Negotiating the Semantic Gap: From Feature Maps to Semantic Landscapes

William I. Grosky[1] and Rong Zhao[2]

[1] Computer and Information Science Department, University of Michigan-Dearborn, Dearborn, Michigan 48128
wgrosky@umich.edu
[2] Computer Science Department, State University of New York at Stony Brook, Stony Brook, New York 11794
roz@cs.sunysb.edu

Abstract. – In this paper, we present the results of our work that seeks to negotiate the gap between low-level features and high-level concepts in the domain of web document retrieval. This work concerns a technique, latent semantic indexing (LSI), which has been used for textual information retrieval for many years. In this environment, LSI determines clusters of co-occurring keywords, sometimes, called concepts, so that a query which uses a particular keyword can then retrieve documents perhaps not containing this keyword, but containing other keywords from the same cluster. In this paper, we examine the use of this technique for content-based web document retrieval, using both keywords and image features to represent the documents.

1 Introduction

The emergence of multimedia technology and the rapidly expanding image and video collections on the internet have attracted significant research efforts in providing tools for effective retrieval and management of visual data. Image retrieval is based on the availability of a representation scheme of image content. Image content descriptors may be visual features such as color, texture, shape, and spatial relationships, or semantic primitives.

Conventional information retrieval was based solely on text, and those approaches to textual information retrieval have been transplanted into image retrieval in a variety of ways. However, “a picture is worth a thousand words”. Image contents are much more versatile compared with texts, and the amount of visual data is already enormous and still expanding very rapidly. Hoping to cope with these special characteristics of visual data, content-based image retrieval methods have been introduced. It has been widely recognized that the family of image retrieval techniques should become an integration of both low-level visual features addressing the more detailed perceptual aspects and high-level semantic features underlying the more general conceptual aspects of visual data. Neither of these two types of features is sufficient to retrieve or manage visual data in an effective or efficient way [1].

Although efforts have been devoted to combining these two aspects of visual data, the gap between them is still a huge barrier in front of researchers. Intuitive and heuristic approaches do not provide us with satisfactory performance. Therefore, there is an urgent need of finding the latent correlation between low-level features and high-level concepts and merging them from a different perspective. How to find this new perspective and bridge the gap between visual features and semantic features has been a major challenge in this research field.

1.1 Image Retrieval

Image retrieval is an extension to traditional information retrieval. Its purpose is to retrieve images, from a data source (usually a database or the entire internet), that are relevant to a piece of query data. Approaches to image retrieval are somehow derived from conventional information retrieval and are designed to manage the more versatile and enormous amount of visual data which exists.

The different types of information items that are normally associated with images are as follows:

- Content-independent metadata: data that is not directly concerned with image content, but related to it. Examples are image format, author's name, date and location.
- Content metadata:
  - Content-dependent metadata: data referring to low-level or intermediate-level features, such as color, texture, shape, spatial relationships, and their various combinations.
  - Content-descriptive metadata: data referring to content semantics, concerned with relationships of image entities to real-world entities.

Low-level visual features such as color, texture, shape and spatial relationships are directly related to perceptual aspects of image content. Since it is usually easy to extract and represent these features and fairly convenient to design similarity measures by using the statistical properties of these features, a variety of content-based image retrieval techniques have been proposed in the past few years. High-level concepts, however, are not extracted directly from visual contents, but they represent the relatively more important meanings of objects and scenes in the images that are perceived by human beings. These conceptual aspects are more closely related to users' preferences and subjectivity. Concepts may vary significantly in different circumstances. Subtle changes in the semantics may lead to dramatic conceptual differences. Needless to say, it is a very challenging task to extract and manage meaningful semantics and to make use of them to achieve more intelligent and user-friendly retrieval. The next section analyzes these challenges in more detail.

1.2 Challenges

High-level conceptual information is normally represented by using text descriptors. Traditional indexing for image retrieval is text-based. In certain content-based retrieval techniques, text descriptors are also used to model perceptual aspects. However, the inadequacy of text description is very obvious:

- It is difficult for text to capture the perceptual saliency of visual features.
- It is rather difficult to characterize certain entities, attributes, roles or events by means of text only.
- Text is not well suited for modeling the correlation between perceptual and conceptual features.
- Text descriptions reflect the subjectivity of the annotator and the annotation process is prone to be inconsistent, incomplete, ambiguous, and very difficult to be automated.

Although it is an obvious fact that image contents are much more complicated than textual data stored in traditional databases, there is an even greater demand for retrieval and management tools for visual data, since visual information is a more capable medium of conveying ideas and is more closely related to human perception of the real world. Image retrieval techniques should provide support for user queries in an effective and efficient way, just as conventional information retrieval does for textual retrieval [2]. In general, image retrieval can be categorized into the following two types:

- Exact Matching – This category is applicable only to static environments or environments in which features of the image do not evolve over an extended period of time. Databases containing industrial and architectural drawings, or electronics schematics are examples of such environments.
- Similarity-Based Searching – In most cases, it is not quite obvious to know which images best satisfy the query. Different users may have different ideas. Even the same user may have different preferences under different circumstances. Thus, it is desirable to return the top several similar images based on the similarity measure, so as to give users a good sampling. User interaction plays an important role in this type of retrieval. Databases containing natural scenes or human faces are examples of such environments.

For either type of retrieval, the dynamic and versatile characteristics of image content require expensive computations and sophisticated methodologies in the areas of computer vision, image processing, data visualization, indexing, and similarity measurement. In order to manage image data effectively and efficiently, many schemes for data modeling and image representation have been proposed. Typically, each of these schemes builds a symbolic image for each given physical image to provide logical and physical data independence. Symbolic images are then used in conjunction with various index structures as proxies for image comparisons to reduce the searching scope. The high-dimensional visual data is usually reduced into a lower-dimensional subspace so that it is easier to index and manage the visual contents. Once the similarity measure has been determined, indexes of corresponding images are located in the image space and those images are retrieved from the

database. Due to the lack of any unified framework for image representation and retrieval, certain methods may perform better than others under certain query situations. Therefore, these schemes and retrieval techniques have to be somehow integrated and adjusted on the fly to facilitate effective and efficient image data management.

1.3 Research Goals

The work presented in this chapter aims to improve several aspects of content-based image retrieval by finding the latent correlation between low-level visual features and high-level semantics and integrating them into a unified vector space model. To be more specific, the significance of this approach is to design and implement an effective and efficient framework of image retrieval techniques, using a variety of visual features such as color, texture, shape and spatial relationships. Latent semantic indexing, an information retrieval technique, is incorporated with content-based image retrieval. By using this technique, we hope to extract the underlying semantic structure of image content and hence to bridge the gap between low-level visual features and high-level conceptual information. Improved retrieval performance and more efficient indexing structure can also be achieved. We have investigated the following issues in our preliminary research and the experimental results are very promising. Our goals are as follows:

- We aim to present a novel approach based on latent semantic indexing to image retrieval and explore how it helps to reveal the latent correlation between feature sets and semantic clusters.
- We aim to experiment with a special feature extraction method, namely, the *anglogram*, which captures the spatial distribution of feature points. This technique is based on the extraction of information from a Delauney triangulation of these feature points.
- We aim to present a unified framework to integrate multiple visual features including color histograms, shape anglograms and color anglograms with latent semantic indexing and demonstrate the efficacy of our image indexing scheme by comparing it with relevant image indexing techniques.
- We aim to incorporate textual annotation with visual features in the proposed framework of image retrieval and indexing to further our efforts of negotiating the semantic gap. Relevance feedback and other techniques will also be integrated.

The remaining part of this chapter is organized as follows. Section 2 introduces the feature extraction techniques applied in our approach. The theoretical background of latent semantic indexing and its application in textual information retrieval are detailed in Section 3. In Section 4, we present the preliminary results of our study of finding the latent correlation between features and semantics. Finally, Section 5 summarizes the chapter and highlights some proposed future work.

2 Features

In this section we present the feature extraction techniques that are applied in this research work. We propose to integrate a variety of visual features with the latent semantic indexing technique for image retrieval. These visual features include global and subimage color histograms, as well as anglograms. Anglograms can be used for shape-based and color-based representations, as well as for the spatial-relationships of image objects. Thus, a unified framework of image retrieval techniques is going to be generated in our proposed study.

2.1 Color Histogram

Color is a visual feature that is immediately perceived when looking at an image. Retrieval by color similarity requires that models of color stimuli are used, such that distances in the color space correspond to human perceptual distances between different colors. Moreover, color patterns must be represented in such a way that salient chromatic properties are captured.

A variety of color models have been introduced, such as *RGB, HSV, CIE, LUV and MTM*. Humans perceive colors through hue, saturation, and brightness. *Hue* describes the actual wavelength of the color. *Saturation* indicates the amount of white light that is present in a color. Highly saturated colors, also known as pure colors, have no white light component. *Brightness*, which is also called intensity, value, or lightness, represents the intensity of color. Since the combination of hue, saturation and value reflects human perception of color, the *HSV* color model has been selected to be the basis for our color-based extraction approach.

The *color histogram* is the most traditional and the most widely used way to represent color patterns in an image. It is a relatively efficient representation of color content and it is fairly insensitive to variations originated by camera rotation or zooming [1]. Also, it is fairly insensitive to changes in image resolution when images have quite large homogeneous regions, and insensitive to partial occlusions as well.

In our study, the *HSV* color histogram is generated for each image on either the whole image level or the subimage level. On whole image level, a two-dimensional global histogram of both the hue component and saturation component is computed. Since the human perception of color depends mostly on hue and saturation, we ignore the intensity value component in our preliminary research, in order to simplify the computation. Each image is first converted from the *RGB* color space to the *HSV* color space. For each pixel of the resulting image, hue and saturation are extracted and each quantized into a 10-bin histogram. Then, the two histograms h and s are combined into one $h \times s$ histogram with 100 bins, which is taken to be the representing feature vector of each image. This is a vector of 100 elements, $\mathbf{V} = [f_1, f_2, f_3, \ldots f_{100}]^T$, where each element corresponds to one of the bins in the hue-saturation histogram.

On the subimage level, each image is decomposed into 5 subimages, which is illustrated by the sample image in Figure 1. Such an approach was used in [3], and is a step toward identifying the *semcons* [4] appearing in an image. Considering that it is very common to have the major object located in central position in the image, we have one subimage to capture the central region in each image, and the other four subimages cover the upper-left, upper-right, lower-left, and lower-right areas in the image. For each pixel of the resulting subimage, hue and saturation are extracted and each quantized into a 10-bin histogram. Then the two histograms h and s are again combined into one $h \times s$ histogram with 100 bins, which is taken to be the representing feature vector of each image. This is a vector of 100 elements, $\mathbf{V} = [f_1, f_2, f_3, \ldots f_{100}]^T$, where each element again corresponds to one of the bins in the hue-saturation histogram.

Fig. 1. Subimage Decomposition

Since both global and subimage color histograms are formulated as a feature vector, it is very easy to use them as the input for latent semantic indexing.

2.2 Anglogram

In this section, we first provide some background concepts for Delaunay triangulation in computational geometry, and then present the geometric triangulation-based *anglogram* for encoding spatial correlation, which is invariant to translation, scale, and rotation.

Let $P = \{ p_1, p_2, \ldots, p_n \}$ be a set of points in the two-dimensional Euclidean plane, namely the *sites*. Partition the plane by labeling each point in the plane to its nearest site. All those points labeled as p_i form the *Voronoi region* $V(p_i)$. $V(p_i)$ consists of all the points x, which are at least as close to p_i as to any other site:

$$V(p_i) = \{ x: | p_i - x| \leq | p_j - x|, \forall j \neq i \}.$$

Some of the points do not have a unique nearest site, however. The set of all points that have more than one nearest site form the *Voronoi diagram* $V(P)$ for the set of sites.

Construct the *dual* graph G for a Voronoi Diagram $V(P)$ as follows: the nodes of G are the sites of $V(P)$, and two nodes are connected by an arc if their corresponding Voronoi polygons share a Voronoi edge. In 1934, Delaunay proved that when the dual graph is drawn with straight lines, it produces a planar triangulation of the

Voronoi sites *P*, so called the *Delaunay triangulation D(P)*. Each face of *D(P)* is a triangle, so called the *Delaunay triangle*.

The spatial layout of a set of points can be coded through such an *anglogram.* One discretizes and counts the angles produced by the Delaunay triangulation of a set of unique feature points in some context, given the selection criteria of what the bin size will be and of which angles will contribute to the final angle histogram. An important property of our proposed anglogram for encoding spatial correlation is its invariance to translation, scale, and rotation. An O(*max(N, #bins)*) algorithm is necessary to compute the anglogram corresponding to the Delaunay triangulation of a set of *N* points.

The *shape anglogram* approach can be used for image object indexing, while the *color anglogram* can be used as a spatial color representation technique.

In the *shape anglogram* approach, those objects that will be used to index the image are identified, and then a set of high-curvature points along the object boundary are obtained as the feature points. The Delaunay triangulation is performed on these feature points and thus the feature point histogram is computed by discretizing and counting the number of either the two largest angles or the two smallest angles in the Delaunay triangles.

To apply the *color anglogram* approach, color features and their spatial relationship are extracted and then coded into the Delaunay triangulation. Each image is decomposed into a number of non-overlapping blocks. Each individual block is abstracted as a unique feature point labeled with its spatial location and feature values. The feature values in our experiment are dominant or average hue and saturation in the corresponding block. Then, all the normalized feature points form a point feature map for the corresponding image. For each set of feature points labeled with a particular feature value, the Delaunay triangulation is constructed and then the feature point histogram is computed by discretizing and counting the number of either the two largest angles or the two smallest angles in the Delaunay triangles. Finally, the image will be indexed by using the concatenated feature point histogram for each feature value.

3 Latent Semantic Indexing

In this section we describe an approach to automatic information indexing and retrieval, namely, *latent semantic indexing (LSI).* It is introduced to overcome a fundamental problem that plagues existing retrieval techniques that try to match words of queries with words of documents. The problem is that users want to retrieve on the basis of conceptual content, while individual words provide unreliable evidence about the conceptual meaning of a document. There are usually many ways to express a given concept. Therefore, the literal terms used in a user's query may not match those of a relevant document. In addition, most words have multiple meanings and are used in different contexts. Hence, the terms in a user's query may literally match the terms in documents that are not of any interest to the user at all.

In information retrieval these two problems are addressed as *synonymy* and *polysemy*. The concept *synonymy* is used in a very general sense to describe the fact that there are many ways to refer to the same object. Users in different contexts, or with different needs, knowledge, or linguistic habits will describe the same concept using different terms. The prevalence of synonyms tends to decrease the *recall* performance of the retrieval. By *polysemy* we refer to the general fact that most words have more than one distinct meaning. In different contexts or when used by different people the same term takes on varying referential significance. Thus the use of a term in a query may not necessarily mean that a document containing the same term is relevant at all. Polysemy is one factor underlying poor *precision* performance of the retrieval [5].

Latent semantic indexing tries to overcome the deficiencies of term-matching retrieval by treating the unreliability of observed term-document association data as a statistical problem. It is assumed that there exists some underlying latent semantic structure in the data that is partially obscured by the randomness of word choice with respect to retrieval. Statistical techniques are used to estimate this latent semantic structure, and to get rid of the obscuring noise. By semantic structure we mean the correlation structure in which individual words appear in documents; semantic implies only the fact that terms in a document may be taken as referents to the document itself or to its topic.

The latent semantic indexing technique makes use of the singular value decomposition (SVD). We take a large matrix of term-document association data and construct a semantic space wherein terms and documents that are closely associated are placed near to each other. Singular value decomposition allows the arrangement of the space to reflect the major associative patterns in the data, and ignore the smaller, less important influences. As a result, terms that did not actually appear in a document may still end up close to the document, if that is consistent with the major patterns of association in the data. Position in the space then serves as a new kind of semantic indexing. Retrieval proceeds by using the terms in a query to identify a point in the semantic space, and documents in its neighborhood are returned as relevant results to the query.

Latent semantic indexing is based on the fact that the term-document association can be formulated by using the vector space model, in which each document is encoded as a vector, where each vector component reflects the importance of a particular term in representing the semantics of that document. The vectors for all the documents in a database are stored as the columns of a single matrix. Latent semantic indexing is a variant of the vector space model in which a low-rank approximation to the vector space representation of the database is employed. That is, we replace the original matrix by another matrix that is as close as possible to the original matrix but whose column space is only a subspace of the column space of the original matrix. Reducing the rank of the matrix is a means of removing extraneous information or noise from the database it represents. Rank reduction is used in various applications of linear algebra and statistics as well as in image processing, data compression, cryptography, and seismic tomography. According to

[6], latent semantic indexing has achieved average or above average performance in several experiments with the TREC collections.

3.1 The Vector-Space Model

In the vector space model, a vector is used to represent each item or *document* in a collection. Each component of the vector reflects a particular concept, keyword, or term associated with the given document. The value assigned to that component reflects the importance of the term in representing the semantics of the document. Typically, the value is a function of the frequency with which the term occurs in the document or in the document collection as a whole [7].

A database containing a total of d documents described by t terms is represented as a $t \times d$ *term-by-document matrix A*. The d vectors representing the d documents form the columns of the matrix. Thus, the matrix element a_{ij} is the weighted frequency at which term i occurs in document j. The columns of A are called the *document vectors*, and the rows of A are the *term vectors*. The semantic content of the database is contained in the column space of A, meaning that the document vectors span that content. We can exploit geometric relationships between document vectors to model similarity and differences in content. Meanwhile, we can also compare term vectors geometrically in order to identify similarity and differences in term usage.

A variety of schemes are available for weighting the matrix elements. The element a_{ij} of the term-by-document matrix A is often assigned values as $a_{ij} = l_{ij}g_i$. The factor g_i is called the *global weight*, reflecting the overall value of term i as an indexing term for the entire collection. As one example, consider a very common term like *image* within a collection of articles on image retrieval. It is not important to include that term in the description of a document as all of the documents are known to be about image so a small value of the global weight g_i is appropriate. Global weighting schemes range from simple normalization to advanced statistics-based approaches [7]. The factor l_{ij} is a local weight that reflects the importance of term i within document j itself. Local weights range in complexity from simple binary values to functions involving logarithms of term frequencies. The latter functions have a smoothing effect in that high-frequency terms having limited discriminatory value are assigned low weights.

4.2 Singular-Value Decomposition

The singular value decomposition (SVD) is a dimension reduction technique which gives us reduced-rank approximations to both the column space and row space of the vector space model. The SVD also allows us to find a rank-k approximation to a matrix A with minimal change to that matrix for a given value of k [6]. The decomposition is defined as $A = U\ S\ V^T$, where U is the $t \times t$ orthogonal matrix having the left singular vectors of A as its columns, V is the $d \times d$ orthogonal matrix

having the right singular vectors of A as its columns, and S is the $t \times d$ diagonal matrix having the singular values $s_1 \geq s_2 \geq \ldots \geq s_r$ of the matrix A in order along its diagonal, where $r = min(t, d)$. This decomposition exists for any given matrix A [8].

The rank r_A of the matrix A is equal to the number of nonzero singular values. It follows directly from the orthogonal invariance of the *Frobenius* norm that $\| A \|_F$ is defined in terms of those values,

$$\|A\|_F = \|U\Sigma V^T\|_F = \|\Sigma V^T\|_F = \|\Sigma\|_F = \sqrt{\sum_{j=1}^{r_A} s_j^2}$$

The first r_A columns of matrix U are a basis for the column space of matrix A, while the first r_A rows of matrix V^T are a basis for the row space of matrix A. To create a rank-k approximation A_k to the matrix A, where $k \leq r_A$, we can set all but the k largest singular values of A to be zero. A classic theorem about the singular value decomposition states that the distance between the original matrix A and its rank-k approximation is minimized by the approximation A_k. The theorem further shows how the norm of that distance is related to singular values of matrix A. It is described as

$$\|A - A_k\|_F = \min_{rank(X) \leq k} \|A - X\|_F = \sqrt{s_{k+1}^2 + \ldots + s_{r_A}^2}$$

Here $A_k = U_k S_k V_k^T$, where U_k is the $t \times k$ matrix whose columns are the first k columns of matrix U, V_k is the $d \times k$ matrix whose columns are the first k columns of matrix V, and S_k is the $k \times k$ diagonal matrix whose diagonal elements are the k largest singular values of matrix A.

How to choose the rank that provides optimal performance of latent semantic indexing for any given database remains an open question and is normally decided via empirical testing. For very large databases, the number of dimensions used usually ranges between 100 and 300. Normally, it is a choice made for computational feasibility as opposed to accuracy. Using the SVD to find the approximation A_k, however, guarantees that the approximation is the best that can be achieved for any given choice of k.

4.3 Similarity Measure

In the vector space model, a user queries the database to find relevant documents, using the vector space representation of those documents. The query is also a set of terms, with or without weights, represented by using a vector just like the documents. It is likely that many of the terms in the database do not appear in the query, meaning that many of the query vector components are zero. Meanwhile, even though some of the terms in the query and in the documents are common, they may be used to refer to different concepts. Considering the general problems of *synonymy* and *polysemy*, we are trying to reveal the underlying semantic structure of the database and thus improve the query performance by using the latent semantic indexing technique. A query can be issued after the SVD has been performed on the database and an appropriate lower rank approximation has been generated. The

matching process is to find the documents most similar to the query in the use and weighting of terms. In the vector space model, the documents selected are those geometrically closest to the query in the transformed semantic space.

One common measure of similarity is the cosine of the angle between the query and document vectors. If the term-by-document matrix A has columns a_j, $j = 1, 2, \ldots, d$, those d cosines are computed according to the following formula

$$\cos q_j = \frac{a_j^T q}{\|a_j\|_2 \|q\|_2} = \frac{\sum_{i=1}^{t} a_{ij} q_i}{\sqrt{\sum_{i=1}^{t} a_{ij}^2} \sqrt{\sum_{i=1}^{t} q_i^2}}$$

for $j = 1, 2, \ldots, d$, where the Euclidean vector norm $\| x \|_2$ is defined by

$$\|x\|_2 = \sqrt{x^T x} = \sqrt{\sum_{i=1}^{t} x_i^2}$$

for any t-dimensional vector x. Because the query and document vectors are typically sparse, the dot product and norms are generally inexpensive to compute. Furthermore, the document vector norms $\|a_j\|_2$ need to be computed only once for any given term-by-document matrix. Note that multiplying either a_j or q by a constant does not change the cosine value, thus, we may scale the document vectors or the queries by any convenient factor.

With any given document database and user's query, we can always generate the term-by-document matrix and then apply the singular value decomposition to this matrix. We hope to choose a good lower-ranked approximation after the SVD and use this transformed matrix to construct the semantic space of the database. Then, the query process will be to locate those documents geometrically closest to the query vector in the semantic space.

The latent semantic indexing technique has been successfully applied to information retrieval, in which it shows distinctive power of finding the latent correlation between terms and documents. This inspires us to attempt to borrow this technique from traditional information retrieval and apply it to visual information retrieval. We hope to make use of the power of latent semantic indexing to reveal the underlying semantic nature of visual contents, and thus to find the correlation between visual features and semantics of visual documents or objects. We will explore this approach further by correlating low-level feature groups and high-level semantic clusters, hoping to figure out the semantic nature behind those visual features. Some preliminary experiments have been conducted and the results show that integrating latent semantic indexing with content-based retrieval is a promising approach. Details of these experiments are presented in the next section.

4 Finding Latent Correlation between Visual Features and Semantics

Existing management systems for image collections and their users are typically at cross-purposes. While these systems normally retrieve images based on low-level features, users usually have a more abstract notion of what will satisfy them. Using low-level features to correspond to high-level abstractions is one aspect of the *semantic gap* [9] between content-based system organization and the concept-based user. Sometimes, the user has in mind a concept so abstract that he himself doesn't know what he wants until he sees it. At that point, he may want images similar to what he has just seen or can envision. Again, however, the notion of similarity is typically based on high-level abstractions, such as activities taking place in the image or evoked emotions. Standard definitions of similarity using low-level features generally will not produce good results.

In reality, the correspondence between user-based semantic concepts and system-based low-level features is many-to-many. That is, the same semantic concept will usually be associated with different sets of image features. Also, for the same set of image features, different users could easily find dissimilar images relevant to their needs, such as when their relevance depends directly on an evoked emotion.

In this section, we present the results of a series of experiments that seeks to transform low-level features to a higher level of meaning. This study concerns a technique, latent semantic analysis, which has been used for information retrieval for many years. In this environment, this technique determines clusters of co-occurring keywords, sometimes, called *concepts*, so that a query which uses a particular keyword can then retrieve documents perhaps not containing this keyword, but containing other keywords from the same cluster. In this preliminary study, we examine the use of this technique for content-based image retrieval to find the correlation between visual features and semantics.

4.1 The Effects of Latent Semantic Indexing, Normalization, and Weighting for Global and Subimage Color Histograms

In this and the next section, we show the improvement that latent semantic analysis, normalization, and weighting can give to two simple and straightforward image retrieval techniques, both of which use standard color histograms. For our experiments, we use a database of 50 JPEG images, each of size 192×128. This image collection consists of ten semantic categories of five images each. The categories consist of: ancient towers, ancient columns, birds, horses, pyramids, rhinos, sailing scenes, skiing scenes, sphinxes, and sunsets.

Our first approach uses global color histograms. Each image is first converted from the RGB color space to the HSV color space. For each pixel of the resulting image, hue and saturation are extracted and each quantized into a 10-bin histogram. Then the two histograms h and s are combined into one $h \times s$ histogram with 100 bins, which is the representing feature vector of each image. This is a vector of 100

elements, $\mathbf{V} = [f_1, f_2, f_3, \ldots f_{100}]^T$, where each element corresponds to one of the bins in the hue-saturation histogram.

We then generate the feature-image-matrix, $\mathbf{A} = [\mathbf{V_1},\ldots,\mathbf{V_{50}}]$, which is 100×50. Each row corresponds to one of the elements in list of features and each column is the entire feature vector of the corresponding image. This matrix is written into a file so the computation is done only once. The matrix will be retrieved from the file during the query process.

A singular value decomposition is then performed on the feature-image-matrix. The result comprises three matrices, **U, S** and **V**, where $\mathbf{A = USV^T}$. The dimensions of **U** are 100×100, **S** is 100×50, and **V** is 50×50. The rank of matrix **S**, and thus the rank of matrix **A**, in our case is 50. Therefore, the first 50 columns of **U** spans the column space of **A** and all the 50 rows in $\mathbf{V}^T$ spans the row space of **A. S** is a diagonal matrix of which the diagonal elements are the singular values of **A**. To reduce the dimensionality of the transformed latent semantic space, we use a rank-*k* approximation, $\mathbf{A}_k$, of the matrix **A**, for k = 34, which worked better than other values tried. This is defined by $\mathbf{A_k = U_k S_k V_k^T}$. The dimension of $\mathbf{A}_k$ is the same as **A**, 100 by 50. The dimensions of $\mathbf{U}_k$, $\mathbf{S}_k$, and $\mathbf{V}_k$ are 100×34, 34×34, and 50×34, respectively.

The query process in this approach is to compute the distance between the transformed feature vector of the query image, **q**, and that of each of the 50 images in the database, **d**. This distance is defined as $dist(\mathbf{q}, \mathbf{d}) = \mathbf{q^T d} / \|\mathbf{q}\| \|\mathbf{d}\|$, where $\|\mathbf{q}\|$ and $\|\mathbf{d}\|$ are the norms of those vectors. The computation of $\|\mathbf{d}\|$ for each of the 50 images is done only once and then written into a file. Using each image as a query, in turn, we find the average sum of the positions of all of the five correct answers. Note that in the best case, where the five correct matches occupy the first five positions, this average sum would be 15, whereas in the worst case, where the five correct matches occupy the last five positions, this average sum would be 240. A measure that we use of how good a particular method is defined as,

$$measure-of-goodness = \frac{48 - average-sum/5}{45}.$$

We note that in the best case, this measure is equal to 1, whereas in the worst case, it is equal to 0.

This approach was then compared to one without using latent semantic analysis. We also wanted to see whether the standard techniques of normalization and term weighting from text retrieval would work in this environment.

The following *normalization* process will assign equal emphasis to each component of the feature vector. Different components within the vector may be of totally different physical quantities. Therefore, their magnitudes may vary drastically and thus bias the similarity measurement significantly. One component may overshadow the others just because its magnitude is relatively too large. For the feature image matrix $\mathbf{A}=[\mathbf{V_1},\mathbf{V_2}, \ldots, \mathbf{V_{50}}]$, we have $\mathbf{A_{i,j}}$ which is the i^{th} component in vector $\mathbf{V_j}$. Assuming a Gaussian distribution, we can obtain the mean m_i and standard deviation s_i for the i^{th} component of the feature vector across the whole

image database. Then we normalize the original feature image matrix into the range of [-1,1] as follows,

$$A_{i,j} = \frac{A_{i,j} - m_i}{s_i}.$$

It can easily be shown that the probability of an entry falling into the range of [-1, 1] is 68%. In practice, we map all the entries into the range of [-1, 1] by forcing the out-of-range values to be either –1 or 1. We then shift the entries into the range of [0, 1] by using the following formula

$$A_{i,j} = \frac{A_{i,j} + 1}{2}.$$

After this normalization process, each component of the feature image matrix is a value between 0 and 1, and thus will not bias the importance of any component in the computation of similarity.

One of the common and effective methods for improving full-text retrieval performance is to apply different weights to different components [7]. We apply these techniques to our image environment. The raw frequency in each component of the feature image matrix, with or without normalization, can be weighted in a variety of ways. Both global weight and local weight are considered in our approach. A *global weight* indicates the overall importance of that component in the feature vector across the whole image collection. Therefore, the same global weighting is applied to an entire row of the matrix. A *local weight* is applied to each element indicating the relevant importance of the component with its vector. The value for any component $\mathbf{A_{i,j}}$ is thus *L(i, j)G(i),* where *L(i, j)* is the local weighting for feature component *i* in image *j*, and *G(i)* is the global weighting for that component.

Common local weighting techniques include term frequency, binary, and log of term frequency, whereas common global weighting methods include *Normal, GfIdf, Idf,* and *Entropy*. Based on previous research it has been found that log of term frequency helps to dampen effects of large differences in frequency and thus has the best performance as a local weight, whereas Entropy is the appropriate method for global weighting [7].

The entropy method is defined by having a component global weight of,

$$1 + \sum_j \frac{p_{ij} \log(p_{ij})}{\log(number\_of\_images)}$$

where pij = tf_{ij} / gf_i is the probability of that component, tf_{ij} is the raw frequency of component $\mathbf{A_{i,j}}$, and gf_i is the global frequency, i.e., the total number of times that component *i* occurs in the whole collection.

The global weights give less emphasis to those components that occur frequently or in many images. Theoretically, the entropy method is the most sophisticated weighting scheme and it takes the distribution property of feature components over the image collection into account.

We conducted similar experiments for these four cases:

1. Global color histograms, no normalization, no term weighting, no latent-semantic indexing (raw data)

2. Global color histograms, normalized and term-weighted, no latent semantic indexing
3. Global color histograms, no normalization, no term-weighting, with latent semantic indexing
4. Global color histograms, normalized and term-weighted, with latent semantic indexing

The results are shown in Table 1, where each table entry is a measure-of-goodness of the corresponding technique. We note that the improvements under LSI don't seem very large. This is an artifact of the small size and nature of our database and the fact that any of the techniques mentioned work well. It is, however, an indication that LSI is a technique worthy of further study in this environment.

	Global Color Histogram	**Color Anglogram**
Raw Data	0.9257	0.9508
Raw Data with LSI	0.9377	0.9556
Normalized and Weighted Data	0.9419	0.9272
Normalized and Weighted Data with LSI	0.9446	0.9284

Table 1. Results for Global Color Histogram and Color Anglogram Representations

Thus, for the global histogram approach, using normalized and weighted data or using latent semantic indexing with the raw data improves performance, while using both techniques is even better.

Our next approach uses sub-image matching in conjunction with color histograms. Each image is first converted from the RGB color space to the HSV color space. Each image is decomposed into 5 overlapping subimages, as shown in Figure 1. For the 50 images in our case, 250 subimages will be used in the following feature extraction process. For each pixel of the resulting image, hue and saturation are extracted and each quantized into a 10-bin histogram. Then the two histograms h and s are combined into one $h \times s$ histogram with 100 bins, which is the representing feature vector of each image. This is a vector of 100 elements, $\mathbf{V} = [f_1, f_2, f_3, \ldots f_{100}]^T$, where each element corresponds to one of the bins in the hue-saturation histogram.

We then generate the feature-subimage-matrix, $\mathbf{A} = [\mathbf{V_1},\ldots,\mathbf{V_{250}}]$, which is 100×250. Each row corresponds to one of the elements in the feature vector and each column is the whole feature vector of the corresponding subimage. This matrix is written into a file so the computation is done only once. The matrix will be retrieved from the file during the query process.

A singular value decomposition is then performed on the feature-subimage-matrix. The result comprises three matrices, **U, S** and **V**, where $\mathbf{A} = \mathbf{USV^T}$. The dimensions of **U** are 100×100, **S** is 100×250, and **V** is 250×250. The rank of matrix **S**, and thus the rank of matrix **A**, in our case is 100. Therefore, the first 100 columns of **U** spans the column space of **A** and all the 100 rows in $\mathbf{V^T}$ spans the row

space of **A. S** is a diagonal matrix of which the diagonal elements are the singular values of **A**. To reduce the dimensionality of the transformed latent semantic space, we use a rank-*k* approximation, $\mathbf{A}_k$, of the matrix **A**, for k = 55. This is defined by $\mathbf{A_k} = \mathbf{U_k S_k V_k}^T$. The dimension of $\mathbf{A}_k$ is the same as **A**, 100 by 50. The dimensions of $\mathbf{U}_k$, $\mathbf{S}_k$, and $\mathbf{V}_k$ are 100×55, 55×55, and 250×55, respectively.

The first step of the query process in this approach is to compute the distance between the transformed feature vector of each subimage of the query image, **q**, and that of each of the 250 images in the database, **d**. This distance is defined as $dist(\mathbf{q}, \mathbf{d}) = \mathbf{q}^T\mathbf{d} \;/\; \|\mathbf{q}\| \; \|\mathbf{d}\|$, where $\|\mathbf{q}\|$ and $\|\mathbf{d}\|$ are the norms of those vectors. The computation of $\|\mathbf{d}\|$ for each of the 250 subimages is done only once and then written into a file.

With respect to the query image and each of the 50 database images, we now have the distances between each pair of subimages by the previous step. These distance values $dist(\mathbf{q}_i, \mathbf{d}_i)$ are then combined into one distance value between these two images in an approach similar to the computation of Euclidean distance. Given a query image **q**, with corresponding subimages $q_1, \ldots, q_5$, and a candidate database image **d**, with corresponding subimages $d_1, \ldots, d_5$, we define,

$$dist(q, d) = \frac{1}{5}\sqrt{\sum_{i=1}^{5}\left[dist(q_i, d_i)\right]^2}$$

This approach was again compared to one without using latent semantic analysis. Each image is decomposed into five subimages which are then represented by their hue-saturation histograms ***V***. Then the cosine measure between corresponding subimages is computed and used as the similarity metric. We thus have the distance between the query image and each of the 50 database images. These similarity values are then combined into one similarity measure between these two images. Given a query image **q** and a candidate image **d** in the database, we define,

$$dist(q, d) = \frac{1}{5}\sum_{i=1}^{5} sim(q_i, d_i)$$

Using each image as a query, we again find the average sum of the positions of all of the five correct answers. Now, without using latent semantic analysis, using the measure previously introduced, the result is 0.9452, while the use of latent semantic analysis brings this measure to 0.9502.

We also did a similar experiment where $dist(q, d)$ weighted the center subimage twice as much as the peripheral subimages. Using the same measure, the results of these experiments are 0.9475 for the experiment without using latent semantic analysis and 0.9505 for that using latent semantic analysis. Therefore, latent semantic indexing does improve the retrieval performance for both global and subimage color histogram based retrieval. Comparison of global and subimage results shows that subimage provides better performance than global color histogram either with or without latent semantic indexing.

4.2 The Effects of Latent Semantic Indexing, Normalization, and Weighting for Color Anglograms

Our next approach performs similar experiments utilizing our previously formulated approach of color anglograms [10]. This is a novel spatial color indexing scheme based on the point feature map obtained by dividing an image evenly into a number of *M*N* non-overlapping blocks with each individual block abstracted as a unique feature point labeled with its spatial location, dominant hue, and dominant saturation.

For our experiments, we divide the images into 8*8 blocks, have 10 quantized hue values and 10 quantized saturation values, count the two largest angles for each Delauney triangle, and have an anglogram bin of 5°. Our vector representation of an image thus has 720 elements: 36 hue bins for each of 10 hue ranges and 36 saturation bins for each of 10 saturation ranges. We use the same approach to querying as in the previous section.

We conducted similar experiments for these four cases:

1. Color anglograms, no normalization, no term weighting, no latent-semantic indexing (raw data)
2. Color anglograms, normalized and term-weighted, no latent semantic indexing
3. Color anglograms, no normalization, no term-weighting, with latent semantic indexing
4. Color anglograms, normalized and term-weighted, with latent semantic indexing

with the results shown in Table 1.

From these results, one notices that our anglogram method is better than the standard global color histogram, which is consistent with our previous results [10,11]. One also notices that latent semantic indexing improves the performance of this method. However, it seems that normalization and weighting has a negative impact on query performance. We more thoroughly examined the impact of these techniques and derived the data shown in Table 2.

	Color Anglogram
Raw Data with LSI	0.9556
Normalized Data with LSI	0.9476
Weighted Data with LSI	0.9529
Normalized and Weighted Data with LSI	0.9284

Table 2. More Detailed Results for Color Anglogram Representation

The impact of normalization is worse than that of weighting. Normalization is a compacting process which transforms the original feature image matrix (the anglogram elements) to the range [0, 1]. Now, the feature image matrix in this case is a sparse matrix with many 0's, some small integers, and a relatively small number of large integers. We believe that these large integers represent the discriminatory power of the anglogram and that the compacting effect of normalization weakens their significance. Local log-weighting also has a compacting effect. Since both the local and global weighting factors lie between 0 and 1, the transformed matrix

always has smaller values than the original one, even though no normalization is applied. Thus, normalization and weighting don't help improve the performance, but actually makes it worse.

4.3 Utilizing Image Annotations

We conducted various experiments to determine whether image annotations could improve the query results of our various techniques. The results indicate that they can.

For both the global color histogram and color anglogram representation, we appended an extra 15 elements to each of these vectors (called *category bits*) to accommodate the following 15 keywords associated with these images: *sky, sun, land, water, boat, grass, horse, rhino, bird, human, pyramid, column, tower, sphinx, and snow*. Thus, the feature vector for the global histogram representation now has 115 elements (100 visual elements and 15 textual elements), while the feature vector for the color anglogram representation now has 735 elements (720 visual elements and 15 textual elements). Each image is annotated with appropriate keywords and the area coverage of each of these keywords. For instance, one of the images is annotated with sky(0.55), sun(0.15), and water(0.30). This is a very simple model for incorporating annotation keywords. One of the strengths of the LSA technique is that it is a vector-based method that helps us to integrate easily different features into one feature vector and to treat them just as similar components. Hence, ostensibly, we can apply the normalization and weighting mechanisms introduced in the previous sections to the expanded feature image matrix without any concern.

For the global color histogram representation, we start with an image feature matrix of size 115×50. Then, using the SVD, we again compute the rank 34 approximation to this matrix, which is also 115×50. For each query image, we fill bits 101 through 115 with 0's. We also fill the last 15 rows of the transformed image feature matrix with all 0's. Thus, for the querying, *we do not use any annotation information*. We also note, that as before, we apply normalization and weighting, as this improves the results, which are shown in Table 3. The first two results are from Table 1, while the last result shows how our technique of incorporating annotation information improves the querying process.

	Global Color Histogram
Normalized and Weighted Data	0.9419
Normalized and Weighted Data with LSI	0.9446
Normalized and Weighted Data with LSI and Annotation Information	0.9465

Table 3. Global Color Histograms with Annotation Information

For the color anglogram representation, we start with an image feature matrix of size 735×50. Then, using the SVD, we again compute the rank 34 approximation to this matrix, which is also 735×50. For each query image, we fill bits 721 through

735 with 0's. We also fill the last 15 rows of the transformed image feature matrix with all 0's. Thus, for the querying, *we do not use any annotation information*. We also note that as before, we do not apply normalization and weighting, as this improves the results, which are shown in Table 4. The first two results are from Table 1, while the last result shows how our technique of incorporating annotation information improves the querying process.

	Color Anglogram
Raw Data	0.9508
Raw Data with LSI	0.9556
Raw Data with LSI and Annotation Information	0.9590

Table 4. Global Color Histograms with Annotation Information

Note that annotations improve the query process for color anglograms, even though we do not normalize the various vector components, nor weight them. This is quite surprising, given that the feature image vector consists of 720 visual elements, which are relatively large integers, and only 15 annotation elements, which are in the range [0,1].

5 Conclusion and Future Work

In this chapter we proposed image retrieval schemes that incorporate multiple visual feature extraction represented by color histograms and color anglograms. Features are extracted on both whole image level and subimage level to better capture salient object descriptions. To negotiate the gap between low-level visual features and high-level concepts, latent semantic indexing is applied and integrated with these content-based retrieval techniques in a vector space model. Correlation between visual features and semantics are explored. Annotations are also fused into the feature vectors to improve the efficiency and effectiveness of the retrieval process.

The results presented in the previous section are quite interesting and are certainly worthy of further study. Our hope is that latent semantic analysis will find that different image features co-occur with similar annotation keywords, and consequently lead to improved techniques of semantic image retrieval. We are currently experimenting with the integration of shape anglograms, color anglograms, and structural features with latent semantic indexing and developing a unified framework to accommodate multiple features and their representation. We will further test and benchmark this integrated image retrieval framework over various large image databases, along with tuning the latent semantic indexing scheme to achieve optimal performance with highly reduced dimensionality. We will further our study of image semantics and incorporation of textual annotations and explore the correlation between visual feature groups and semantic clusters. We also consider applying various clustering techniques and use the cluster identifier in place of annotation information. Analyzing the patterns of user interaction, either in the query process or in the browsing process, is another interesting research topic.

Making use of relevance feedback to infer user preference should also be incorporated to elevate the retrieval performance. Finally, considering that the image archives on the internet are normally associated with other sources of information such as captions, titles, labels, and surrounding texts, we also propose to extend the application of the latent semantic indexing technique to analyze the structure of different types of visual and hypermedia documents.

References

1. A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, Content-Based Image Retrieval at the End of the Early Years, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, 2000.
2. R. Zhao and W. I. Grosky, From Features to Semantics: Some Preliminary Results, *Proceedings of the IEEE International Conference on Multimedia & Expo*, New York, New York, 2000.
3. M. Stricker and A. Dimai. Color Indexing with Weak Spatial Constraints. *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, Vol. 2670, 1996, pp. 29-40.
4. W. I. Grosky, F. Fotouhi, and Z. Jiang. Using Metadata for the Intelligent Browsing of Structured Media Objects, *Managing Multimedia Data: Using Metadata to Integrate and Apply Digital Data*, A. Sheth and W. Klas (Eds.), McGraw Hill Publishing Company, New York, 1998, pp. 67-92.
5. S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, Volume 41, Number 6 (1990), pp. 391-407.
6. M. Berry, Z. Drmac, and E. Jessup, Matrices, Vector Spaces, and Information Retrieval, *SIAM Review*, Vol. 41, No. 2, 1999, pp. 335-362.
7. S. Dumais, Improving the Retrieval of Information from External Sources, *Behavior Research Methods, Instruments, and Computers*, Vol. 23, Number 2 (1991), pp. 229-236.
8. G. H. Golub and C. Van Loan, *Matrix Computation*, Johns Hopkins Univ. Press, Baltimore, MD, 1996.
9. V. N. Gudivada and V. Raghavan. Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity. *ACM Transactions on Information Systems*, Vol. 13, No. 1 (April 1995), pp. 115-144.
10. Y. Tao and W. I. Grosky, Spatial Color Indexing Using Rotation, Translation, and Scale Invariant Anglograms, *Multimedia Tools and Applications*, To Appear.
11. Y. Tao and W. I. Grosky, Object-Based Image Retrieval Using Point Feature Maps, *Proceedings of The 8th IFIP 2.6 Working Conference on Database Semantics (DS8)*, Rotozur, New Zealand, January 5-8, 1999.