AN EFFICIENT LOW-DIMENSIONAL COLOR INDEXING SCHEME FOR REGION-BASED IMAGE RETRIEVAL

Yining Deng and B. S. Manjunath

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560
deng@iplab.ece.ucsb.edu, manj@ece.ucsb.edu

ABSTRACT

In this work, an efficient low-dimensional color indexing scheme for region-based image retrieval is presented. The colors in each image region are first quantized so that only a small number of cluster centroids are needed to represent the region color information. The proposed color feature descriptor consists of these quantized colors and their percentages in the region. A similarity distance measure is defined and shown to be equivalent to the quadratic color histogram distance measure. The quantized colors are indexed in the 3-D color space so that high-dimensional indexing can be avoided. During the search process, each quantized color in the query is used as a separate cue to find matches containing that color. The matches from all the query colors are then joined to obtain the final retrievals. Experimental results show that the proposed scheme is fast and accurate compared to the color histogram approach.

1. INTRODUCTION

The use of low-level visual features to search and retrieve relevant information in the image and video databases has drawn much research attention in the recent years. Among all the visual features, color is perhaps the most dominant and distinguishing one in many applications. A number of methods have been proposed to characterize color information in the images, among which the color histogram method is one of the more popular approaches. A histogram intersection method is proposed to measure the similarity between two color histograms [12], and in [6] a quadratic distance measure is described that can model the perceptual similarity between different color bins. Other proposed color matching techniques include color moments [11] and color correlograms [7].

While all these approaches provide good ways of characterizing color information, most of them share the common problems of high-dimensional features:

- high computational complexity in distance calculation.
- inefficiency in indexing and search because the performance degenerate rapidly with the increase in dimension.

Many methods have also been proposed to overcome these problems, such as, the use of SVD [6] to reduce the dimensionality, the use of only dominant colors in the histogram [15], and a multiresolution approach to color clustering [13]. However, these methods have their own drawbacks. In this work, an efficient color indexing scheme is proposed to avoid the problems of high-dimensional indexing. The scheme is built upon a region-based image retrieval system [9] and is based on the observed fact that a small number of colors are usually enough to characterize the color information in an image region.

The key to this approach is to index these representing colors individually in the 3-D color space rather than forming a histogram feature vector and indexing it in the feature space. The idea is similar to the multiple text keyword search. Each color can be thought of as a keyword and each entry in the database contains several "color keywords". During the search process, matching entries containing each "color keyword" are found and the final results are the join of these matches.

A similar idea to this color indexing scheme is also described in [1], but our approach differs from theirs in the following aspects:

- The distance measure defined in our method is more generalized and is shown to be equivalent to the quadratic color histogram distance measure.
- R-tree [5] is used for indexing in the work of [1], while in our method a fixed lattice structure is used and no tree traversing is needed during the search, which results in faster retrieval.
- The methods for joining the matches from individual query colors are also different.

2. COLOR FEATURE

The first step in our color image indexing scheme is to extract the color features. Before the feature extraction, the image is segmented into a set of regions by using the Edgeflow algorithm [8]. A perceptual color quantization algorithm [4] is then used to quantize the colors in each image region. After quantization, only a small number of colors remain and the percentage of these colors are calculated. Each quantized color and its corresponding percentage form a pair of attributes that describe the color characteristics in an image region. The color feature descriptor F is defined to be a set of such attribute pairs:

$$F = \{\{c_i, p_i\}, i = 1, \ldots, N\} \quad (1)$$

where N is the total number of quantized colors in the region, c_i is a 3-D color vector, p_i is its percentage, and $\sum p_i = 1$.

The similarity between two color features F_1 and F_2 is measured by the following distance function $D(F_1, F_2)$,

This work was supported in part by a grant from Samsung Electronics.

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{1i,2j}p_{1i}p_{2j} \quad (2)$$

where $a_{k,l}$ is the similarity coefficient between colors c_k and c_l,

$$a_{k,l} = \begin{cases} 1 - d_{k,l}/d_{max} & d_{k,l} \le T_d \\ 0 & d_{k,l} > T_d \end{cases} \quad (3)$$

where $d_{k,l}$ is the distance between color c_k and c_l,

$$d_{k,l} = \|c_k - c_l\| \quad (4)$$

T_d is the maximum distance for two colors to be considered similar which is determined experimentally, $d_{max} = \alpha T_d$, and α is set to 1.2 in the experiments.

The above distance measure can be shown to be equivalent to the quadratic color histogram distance measure:

$$D'^2(F_1', F_2') = (F_1' - F_2')^T A'(F_1' - F_2') \quad (5)$$

where F' is a histogram vector $[p_1' \ldots p_N']$ of length N', if the coefficients of matrix A', $a_{k,l}'$ are defined the same as $a_{k,l}$ in (3). In fact, if the number of color bins in the histogram vector N' is large enough such that all the quantized colors are color bins of the histogram method, a color histogram vector can be constructed using the percentage values p_i of the proposed new method. Ignoring all the zero entries and rewriting the quadratic distance in terms of summation of the coefficients gives:

$$D'^2(F_1', F_2') = \sum_{i=1}^{N_1}\sum_{j=1}^{N_1} a_{1i,1j}p_{1i}p_{1j} + \sum_{i=1}^{N_2}\sum_{j=1}^{N_2} a_{2i,2j}p_{2i}p_{2j} - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{1i,2j}p_{1i}p_{2j} \quad (6)$$

Since during quantization, the minimum distance between two quantized colors can be set to T_d using an agglomerative clustering algorithm [4], Eqn. (6) can be further simplified because

$$a_{1i,1j} \text{ or } a_{2i,2j} = \begin{cases} 1 & i = j \\ 0 & i \ne j \end{cases} \quad (7)$$

and therefore $D'(F_1', F_2') = D(F_1, F_2)$.

The procedures to calculate $D(F_1, F_2)$ is as follows:

1. Calculate the square terms in Eqn. (2).
2. Take either color feature, say F_1, as the reference feature. For each color in F_1, find all its similar colors in F_2, i.e., the colors that have distance less than T_d.
3. Calculate the negative term in Eqn. (2).

In the database searching process, the reference feature is usually the query feature and the matching colors are found as the direct results of database searching.

3. INDEXING AND SEARCH SCHEME

The key difference of the proposed approach from other color indexing methods is that the data are indexed in the 3-D color space instead of the feature space. The indexing is built upon the quantized colors and each color is indexed individually. The corresponding percentages and the image region labels are stored along with the quantized colors in the database indexing nodes.

Similarly, during the query process, each quantized color in the query feature is searched separately to find matching image regions which contain that color. The results are then combined together so that image regions containing similar colors as the query are found. The entries in each indexing leaf node are presorted based on the region labels during the indexing stage to speed up the joining process. The final retrievals are rank ordered by their color feature distances. Fig. 1 illustrates the basic indexing and search scheme.

3.1 Lattice Indexing Structure

Since the indexing is built on the low-dimensional 3-D color space, there are a few efficient indexing structures such as R*-tree [2] and SS tree [14] that can be directly used. However, for the particular application of similarity retrieval, an indexing tree might not be necessary because of the following reason. Most of the time the goal is only to find similar images in the database, and the system, for example, does not have to retrieve all top 100 matches if only top 40 of them are actually similar. Therefore, fixed range queries can be used to find all the matches within a certain distance of the query feature and these matches can then be ranked in order. If such kind of queries are performed, there is no need for a balanced indexing tree.

Based on this fact, a lattice structure is used for indexing. The lattice points are fixed uniformly in space during the database designing stage and can be thought of as leaf nodes of the indexing tree. During the indexing process, each color is assigned to its nearest lattice point. Since there are no parent nodes, there is no tree traversing during the search process. It is to be noted that in addition to the nearest lattice point to the query, other nearby lattice points are also to be considered because the query point can lie near the boundaries. Fig 2. illustrates this in the 2-D plane, where the desired search radius r is the query search range and the actual search radius R is the minimum search distance for lattice points such that the desired sphere of radius r is covered. Let ρ denotes the minimum radius of a sphere that can cover a *Voronoi* cell, as shown in Fig. 2. r and R are related by:

$$R = r + \rho \quad (8)$$

Since $R > r$ and an indexing node contains all the entries in its *Voronoi* cell, part of the search space could be unnecessary. For

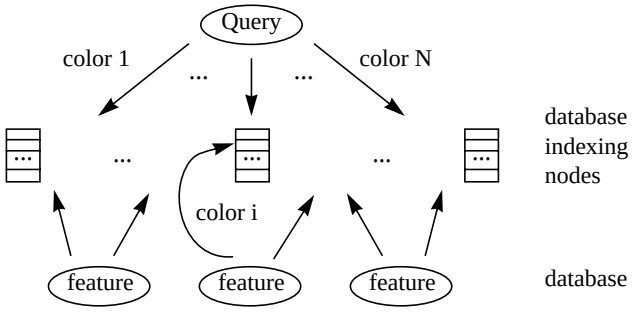

Fig. 1. Basic indexing and search scheme.

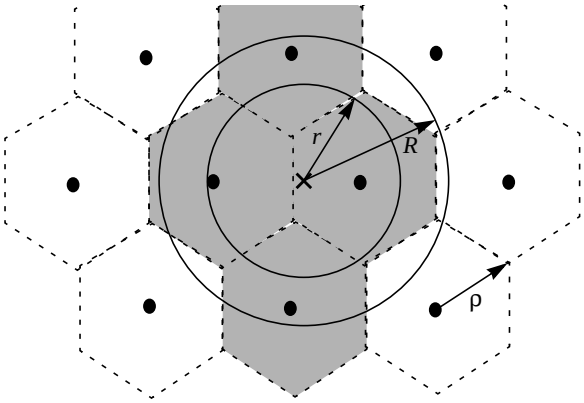

Fig. 2. Lattice structure and search mechanism. Point "x" is the query. A hexagonal lattice structure is shown. *r* and *R* are the desired and actual search radius respectively. ρ is the minimum radius of a sphere that can cover a *Voronoi* cell.

example, in Fig. 2 it can be seen that the actual search space includes all the shaded areas. For a given query range *r*, The value of ρ in the lattice design is important to the retrieval performance. A smaller value of ρ means more efficiency in the search space. However, there is a trade-off because the number of the indexing nodes are increased and the indexing itself becomes less efficient. Regardless of the design, however, the search complexity is low. The number of indexing nodes need to be accessed per query color is on the order of $O(R^3/\rho^3)$ and does not depend on the database size.

The D_3* type of lattice [3] is chosen for indexing due to its optimal properties in both accuracy and efficiency. The structure of D_3* lattice is quite simple. The basic lattice consists of the points (x, y, z) where *x*, *y* and *z* are all even or all odd integers. These points can be scaled and shifted to have desired lattice point intervals and locations.

By using the proposed indexing scheme, insertions and deletions of database entries are straightforward, which allows the database to be dynamic. Because the positions of the lattice points are known, only a few calculations are needed to find the nearest lattice point for any color to be indexed. The region label of the new entry is then compared to the list of sorted region labels in the indexing node and is inserted in the right order.

3.2 Search Procedures

The complete search procedures include the following steps:

1. For each query color, find the matching regions that contains the similar color by using the lattice indexing structure. To quickly eliminate some false matches, a threshold T_p is set for a query percentage p_q and a retrieved one p_r. A matching region is eliminated if $|p_q - p_r| > T_p$.
2. Join the matching results from all the query colors and eliminate all the false matches. A simple "and" operation in the joining process could falsely eliminate some good retrievals, because even if two images do not have exact match of each color, their can still look similar if the majority of their colors match well. Therefore, the total percentages of the matched color in both the query and the retrievals are used in the following determining criterions to be satisfied:

$$\sum_i p_{qi} \geq T_t \text{ and } \sum_i p_{ri} \geq T_t \quad (9)$$

where the summations are over all the matched colors and T_t is a threshold set depending on the application need. Since only additions and comparisons are involved in this operation, it is very fast and effectively eliminates a large number of false matches.

3. Calculate the distances between the query and the retrievals and rank them in order. Because the matching colors between a retrieval and the query are obtained directly as the results of indexing and searching, Step 2 of the distance calculation procedures described in Section 3 can be omitted. For indexing and distance measure to be consistent, the desired search radius *r* should be the same as T_d, the maximum distance for two colors to be considered similar, defined in Section 3, i.e., $r = T_d$.

4. EXPERIMENTAL RESULTS

The proposed algorithm is tested on a 2,500 color image database. After segmentation, more than 26,000 regions are obtained. A set of 25 image regions containing a variety colors and color combinations are chosen as queries for evaluation. To best characterize the color information, all the processing is done in the perceptual uniform CIE LUV color space. The parameter values set in the experiments are $\rho = 6.708$, $T_d = r = 13.416$, $R = 20.124$, $T_p = 0.5$, $T_t = 0.5$. Table 1 summarizes some of the experimental data. It can be seen that on average there are 3.5 representing colors per region after quantization. Because 4 numbers are needed to represent each color (3 for color and 1 for percentage), on average each color feature takes about only 14 numbers. The average number of nodes accessed per query feature is a small fraction ($134.9/1553 = 8.7\%$) of all the indexing nodes. The search speed is very fast and the average execution time including both CPU and I/O is only 0.16 s on a SGI origin200 server.

Table 1: Experimental Data

average number of colors per region	3.5
total number of indexing nodes	1553
average number of nodes accessed per query	134.9
average execution time (CPU plus I/O) per query	0.16 s

To evaluate the retrieval accuracy, the results of using color histogram features are used for comparison, because the proposed new method can be seen as a variation of the histogram approach in terms of the color feature and the distance measure. A 1024-D color histogram feature vector is extracted from each image region in the database. The dimension is set high to achieve the best possible results. The histogram features of the query set were compared with all the histogram features in the database through exhaustive searches to obtain top 100 retrievals. It is noticed,

however, that the new method needs only 14 numbers per color feature on average. For comparison, an alternative way of using only 14 numbers per feature is to perform SVD on quadratic matrix A as suggested in [6] to obtain a 14-D transformed color histogram vector. Again exhaustive searches in the database are performed to find the top 100 retrievals by using this approach.

Before the evaluation, subjective testing was done to determine the relevant matches for the query image regions in the database. The results from the histogram and the new method are marked by subjects to decide whether they are relevant or not. Because it is impractical to go through the entire database to find all the relevant matches for the queries, the union of the relevant retrievals from the two methods are used as approximate "ground truth" to evaluate the retrieval accuracy, which is measured by precision and recall defined as [10]

$$Precision(K) = C_K/K \text{ and } Recall(K) = C_K/M \qquad (10)$$

respectively, where K is the number of retrievals, C_K is the number of relevant matches among all the K retrievals, and M is the total number of relevant matches in the database obtained through the subjective testing.

The average precision and recall curves are plotted in Fig. 3 and Fig. 4. It can be seen from the figure that the new method achieves good results in terms retrieval accuracy compared to the histogram method. Its performance is close to the high-dimensional histogram method and is much better than the SVD approach.

A demo of region-based image search using the proposed color indexing method can be accessed at *http://maya.ece.ucsb.edu/Netra/*. The query image set can be seen at *http://maya.ece.ucsb.edu/~deng/Netra/Query/*.

5. CONCLUSIONS

In this work, an efficient low dimensional color indexing scheme in region-based image retrieval is presented. Experimental results show that the proposed method is fast while achieving good retrieval performance. How to extend this method from region color matching to global color matching of the entire images is our future research goal.

6. REFERENCES

[1] G.P. Babu, B.M. Mehtre and M.S. Kankanhalli, "Color indexing for efficient image retrieval", *Multimedia Tools and Applications*, vol. 1, no. 4, p. 327-48, November, 1995.

[2] N. Beckmann etal., "The R*-tree, an efficient and robust access method for points and rectangles", *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, p. 322-31, 1990.

[3] J.H. Conway and N.J.A. Sloane, *Sphere Packings, Lattices and Groups*, Springer-Verlag, New York, 1993.

[4] Y. Deng, C. Kenney, M.S. Moore and B.S. Manjunath, "Peer Group Filtering and Perceptual Color Quantization", Technical Report #98-25, ECE Dept., UCSB, September, 1998.

[5] A. Guttman, "R-trees: a dynamic index structure for spatial search", *Proc. of ACM SIGMOD Intl. Conf. on Management of Data*, p. 47-57, 1984.

[6] J. Hafner etal., "Efficient color histogram indexing for quadratic form distance functions", *PAMI*, vol. 17, no. 7, p. 729-36, July 1995.

[7] J. Huang etal., "Image indexing using color correlograms", *Proc. of CVPR*, p. 762-68, 1997.

[8] W.Y. Ma and B.S. Manjunath, "Edge flow: a framework of boundary detection and image segmentation", *Proc. of CVPR*, p. 744-749, 1997.

[9] W.Y. Ma and B.S. Manjunath, "NeTra: a toolbox for navigating large image databases", *Proc. ICIP*, vol.1, p. 568-71, 1997.

[10] J.R. Smith, "Image retrieval evaluation", *Proc. of IEEE Workshop on Content-based Assess of Image and Video Libraries*, p. 112-13, Santa Barbara, 1998.

[11] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints", *Proc. of SPIE Storage and Retrieval for Still Image and Video Databases IV*, vol. 2670, p. 29-40, 1996.

[12] M.J. Swain and D.H. Ballard, "Color indexing", *Intl. Journal of Computer Vision*, vol. 7, no. 1, p. 11-32, 1991.

[13] X. Wan and C.J. Kuo, "A multiresolution color clustering approach to image indexing and retrieval", *Proc. ICASSP*, 1998.

[14] D.A. White and R. Jain, "Similarity indexing with the SS-tree", *Proc. of Intl. Conf. on Data Engineering*, p. 516-23, 1996.

[15] H. Zhang etal., "Image retrieval based on color features: an evaluation study", *Proc. of SPIE*, vol. 2606, p. 212-220, 1995.

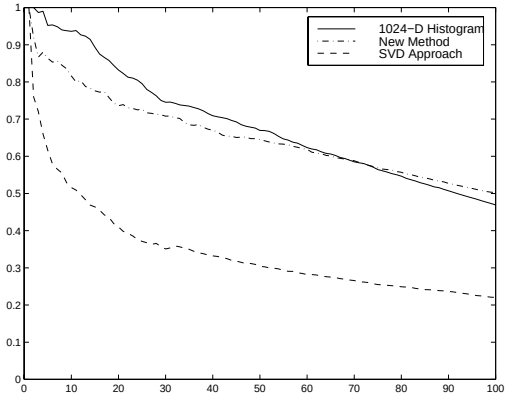

Fig. 3. Precision vs. number of retrievals.

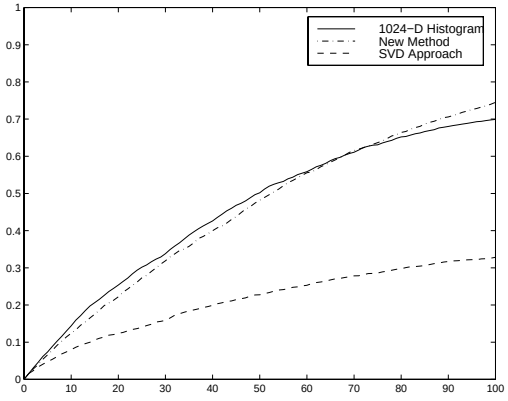

Fig. 4. Recall vs. number of retrievals.