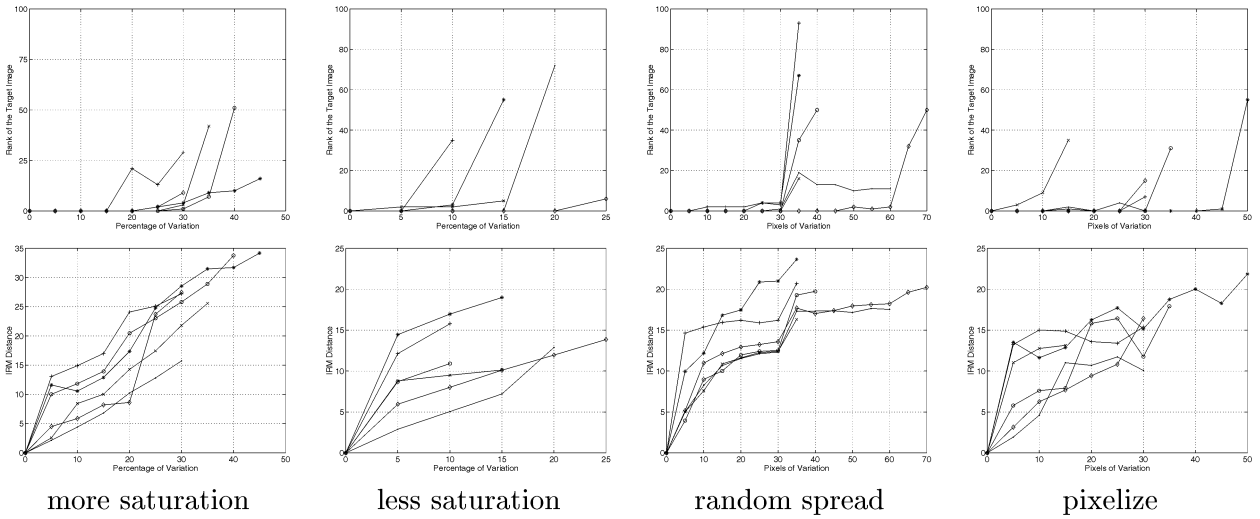


Fig. 18. The robustness of the system compared to image alterations.

REFERENCES

[1] M.C. Burl, M. Weber, and P. Perona, "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry," *Proc. European Conf. Computer Vision,* pp. 628-641, June 1998.

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

[3] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia: SIAM, 1992.

[4] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom et al. "Query by Image and Video Content: The QBIC System," *IEEE Computer,* vol. 28, no. 9, 1995.

[5] M. Fleck, D.A. Forsyth, and C. Bregler, "Finding Naked People," *Proc. European Conf. Computer Vision,* vol. 2, pp. 593-602, 1996.

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

[7] A. Gupta and R. Jain, "Visual Information Retrieval," *Comm. ACM,* vol. 40, no. 5, pp. 70-79, May 1997.

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

[9] R. Jain, S.N.J. Murthy, P.L.-J. Chen, and S. Chatterjee, "Similarity Measures for Image Databases," *Proc. SPIE,* vol. 2420, pp. 58-65, Feb. 1995.

[10] K. Karu, A.K. Jain, and R.M. Bolle, "Is There any Texture in the Image?" *Pattern Recognition,* vol. 29, pp. 1437-1446, 1996.

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

[12] T.P. Minka and R.W. Picard, "Interactive Learning Using a Society of Models," *Pattern Recognition,* vol. 30, no. 3, p. 565, 1997.

[13] S. Mukherjea, K. Hirata, and Y. Hara, "AMORE: A World Wide Web Image Retrieval Wngine," *Proc. World Wide Web,* vol. 2, no. 3, pp. 115-132, 1999.

[14] A. Natsev, R. Rastogi, and K. Shim, "WALRUS: A Similarity Retrieval Algorithm for Image Databases," *SIGMOD Record,* vol. 28, no. 2, pp. 395-406, 1999.

[15] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proc. SPIE,* vol. 2185, pp. 34-47, Feb. 1994.

[16] E.G.M. Petrakis and A. Faloutsos, "Similarity Searching in Medical Image Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 9, no. 3, pp. 435-447, May/June 1997.

[17] R.W. Picard and T. Kabir, "Finding Similar Patterns in Large Image Databases," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* vol. 5, pp. 161-164, 1993.

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

[19] G. Sheikholeslami, W. Chang, and A. Zhang, "Semantic Clustering and Querying on Heterogeneous Features for Visual Data," *Proc. ACM Multimedia,* pp. 3-12, 1998.

[20] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Proc. Computer Vision and Pattern Recognition,* pp. 731-737, June 1997.

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

[22] J.R. Smith and C.S. Li, "Image Classification and Querying Using Composite Region Templates," *Int'l J. Computer Vision and Image Understanding,* vol. 75, nos. 1-2, pp. 165-174, 1999.

[23] S. Stevens, M. Christel, and H. Wactlar, "Informedia: Improving Access to Digital Video," *Interactions,* vol. 1, no. 4, pp. 67-71, 1994.

[24] M. Szummer and R.W. Picard, "Indoor-Outdoor Image Classification," *Proc. Int'l Workshop Content-Based Access of Image and Video Databases,* pp. 42-51, Jan. 1998.

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

[26] A. Vailaya, A. Jain, and H.J. Zhang, "On Image Classification: City versus Landscape," *Proc. IEEE Workshop Content-Based Access of Image and Video Libraries,* pp. 3-8, June 1998

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

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

[29] J.Z. Wang, J. Li, G. Wiederhold, and O. Firschein, "System for Screening Objectionable Images," *Computer Comm.,* vol. 21, no. 15, pp. 1355-1360, 1998.

[30] J.Z. Wang and M.A. Fischler, "Visual Similarity, Judgmental Certainty and Stereo Correspondence," *Proc. DARPA Image Uunderstanding Workshop,* 1998.

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

Jia Li received the BS degree in electrical engineering from Xi'an JiaoTong University, China, in 1993, the MSc degree in electrical engineering in 1995, the MSc degree in statistics in 1998, and the PhD degree in electrical engineering in 1999, all from Stanford University. She is an assistant professor of statistics at The Pennsylvania State University. In 1999, she worked as a research associate in the Computer Science Department at Stanford University. She was a researcher at the Xerox Palo Alto Research Center from 1999 to 2000. Her research interests include statistical classification and modeling, data mining, image processing, and image retrieval. She is a member of the IEEE.

Gio Wiederhold received a degree in aeronautical engineering in Holland in 1957 and the PhD degree in medical information science from the University of California at San Francisco in 1976. He is a professor of computer science at Stanford University with courtesy appointments in medicine and electrical engineering. He has supervised 30 PhD theses and published more than 350 books, papers, and reports. He has been elected fellow of the ACMI, the IEEE, and the ACM. His current research includes privacy protection in collaborative settings, software composition, access to simulations to augment information systems, and developing an algebra over ontologies. Prior to his academic career, he spent 16 years in the software industry. His Web page is http://www-db.stanford.edu/people/gio.html.