Mapping low-level image features to semantic concepts

Daniela Stan and Ishwar K. Sethi

Intelligent Information Engineering Laboratory
Department of Computer Science and Engineering
Oakland University, Rochester, Michigan 48309-4478, USA

ABSTRACT

Humans tend to use high-level semantic concepts when querying and browsing multimedia databases; there is thus, a need for systems that extract these concepts and make available annotations for the multimedia data. The system presented in this paper satisfies this need by automatically generating semantic concepts for images from their low-level visual features. The proposed system is built in two stages. First, an adaptation of k-means clustering using a non-Euclidean similarity metric is applied to discover the natural patterns of the data in the low-level feature space; the cluster prototype is designed to summarize the cluster in a manner that is suited for quick human comprehension of its components. Second, statistics measuring the variation within each cluster are used to derive a set of mappings between the most significant low-level features and the most frequent keywords of the corresponding cluster. The set of the derived rules could be used further to capture the semantic content and index new untagged images added to the image database. The attachment of semantic concepts to images will also give the system the advantage of handling queries expressed in terms of keywords and thus, it reduces the semantic gap between the user's conceptualization of a query and the query that is actually specified to the system. While the suggested scheme works with any kind of low-level features, our implementation and description of the system is centered on the use of image color information. Experiments using a 21 00 image database are presented to show the efficacy of the proposed system.

Keywords: semantic features, content-based image retrieval, similarity metrics, K-means clustering

1. INTRODUCTION

In today's multimedia rich society, searching efficiently through digital libraries containing large number of digital images and video sequences has become very crucial. Every day, many people use the Internet for searching and browsing through different multimedia databases. To make such searching practical and successful, effective image indexing and searching techniques based on both image's semantics content (keywords) and compositional aspects (color, shape and texture) will be necessary. There are several approaches to retrieve images by the exclusive use of keywords or primitive image features.

Keyword indexing techniques can be used to capture an image's semantic content, describing objects clearly identifiable by linguistic cues. These techniques assign keywords or classification codes to each image when it is first added to the collection and use these descriptors as retrieval keys at search time. These kinds of techniques are often encountered in both newspaper and art libraries. Their advantages consist of high expressive power and the ability to describe image content from the primitive level (low-level features) to the abstract level (high-level features); the high level features involve a significant amount of reasoning about the meaning and purpose of the objects or scenes depicted (such as subjective emotions associated with an image). One of the drawbacks of the current manual indexing techniques is the time of assigning the keywords. When the indexing time for every image takes several minutes, the indexing for a considerable large collection of images is an intensive and time consuming task. Beside the amount of effort that is needed in order to complete such a job, manual indexing has another drawback that cannot be solved with either time or labor. This drawback comes from the fact that the same picture can have different meanings for different people or even for the same person at different times [1]. On the other hand, there are images (such as trademarks) that cannot be described by linguistic cues. Therefore, more effectively indexing techniques are necessary.

Methods that permit image searching based on features automatically extracted from the images themselves are referred as content-based image retrieval (CBIR) techniques[2]. The CBIR systems extract and compare primitive features (such as color, texture and shape) from stored and query images. Then the most similar stored images with the query image, in terms of feature values, are displayed in a ranked order on the screen. Color retrieval yields the best results, in that the computer results of color similarity are similar to those derived by a human visual system[3]. The retrieval becomes more efficient when

the spatial arrangement and coupling of colors over the image are taken into account or when one more low-level feature, such as texture or shape, is added to the system.

One drawback of the current content-based image retrieval systems is that it can only use low-level features. To find a photograph of a certain object to exemplify a newspaper article, the CBIR approach is not effective because the compositional aspect is not significant for the retrieval process and keyword indexing becomes much more effective. To overcome the limitations of these two main approaches, several researchers have attempted to build systems that combine keywords and low-level image features. Some of these works, which try to extract semantic concepts from primitive features, use issues as user feedback[4], color arrangements[5], and semantic visual template concepts[6].

The goal of this paper is to provide a CBIR system that is capable of automatically generating associations between the low-level and semantic-level feature representations of an image database. The proposed system uses clustering as a learning tool. The advantages of using clustering are its unsupervised learning ability and the potential of supporting arbitrary similarity measures. Since our implementation and description of the system is centered on the use of image color information, the measure is a non-Euclidean similarity metric that takes into account the non-linear nature of the color space. The cluster prototypes are designed to summarize the clusters in a manner that is suited for quick human comprehension of its components. They will also inform the user about the approximate regions in which clusters are found in the low-level feature space. Each cluster region is further represented in terms of the cluster prototype and standard deviation in a reduced dimensional cluster feature space; the dimensional reduction is obtained by ordering the low-level features in increasing order of their variances. In the semantic space, the cluster region is represented in terms of the most frequent keyword that characterizes the images of the corresponding cluster. By mapping the cluster region representations from the color feature space to their semantic representations, the system finds the associations between the low-level features and the semantic concepts. The associations are expressed as IF – THEN rules and could be used further to capture the semantic content and index new untagged images being added to the image database. The suggested system is also a powerful tool in reducing the semantic gap between the user's conceptualization of a query and the query that is actually specified to the system. Users who are not well versed with the image domain characteristics could specify a query in terms of keywords and thus, avoid the difficulties of specifying queries with features that are often too primitive.

Figure.1 shows the main steps of the proposed system.

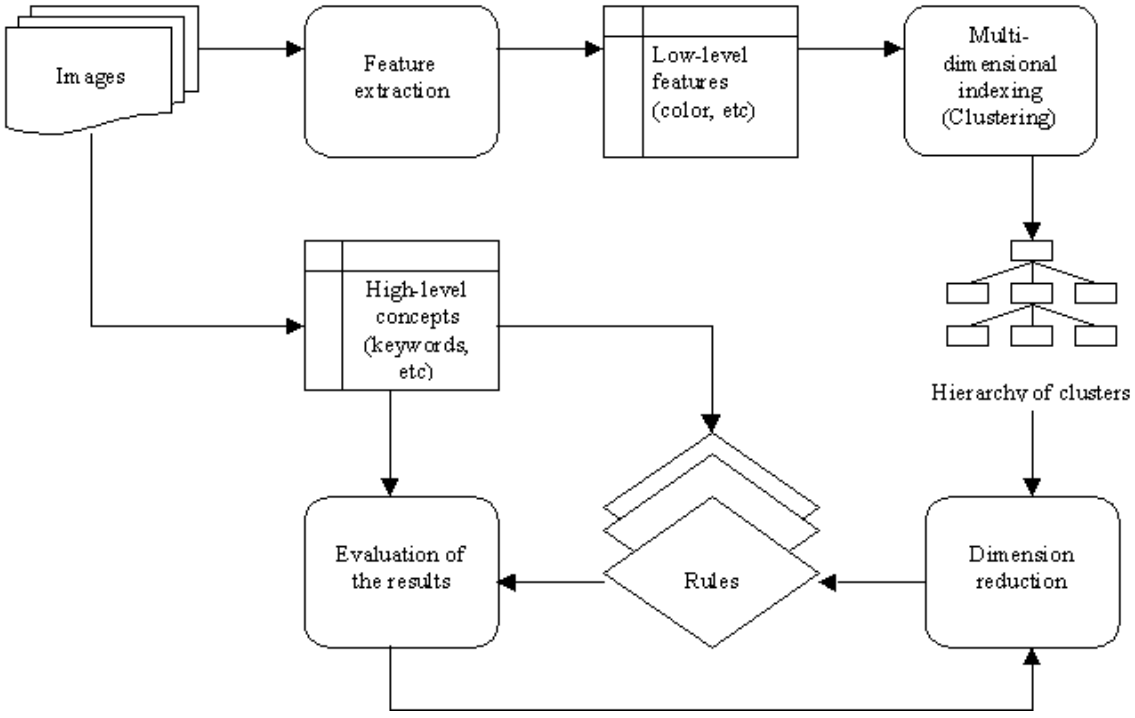

Figure 1.The diagram of the suggested system.

The remainder of the paper is organized as follows. Section 2 describes the low-level and high-level features representations of the image database used in the proposed scheme. Section 3 explains how the image database is organized in related image groups by applying a variation of k-means clustering. Section 4 presents how the mappings between the primitive features and high-level ones are generated and expressed in terms of rules. Section 5 considers the effectiveness of the semantic indexing and retrieval process using the derived rules. These considerations are expounded with experiments on a database of 2100 images. Finally, we conclude with some final comments and a note on future work.

2. IMAGE FEATURE REPRESENTATION

Since our goal is to capture high-level features from low-level image features, we describe in this section how the vector representations for both types of features are obtained.

2.1. Color as low level image feature

While the proposed procedure works with any type of low-level feature representation of images, we describe our system using color information. In this paper, we use the Color-WISE representation[7] for image retrieval in which the representation is guided primarily on three factors. First, the representation must be closely related to human visual perception since a user determines whether a retrieval operation in response to an example query is successful or not. Color-WISE uses the HSV (hue, saturation, value) color coordinate system that correlates well with human color perception and is commonly used by artists to represent color information present in images. The hue component of a color refers to its average spectral wavelength and the saturation component determines the amount of purity in the color perceived. Second, the representation must encode the spatial distribution of color in an image. Because of this consideration, Color-WISE system relies on a fixed partitioning scheme. This is in contrast with several proposals in the literature[8] suggesting color-based segmentation to characterize the spatial distribution of color information. Although the color-based segmentation approach provides a more flexible representation and hence more powerful queries, we believe that these advantages are outweighed by the simplicity of the fixed partitioning approach. In the fixed partitioning scheme, each image is divided into $M \times N$ overlapping blocks as shown in Figure 2.

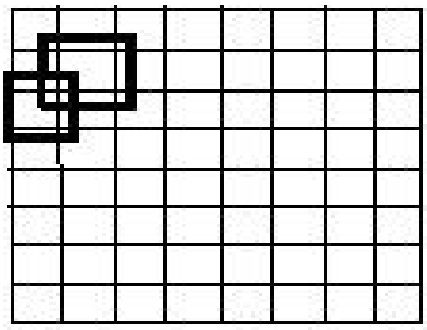

Figure 2.The fixed partitioning scheme with overlapping blocks.

The overlapping blocks allow a certain amount of 'fuzzy-ness' to be incorporated in the spatial distribution of color information, which helps in obtaining a better performance. To provide for partial-image queries, a masking bit is associated to each block. The default value for this bit is one, however during partial-image queries, some of the mask bits are set to zero. Three separate local histograms (hue, saturation and value) for each block are computed. The third factor considered by the Color-WISE system is that fact that the representation should be as compact as possible to minimize storage and computation efforts. To obtain a compact representation, Color-Wise system extracts from each local histogram the location of its area-peak. Placing a fixed-sized window on the histogram at every possible location, the histogram area falling within the window is calculated. The location of the window yielding the highest area determines the histogram area-peak. This value represents the corresponding histogram. Thus, a more compact representation is obtained and each image is reduced to $3 \times M \times N$ numbers (3 represents the number of histograms).

2.2. **Keywords as high level image features**

Keywords are those features that are used to describe the high-level domain concepts[9]. The definition of semantic attributes involves some subjectivity, imprecision or uncertainty. Subjectivity arises due to different viewpoints of the users. Imprecision arises from the difficulty in measuring or specifying some features. In spite of these problems, the use of semantic concepts provides high expressive power and ease of use. Therefore, there is a need for a system that learns semantic concepts and deals with queries containing them automatically.

Two types of scenarios could be used in order to derive semantic concepts that are of interest to users and which can be learned by efficient pattern recognition techniques. The first scenario involves few subjects who are asked to identify and assign meaningful semantic concepts to images. No explicit criterion is given for judging the similarity. The subjects have unlimited time to complete the task and they can create any number of concepts with any number of images per concept. The only restriction is that the subjects should perform the manual indexing independently of each other. At the end of the task, the keywords with no meaning or being synonyms are eliminated and a majority vote scheme is used to decide on the final semantic encoding of each image. The majority vote scheme consists of encoding an image with the keywords that are assigned by the majority of the subjects. The second scenario is related to the idea of getting a more general view about the image database. The image database is posted on the web and users visiting the site are asked to give their own conceptual interpretations for preferred images. The semantic concepts will be stored, preprocessed offline, and used in the mapping process.

3. MULTIDIMENSIONAL INDEXING

We use a hierarchy of clusters to build an effective indexing module that solves both high dimensionality and non-Euclidean nature of the used feature color space. At every level of the hierarchy, the variation of k-means clustering[10] uses a non-Euclidean similarity metric and the cluster prototype is designed to summarize the cluster in a manner that is suited for quick human comprehension of its components. The resultant clusters are further divided into other disjoint sub-clusters performing organization of information at several levels, going for finer and finer distinctions. The adaptation of k-means algorithm is required since the color triplets (hue, saturation, and value) derived from RGB space by non-linear transformation, are not evenly distributed in the HSV space; the representative of a cluster calculated as a centroid also does not make much sense in such a space. Instead of using the Euclidean distance, we need to define a measure that is closer to the human perception in the sense that the distance between two color triplets is a better approximation to the difference perceived by human. We present below the used similarity metric that takes into account both the perceptual similarity between the different histograms bins and the fact that human perception is more sensitive to changes in hue values; we also present how the cluster representatives are calculated and what is the splitting criterion.

3.1 Color similarity metric

Since our retrieval system is designed to retrieve the most similar images with a query image, the proximity index will be defined with respect to similarity. The more two images resemble each other, the larger a similarity index will be.

Different similarity measures have been suggested in the literature to compare images[3,11]. We are using in our clustering algorithm the similarity measure that, besides the perceptual similarity between different bins of a color histogram, recognizes the fact that human perception is more sensitive to changes in hue values[7,12]. It also recognizes that human perception is not proportionally sensitive to changes in hue value.

Let q_i and t_i represent the block number i in a query Q and an image T, respectively. Let $(h_{q_i}, s_{q_i}, v_{q_i})$ and $(h_{t_i}, s_{t_i}, v_{t_i})$ represent the dominant hue-saturation pair of the selected block in the query image and in the image T, respectively. The block similarity is defined by the following relationship:

$$S(q_i, t_i) = \frac{1}{1 + a * D_h(h_{q_i}, h_{t_i}) + b * D_s(s_{q_i}, s_{t_i}) + c * D_v(v_{q_i}, v_{t_i})} \quad (1)$$

Here D_h, D_s and D_v represent the functions that measure similarity in hue, saturation and value. The constants a, b and c define the relative importance of hue, saturation and value in similarity components. Since human perception is more sensitive to hue, a higher value is assigned to a than to b. The following function was used to calculate D_h:

$$D_h\left(h_{q_i}, h_{t_i}\right) = \frac{1 - \cos^k\left(\left\|h_{q_i} - h_{t_i}\right\| * \frac{2\pi}{256}\right)}{2} \tag{2}$$

The function D_h explicitly takes into account the fact that hue is measured as an angle. Through empirical evaluations, a value of k equal to two provides a good non-linearity in the similarity measure to approximate the subjective judgment of the hue similarity.

The saturation similarity is calculated by:

$$D_s\left(s_{q_i}, s_{t_i}\right) = \frac{\left\|s_{q_i} - s_{t_i}\right\|}{256} \tag{3}$$

The value similarity is calculated by using the same formula as for saturation similarity. Using the similarities between the corresponding blocks from the query Q and image T, the similarity between a query and an image is calculated by the following expression:

$$S(Q,T) = \frac{\sum_{i=1}^{M \times N} m_i S\left(q_i, t_i\right)}{\sum_{i=1}^{M \times N} m_i} \tag{4}$$

The quantity m_i in the above expression represents the masking bit for block i and $M \times N$ stands for the number of blocks.

3.2. Cluster prototypes

The cluster prototypes are designed to summarize the clusters in a manner that is suited for quick human comprehension of its components. They will inform the user about the approximate region in which clusters and their descendants are found.

We define the cluster prototype to be the most similar image to the other images from the corresponding cluster; in another words, the cluster representative is the *clustroid* point in the feature space, i.e., the point in the cluster that maximizes the sum of the squares of the similarity values to the other points of the cluster. If C is a cluster, its clustroid M is expressed as:

$$M = \arg\left(\max_{I \in C} \sum_{J \in C} S^2(I,J)\right) \tag{5}$$

Here I and J stand for any two images from the cluster C and $S(I,J)$ is their similarity value. We use arg to denote that the clustroid is the argument/image for which the maximum of the sums is obtained.

3.3. Splitting criterion

To build a partition for a specified number of clusters K, a splitting criterion is necessary to be defined. Since the hierarchy aims to support similarity searches, we would like nearby feature vectors to be collected in the same or nearby nodes. Thus, the splitting criterion in our algorithm will try to find an optimal partition that is defined as one that maximizes the criterion sum-of-squared-error function:

$$J_e(K) = \sum_{k=1}^{K} w_k \sum_{I \in C_k} S^2(I, M_k), \text{where } \mathrm{w_k} = \frac{1}{n_k} \tag{6}$$

M_k and I stand for the clustroid and any image from cluster C_k, respectively; $S^2(I, M_k)$ represents the squared of the similarity value between I and M_k, and n_k represents the number of elements of cluster C_k.

The reason of maximizing the criterion function comes from the fact that the proximity index measures the similarity; that is, the larger a similarity index value is, the more two images resemble one another.

Once the partition is obtained, in order to validate the clusters, i.e. whether or not the samples form one more cluster, several steps are involved. First, we define the null hypothesis and the alternative hypothesis as follows: H_0: there are exactly K clusters for the n samples, and H_A: the samples form one more cluster. According to the Neyman-Pearson paradigm[13], a decision as to whether or not to reject H_0 in favor of H_A is made based on a statistics $T(n)$. The statistic is nothing else than the cluster validity index that is sensitive to the structure in the data:

$$T(n) = \frac{J_e(K)}{J_e(K+1)} \tag{7}$$

To obtain an approximate critical value for the statistic, that is the index is large enough to be 'unusual', we use a threshold that takes into account that, for large n, $J_e(K)$ and $J_e(K+1)$ follow a normal distribution. Following these considerations, we consider the threshold τ defined[14] as:

$$\tau = 1 - \frac{2}{\pi * d} - \alpha * \sqrt{\left[\frac{2 * \left(1 - \frac{8}{\pi^2 * d} \right)}{n * d} \right]} \tag{8}$$

The rejection region for the null hypothesis at the p-percent significance level is:

$$T(n) < \tau \tag{9}$$

The parameter α in (8) is determined from the probability p that the null hypothesis H_0 is rejected when it is true and d is the sample size. The last inequality provides us with a test for deciding whether the splitting of a cluster is justified.

4. MAPPING BETWEEN PRIMITIVE FEATURES AND HIGH-LEVEL CONCEPTS

Users who are not well versed with the image domain characteristics might be more comfortable in working with a CBIR system that allows users to specify a query in terms of keywords, thus eliminating the usually intimidation in dealing with very primitive features. However, the use of semantic features in a query makes the system retrieval to deal with subjectivity, imprecision or uncertainty[9]. The system that we are proposing in this paper is meant to reduce the semantic gap between the user's conceptualization of a query and the query that is actually specified. The system synthesizes the semantic features through a set of mappings on low-level features. The mapping function is based on two things: domain semantics and statistical properties of low-level features. It then assigns them into mappings in the form of IF-THEN rules. Subjectivity and uncertainty in some semantic concepts are solved through user interaction at the query processing time. When a user specifies the query in terms of keywords, the system retrieves and displays those images in the database that are semantically similar with the query. Furthermore, the user marks and clicks on the images that are more relevant to his conceptual view of the expressed query. The system continues the query processing time in the low-level feature space by retrieving now the most similar images with the relevant image.

Let us define the mapping function. Instead of finding mappings for individual images, the goal is to map clusters on the same level of the hierarchy of clusters into their optimal textual characterization. We define the optimal textual characterization of a cluster to be the keyword that is more frequent in that cluster with respect with the other keywords that characterize the images from that cluster. For ranking keywords, we define the following measure:

$$F_k(keyword) = \frac{f_k(keyword)}{\sum_{keyword_i \in T_k} f_k(keyword_i)} \quad , \tag{10}$$

$f_k(keyword)$ denotes the number of times a keyword appears in cluster C_k and T_k is the set of keywords that characterizes C_k. $F_k(keyword)$ is a number between 0 and 1, and the effect of this normalization is to disregard the sizes of the clusters. $F_k(keyword)$ measures the relative importance of a keyword compared to the other keywords occurring in that cluster.

The statistics used to define a cluster C_k in the low-level dimensional space are its *clustroid*, which is defined by formula (5) and its *radius* in each dimension that is analog to the standard deviation in the Euclidean space:

$$radius_{i,k} = \sqrt{w_k \sum_{I \in C_k} \left(I_i - M_{i,k}\right)^2}\ ,\ i = 1 \ldots M \times N \tag{11}$$

I_i and $M_{i,k}$ stand for the i^{th} component of the feature representation of image I and clustroid M_k, respectively. The region R_k of the cluster C_k is defined to include those images whose feature distances to the clustroid are less than the radius multiplied by a parameter β that is determined experimentally.

We define the mapping function as follows:

$$MAP(R_k) = \arg\left(\max_{keyword \in T_k} \left(F_k(keyword)\right) \right),\ k = 1 \ldots K \tag{12}$$

In words, the function maps every cluster region into the keyword that is most frequent in the corresponding cluster. A rapid dimensionality reduction method is needed in order to provide a computationally feasible method to automatically tag new images added to the image database. The low-level dimensional feature space of each cluster is reduced by ordering the features in the increasing order of their standard deviation and selecting the ones with the lowest values. The number of selected features is chosen such that the semantic indexing accuracy in the reduced space will be almost as good as the accuracy in the original space. Formula (13) gives the mappings expressed as IF-THEN rules in the reduced space:

$$\begin{aligned} &\textbf{IF } feature_{\sigma(1),k} \in \left(M_{\sigma(1),k} - \beta * radius_{\sigma(1),k}\ , M_{\sigma(1),k} + \beta * radius_{\sigma(1),,k}\right) \\ &\textbf{AND } \ldots \\ &\ldots\ feature_{\sigma(v),k} \in \left(M_{\sigma(v),k} - \beta * radius_{\sigma(v),k}\ , M_{\sigma(v),k} + \beta * radius_{\sigma(v),k}\right) \\ &\textbf{THEN } keyword_k\ ,\ k = 1 \ldots K \end{aligned} \tag{13}$$

σ stands for the permutation that gives the indices of the ordered low-level features. v stands for the number of the first top selected features. $keyword_k$ is the most frequent keyword in cluster C_k.

5. EXPERIMENTAL RESULTS

We evaluate our algorithm for semantic indexing and retrieval on an image database of 2100 images. The color vector representation of each image has 3*8*8 elements since each image is partitioned into 8*8 overlapping blocks and the image color content is characterized by three components: hue (H), saturation (S) and value (V). We rescale hue and saturation to values between 0 and 255. The semantic indexing of the image database is obtained using the first scenario (Section 2.2). After the image representations in both spaces are obtained, the image database is randomly split in two sets: *training set* (67% out of 2100 images) and *test set* (33%). We use the training set to learn the mappings between the two types of features. First, we apply k-means algorithm to derive a two-level hierarchy of clusters, and the cluster validity is checked for every cluster. The values of the constants (a, b and c) in formula (1) are experimentally chosen as being 2.5, 0.5 and 0, respectively. We end up with a hierarchy containing 30 clusters at the first level and 70 clusters at the second level. The low-level feature spaces of the clusters from the second level are ordered in increasing order of standard deviations, and the most frequent keywords are calculated.

Experimentally, we find that a number between 16 and 20 features (out of 192 features) is sufficient in order to get a good accuracy of indexing on training set almost the same as in the original low-level feature space. We also notice that these features belong to neighboring blocks and which have similar values. Keywords whose meanings are related to color information are the most frequent keywords present in the clusters. Table1 presents the reduced dimensional space $(v = 20)$ for a cluster that is semantically characterized by the keyword *sunset*. Table 1 gives also the ranges for the most significant features when $\beta = 1.96$, which were derived by experimental results. The precision of the rule on the training set is 80% and recall is 50%. On the testing set, the precision is 68% and the recall is 70%. Figure 3 presents some sample images indexed as *sunset* by the corresponding rule.

Table 1. The most significant low-level features from left to right, and top to bottom.

Block #	H58	V51	V63	V64	V1	V62	H57	V59	V58
Range	0..23	6..38	0..19	0..19	1..41	0..41	0..35	0..50	0..56
Block #	V17	V57	V53	V49	V50	V9	V4	V26	V25
Range	19..91	0..54	0..64	0..82	0..88	0..88	0..83	22..146	15..143

Figure 3. Sample images indexed as *sunset* by the rule from Table 1.

For a more complete list of rules and results on keyword retrieval, visit our home page at http://ieelab-secs.secs.oakland.edu

6. CONCLUSIONS AND FURTHER WORK

This paper presented a CBIR system that automatically generates semantic concepts from low-level features. The set of derived associations between the two types of features makes the system overcome the lack of expressive power that low-level features incur. It also helps reduce the semantic gap between the user's conceptualization of a textual query and the query that is actually specified.

Since the presented CBIR system was developed using a hierarchy of clusters, an additional feature of our system will be to enable semantic browsing. On a graphical interface, the keywords may serve as a conceptually summary of the image database; the keywords will function as *landmarks* in the sense that they will help orient and direct the user by providing pointers during the browsing process.

Experimentally, we noticed that many clusters possess the property of having the most significant features situated in neighboring blocks. This leads us in pursuing future work on spatial distribution (top, bottom, left, right, and center) in expressing the set of derived mappings. We would also like to experiment our system with other types of primitive features.

REFERENCES

1. P. G. B. Enser, "Pictorial Information Retrieval," in Journal of Documentation, 51(2), pp. 126-170, 1995
2. M. S. Lew, D. P. Huijsmans and D. Denteneer, "Content based image retrieval: Optimal keys, Texture, Projections or Templates," in Image Databases and Multi-Media Search, 8, pp. 39-47, 1997
3. C. Faloutsos, W. Equitz and M. Flickner, "Efficient and Effective Querying by Image Content," in Journal of Intelligent Information Systems 3, pp. 231-262, 1994
4. T. Minka, "An image database browser that learns from user interaction," in Technical Report #365, MIT Laboratory, 1996
5. J. Corridoni, A. Delbimbo and P. Pala, "Image Retrieval by Color Semantics," in ACM Multimedia System Journal, 1998
6. S. F. Chang et al., "Semantic visual templates: linking visual features to semantics," in IEEE International Conference on Image Processing (ICIP), pp. 531-535, 1998
7. I. K. Sethi, I. Coman, B. Day et al., "Color-WISE: A system for image similarity retrieval using color," in Proc. SPIE: Storage and Retrieval for Image and Video Databases, 3132; pp. 140-149, 1998
8. J. R. Smith and S. F. Chang, "Tools and Techniques for Color Image Retrieval," in Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases IV, 2670, pp. 381-392, 1996
9. V. N. Gudivada, V. V. Raghavan and K. Vanapipat, "A Unified Approach to Data Modeling for a Class of Image Database Applications," in IEEE Transactions on Data and Knowledge Discovery, 1994
10. D. Stan and I. K. Sethi, " Image Retrieval using a Hierarchy of Clusters," to appear in International Conference on Advances in Pattern Recognition, 2001
11. M. J. Swain, D. H. Ballard, " Color Indexing," in International Journal of Computer Vision, 7(1), pp. 11-32, 1991
12. I. K. Sethi, I. Coman I., " Image retrieval using hierarchical self-organizing feature maps," in Pattern Recognition Letters 20, pp. 1337-1345, 1999.
13. J. A. Rice, Mathematical Statistics and Data Analysis, Duxbury Press, California, 1995.
14. R. O. Duda, P. E. Hart, Pattern classification and scene analysis, John Wiley & Sons, Inc., New York, 1973.