Image Retrieval Using a Hierarchy of Clusters

Daniela Stan & Ishwar K. Sethi
Intelligent Information Engineering Laboratory,
Department of Computer Science & Engineering, Oakland University,
Rochester, Michigan 48309-4478
dstan@oakland.edu & isethi@oakland.edu

Abstract

The goal of this paper is to describe an efficient procedure for color-based image retrieval. The proposed procedure consists of two stages. First, the image data set is hierarchically decomposed into disjoint subsets by applying an adaptation of the k-means clustering algorithm. Since Euclidean measure may not effectively reproduce human perception of a visual content, the adaptive algorithm uses a non-Euclidean similarity metric and clustroids as cluster prototypes. Second, the derived hierarchy is searched by a branch and bound method to facilitate rapid calculation of the k-nearest neighbors for retrieval in a ranked order. The proposed procedure has the advantage of handling high dimensional data, and dealing with non-Euclidean similarity metrics in order to explore the nature of the image feature vectors. The hierarchy also provides users with a tool for quick browsing.

1. Introduction

The increasing rate at which images are generated in many application areas, gives rise to the need of image retrieval systems to provide an effective and efficient access to image databases, based on their visual content. While it is perfectly feasible to identify a desired image from a small collection simply by browsing, techniques that are more effective are needed with collections containing thousands, or millions of items. The current image retrieval techniques can be classified according to the type and nature of the features used for indexing and retrieval. Keyword indexing techniques manually 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. Their advantages consist of high expressive power, possibility to describe image content from the level of primitive features to the level of abstract features, involving a significant amount of reasoning about the meaning and purpose of the objects or scenes depicted .

On the other hand, manual indexing presents few drawbacks regarding the usefulness of the assigned keywords and the indexing time. Since the same picture can have different meanings for different people, different keywords could be associated with the same picture [1]. When the indexing time for every image takes few minutes, to index a collection of million images is an intensive and time consuming work.

Methods that permit image searching based on features automatically extracted from the images themselves are referred as content-based image retrieval (CBIR) techniques [2]. 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. To be along with the user's perception of image chromatic contents, the images are partitioned into blocks and a color histogram is calculated for each block. In this case, similarity matching also considers adjacent conditions among blocks with similar histograms. The most notable example of querying by color is IBM's QBIC system [4] that has been applied successfully in color matching of items in electronic mail order catalogues. One drawback of the current content-based image retrieval systems is their limitation to the low level features even if some researchers have attempted to fill the gap between low-level features and semantic features, by deriving high-level semantic concepts (harmony, disharmony, calmness, excitement) from color arrangements [5]. Another problem with the existing image retrieval systems is that these systems do not provide a summary view of the images in their database to their users. The necessity of a summary view appears when the user has no specific query image at the beginning of the search process and wants to explore the image collection to locate images of interest [6]. The indexing structure is also a big issue for CBIR systems. Image features are often very high dimensional or the similarity metrics are too complex to have efficient indexing structures. The existing multi-dimensional indexing techniques concentrate only on how to identify and improve indexing techniques that are scalable to high dimensional feature vectors in image retrieval [7]. The other nature of feature vectors in Image Retrieval, i.e. non-Euclidean similarity measures, cannot be explored using structures that have been developed based on Euclidean distance metrics such as the k-d trees, the R-d trees and its variants.

The goal of this paper is to provide a CBIR system that is scalable to large size image collection and is based on an effective indexing module that solves both high dimensionality and non-Euclidean nature of some color feature spaces. The module is built using an adaptation of k-means clustering in which the metric is 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. These prototypes give the system the capability of quick browsing through the entire image collection. The proposed system also uses a branch and bound tree-search module that applied to the hierarchy of the resultant clusters will facilitate rapid calculation of the nearest neighbors for retrieval.

The paper is organized as follows. Section 2 describes the color feature representation of the images from the database used in the proposed procedure. Section 3 explains how the hierarchy of similar groups is built by the adaptive k-means algorithm and Section 4 describes how the search is carried out by the branch and bound algorithm. Section 5 considers the effectiveness of the approach and how the user can browse elegantly through the image database; these considerations are expounded with experiments on a database of 2100 images. The paper concludes with some final comments and a note on future work.

2. Color feature representation

Color is one of the most widely used features for image similarity retrieval. This is not surprising given the facts that color is an easily recognizable element of an image and the human visual system is capable of differentiating between infinitely large numbers of colors.

In this paper, we use the Color-WISE representation for image retrieval described in detail in [8]. 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. 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 [9] 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. 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. 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 for HSV).

3. Hierarchy of clusters

Clustering is a discovery process in data mining. It groups a set of data in a way that maximizes the similarity within clusters and minimizes the similarity between two different clusters. The discovered clusters can explain the characteristics of the underlying data distribution and serve as foundation for other analysis techniques [10]. Clustering is also useful in implementing the "divide and conquer" strategy to reduce the computational complexity of various decision-making algorithms in pattern recognition.

We use a variation of k-means clustering to build a hierarchy of clusters. At every level of the hierarchy, the variation of k-means clustering 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 results of this hierarchy decomposition are represented by a tree structure in which each node of the tree represents a cluster prototype and at the last level, each leaf represents an image. The hierarchy of the cluster prototypes gives the system the capability of quick browsing through the entire image collection.

This 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

Clustering methods require that an index of proximity or associations be established between pairs of patterns [10]. A proximity index is either a similarity or dissimilarity. The more two images resemble each other, the larger a similarity index and the smaller a dissimilarity index will be.

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. 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 [8]. 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 $\left(h_{q_i}, s_{q_i}, v_{q_i}\right)$ and $\left(h_{t_i}, s_{t_i}, v_{t_i}\right)$ 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:

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

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: $D_s\left(s_{q_i},s_{t_i}\right)=\frac{\left\|s_{q_i}-s_{t_i}\right\|}{256}$ (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: $S(Q,T)=\frac{\sum_{i=1}^{M\times N} m_i S(q_i,t_i)}{\sum_{i=1}^{M\times N} m_i}$ (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. By building the hierarchical tree having the cluster prototypes as interior nodes, the system will allow users to browse the image collection at different levels of details.

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

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

that is defined as one that maximizes the criterion sum-of-squared-error function:

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 [12], 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 in [13] 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. The browsing and search strategy

The significant feature of our scheme is the possibility of quick browsing of the image set when no query image is specified. The user can browse first the highest level of the tree representing the hierarchy and get summary views of the entire image collection in the form of the prototypes of the clusters at that level. By traversing down the tree, the user gets finer and finer details from one level to another. Using an analogy with the view layers defined using a hierarchy of self-organization maps [6], we can consider the first level of the tree as a global view level of the entire image collection, the intermediate levels as regional levels and the last layer of the tree as a local layer giving the most detailed summary views for the images. Each node from the last layer points to a group of similar images named image layer.

When a query image is present, the second phase of our algorithm is involved. The search strategy implies a branch and bound algorithm in order to facilitate rapid calculation of the k-nearest neighbors for retrieval. We use the method defined in [14] which tests the nodes of the tree by two simple stopping rules that eliminates the necessity of calculating many distances. The first rule is meant to eliminate from consideration the node and its corresponding group of samples if the distance between the query and the node (clustroid) is greater than the sum between the current distance to the nearest neighbor and the farthest distance from the centroid to any sample from the cluster. The second rule reduces the number of calculations of distances between the query and the samples of the node that survived to rule 1. If the distance from the query to the clustroid is greater than the sum between the current distance to the nearest neighbor and the distance from the clustroid to a sample, do not calculate the distance between the sample and the query anymore.

To perform similarity search, the color representation of the query image is first matched at the first layer to determine the most similar cluster prototypes (nodes) that should be searched further. We eliminate from considerations each node from first layer for which rule 1 is satisfied. The matching is then repeated for the children of one of the nodes from the previous layer that survived to rule 1, and so on until the last layer is reached, which brings out a group of images that can be the most similar to the query image. We do not need to compare each one of these images with the query image since rule 2 filters out the images that not satisfy it. For the images that finally survive, the distances to the query image are calculated and ordered to find the current nearest neighbors. Then the algorithm is applied for the next node that was carried on after applying rule 1 and the table of the current nearest neighbors is updated as needed.

5. Experimental Results

We evaluate our algorithm for browsing 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, saturation and intensity. To perform color-based similarity retrieval, 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 rescale hue and saturation to values between 0 and 255. In order to obtain the first level (global layer) of the hierarchy, we apply k-means algorithm for k = 2, 3 … and at each consequent k, the cluster validity is checked, to ensure that the number of elements in every cluster is a moderate one and the sum-of-squared-error criterion to be satisfied. Comparing the values of the test statistic (7) and the values of the threshold (8) with respect to inequality (9), the possible number of clusters for different small values of the significance level is obtained. Since the value of the statistic for K = 31 is greater than the threshold for consecutive small values of p, we choose the value of K to be 30. Further, we split the nodes having at least 30 images (at least 2% out of the data set) by applying k-means algorithm again and so, a lower level (regional layer) of the hierarchy is obtained. The minimum number of elements in every cluster to go further with splitting is decided as a compromise between the size of the terminal nodes and the number of nodes in the tree. Fewer elements in the final groups produce fewer distance computations in the retrieval stage, but larger number of distance computation in the search stage. We end up with a search tree having 81 nodes, 4 levels and an average of 40 images per terminal node.
Fig. 2 shows a retrieval result for browsing mode. The user browses the first level (global layer) of the tree and hypothetically speaking, the user decides to look for images similar with the prototype of cluster 9. The image will be updated with the images (found in hierarchical clustering process) that are close to the centers of

clusters at the next layer. Assuming that the user decides to see images similar with the first prototype of the second layer, the third layer (image layer) will display the group of images similar with the previous chosen prototype.

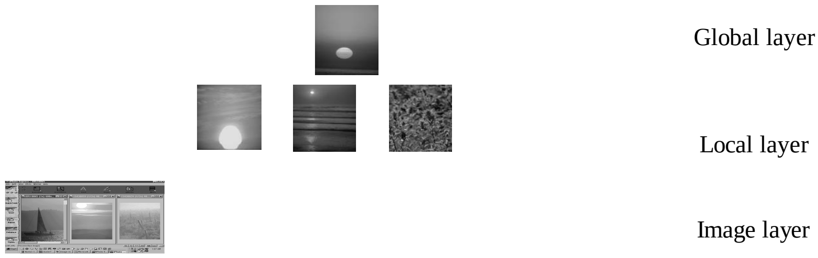

Figure 2.

Fig. 3 shows a retrieval result for search mode. The image query is in the top left of the image. The user wants to retrieve the most three similar images with the image query. Applying the proposed scheme, the following nodes are reached in order to find the 3-nearest neighbors: node 5 at first level, node 28 at second layer, 76 atthe final level. The nearest neighbors are picked up from the group of images pointed by node 76.

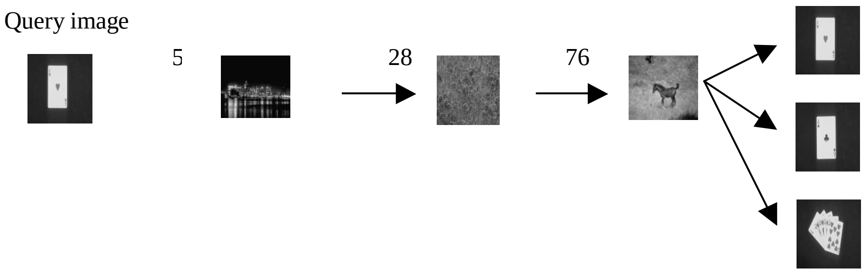

Figure 3.

For more results on color similarity retrieval visit our home page at http://iielab-secs.secs.oakland.edu

6. Conclusions and future work

This paper presented an efficient method for image retrieval. Since the proposed procedure organizes the color information as a hierarchy of different clusters, the user is provided with summary views of the entire image collection at different level of details. Fast calculation of the k-nearest neighbors is possible by using a branch and bound algorithm as a search strategy. As future work, we want to experiment our system with semantic features in addition to the low level ones. Since browsing computerized information has a social dimension, we will also develop an interface for better visualization of the information patterns being browsed and more effective means of communicating the browsing process.

References

1. Enser PGB. Pictorial information retrieval. Journal of Documentation, 51(2), pp. 126-170, 1995

2. Lew MS, Huijsmans DP, Denteneer D. Content based image retrieval: Optimal keys, Texture, Projections or Templates. Image Databases and Multi-Media Search, 1997; 8; 39-47

3. Faloutsos C, Equitz W, Flickner M. Efficient and Effective Querying by Image Content. Journal of Intelligent Information Systems 3, 1994; 231-262

4. Niblack W, Barber R, Equitz W. et al. The QBIC project: querying images by content using color, texture, shape. Proc. SPIE: Storage and Retrieval for Images and Video Databases 1993; 1908; 173-187

5. Corridoni J, Delbimbo A, Pala P. Image retrieval by color semantics. ACM Multimedia System Journal, 1998

6. Sethi IK, Coman I. Image retrieval using hierarchical self-organizing feature maps. Pattern Recognition Letters 20, 1999; 1337-1345

7. Rui Y, Huang TS, Chang SF. Image Retrieval: Past, Present, and Future. Journal of Visual Communication and Image Representation, 1999; 10; 1-23.

8. Sethi IK, Coman I, Day B et al. Color-WISE: A system for image similarity retrieval using color. Proc. SPIE: Storage and Retrieval for Image and Video Databases, 1998; 3312; 140-149

9. Smith JR and Chang SF. Tools and Techniques for Color Image Retrieval. Proceedings of the SPIE: Storage and Retrieval for Image and Video Databases IV, 1996; 2670; 381-392

10. Jain AK, Dubes RC. Algorithms for Clustering Data. Prentices Hall Advanced Reference Series, 1998

11. Swain MJ, Ballard DH. Color Indexing. International Journal of Computer Vision, 1991; 7(1); 11-32

12. Rice JA. Mathematical Statistics and Data Analysis. Duxbury Press, 1995.

13. Duda RO, Hart PE. Pattern classification and scene analysis. John Wiley & Sons, Inc., 1973.

14. Fukanaga K, Narendra PM. A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers, 1975; C-24; 750-753

Daniela Stan is a graduate student in Computer Science and Engineering at Oakland University. She received her M.A. in Computer Science in 1999 from Wayne State University and B.S. in Mathematics in 1993 from University of Bucharest, Romania. She is currently involved in the area of content-based image retrieval and data mining.

Ishwar K. Sethi is currently Professor and Chair of Computer Science and Engineering at Oakland University. His research interests include pattern recognition, data mining, and multimedia information processing and indexing. Professor Sethi serves on the editorial

boards of several international journals including *IEEE Trans. Pattern Analysis and Machine Intelligence and IEEE Multimedia.*