An Efficient Color Representation for Image Retrieval

Yining Deng, *Member, IEEE*, B. S. Manjunath, *Member, IEEE*, Charles Kenney, Michael S. Moore, *Student Member, IEEE*, and Hyundoo Shin

***Abstract*—A compact color descriptor and an efficient indexing method for this descriptor are presented. The target application is similarity retrieval in large image databases using color. Colors in a given region are clustered into a small number of representative colors. The feature descriptor consists of the representative colors and their percentages in the region. A similarity measure similar to the quadratic color histogram distance measure is defined for this descriptor. The representative colors can be indexed in the three-dimensional (3-D) color space thus avoiding the high-dimensional indexing problems associated with the traditional color histogram. For similarity retrieval, each representative color in the query image or region is used independently to find regions containing that color. The matches from all of the query colors are then combined to obtain the final retrievals. An efficient indexing scheme for fast retrieval is presented. Experimental results show that this compact descriptor is effective and compares favorably with the traditional color histogram in terms of overall computational complexity.**

***Index Terms*—Color indexing, dominant color feature, region-based retrieval.**

I. Introduction

THE USE of low-level visual features to retrieve relevant information from image and video databases has drawn much research attention in recent years. Color is perhaps the most dominant and distinguishing visual feature. Color histogram is the most widely used color descriptor in content based retrieval research. A color histogram captures global color distribution in an image. While color histograms are easy to compute, they result in large feature vectors that are difficult to index and have high search and retrieval cost. In addition, spatial information is not preserved in a color histogram. Thus a large red color blob in a green background will have the same color histogram as an image containing the same number of randomly distributed red and green pixels. Several of the recently proposed color descriptors try to incorporate spatial information to varying degrees. These include the compact color moments [20], [21], binary color sets [19], color coherence vector [16], and color correlogram [10].

Manuscript received December 29, 1998; revised August 11, 2000. This work was supported in part by a grant from Samsung Electronics Company and by ONR/ASSERT under Award N00014-98–1-0515. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tsuhan Chen.

Y. Deng is with Hewlett-Packard Laboratories, Palo Alto, CA 94304 USA (e-mail: deng@hpl.hp.com).

B. S. Manjunath, C. Kenney, and M. S. Moore are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara CA 93106 USA.

H. Shin is with Samsung Electronics Company, Seoul, Korea (e-mail: shin@iplab.ece.ucsb.edu).

Publisher Item Identifier S 1057-7149(01)00097-5.

The feature vector dimensions of typical color descriptors are quite large. For example, the number of bins in a typical color histogram range from few tens to a few hundreds. The high dimensionality of the feature vectors result in high computational cost in distance calculation for similarity retrieval, and inefficiency in indexing and search.

Several methods have been proposed to overcome these problems. In [25] the dominant colors in the histogram are used, and a multiresolution color clustering is suggested in [23] to reduce the computational complexity in distance calculation. Singular value decomposition (SVD) [9] and Hilbert curve fitting [4] are used to reduce the dimensionality of the feature vectors. However, these methods have their own drawbacks. In [9] SVD is performed on the quadratic matrix of correlations between the color histogram bins. The resulting eigenvectors are not related to the feature data, and may result in significant errors when lower-dimensional transformed feature vectors are used to approximate the original feature vectors. The results of Hilbert curve fitting depend on the data distributions. Points that are close to each other in the original feature space might be far apart on the Hilbert curve. The distances in the original space might not be preserved well in the curve approximation.

The color moments descriptor proposed in [20], [21] has a compact representation. The moment descriptor includes the average, variance, and the third-order moment of the colors in the image. A recent study [14] shows that the color moment descriptor performs slightly worse than a high-dimensional color histogram. One drawback of the moment descriptor is that the average of all the colors might be quite different from any of the original colors. Given a color moment feature description, it is difficult to recover the actual colors in the image.

The proposed descriptor is also quite compact, and is based on the observation that a small number of colors is usually sufficient to characterize the color information in an image region. Since the descriptor captures the representative or dominant colors in a given region, we refer to it as the dominant color descriptor. The dominant color descriptor consists of the representative colors and their relative distribution in a given region. A similarity measure is defined for the proposed color descriptor and is shown to be equivalent to the popular quadratic color histogram distance measure. However, the difference between the new descriptor and the color histogram descriptor is that the representative colors are computed from each image instead of being fixed in the color space, thus allowing the feature representation to be accurate as well as compact. Unlike the compact color moments descriptor, the dominant color representation allows simple visualization of the color distributions in the image.

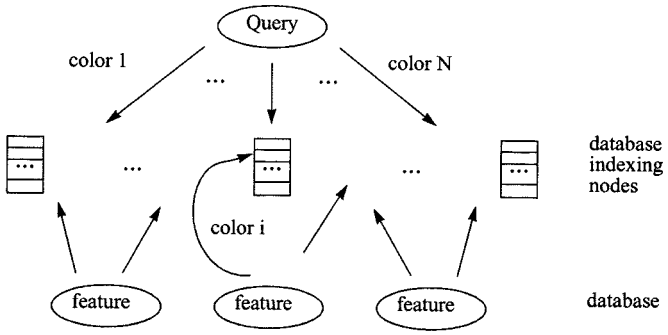

Fig. 1. Basic indexing and search scheme.

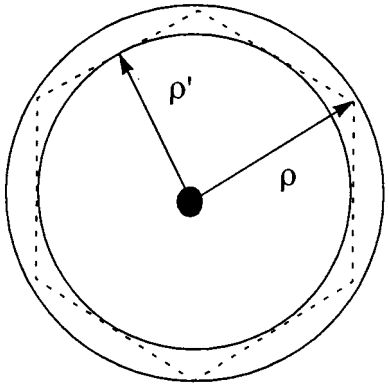

Fig. 2. The 2-D hexagonal lattice ρ is the minimum radius of a sphere that can cover the Voronoi cell, ρ' is the maximum radius of a sphere that the Voronoi cell can cover.

TABLE I
PRECISION AND RECALL FOR $K = 50$

entry	image region ID	3D color vector	percentage of the color in the region
1	ID_1	c_1	p_1
2	ID_2	c_2	p_2
3	ID_3	c_3	p_3
...	...	...	...

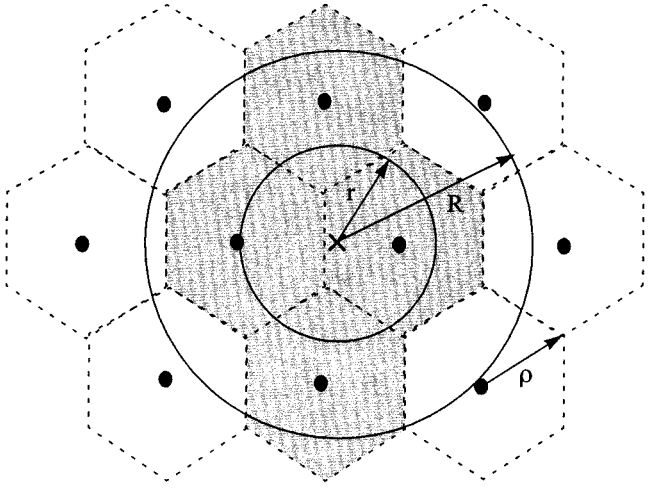

Fig. 3. Illustration of lattice structure and search mechanism in the 2-D plane. Point "x" is the query. A hexagonal lattice structure is shown and lattice points are marked, r and R are the desired and actual search radius, respectively, ρ is the minimum radius of a sphere that can cover a Voronoi cell.

TABLE II
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

An efficient color indexing scheme is proposed for fast search and retrieval using this dominant color descriptor. The key is to index the representative colors individually in a three-dimensional (3-D) color space rather than indexing 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. However, since each color has an associated percentage value that reflects the statistical distribution as well, the search process is more complicated than a keyword text search.

A similar approach to color indexing is described in [1] where the method is applied to databases of trademark and flag images. The images in such databases consist mostly of homogeneous regions and can be represented using a small number of colors. The proposed approach differs from [1] in the following aspects.

- A generalized distance measure to compare two arbitrary dominant color description is proposed, and is shown to be equivalent to the quadratic color histogram distance measure. The actual complexity of computing the distance is considerably lower in our method.
- A lattice structure is used for indexing. For similarity retrieval considered here, only range queries need to be computed. The range corresponds to the maximum distance between two colors that are considered similar. For fixed range queries, a tree index structure is not necessary and a fixed grid type of indexing is quite efficient [18].
- The query and the retrieved feature descriptions are not required to have the same sets of colors as long as the majority colors in the two feature descriptions match each other. In [1], however, each query color must have a matching color in the retrieval. While for the trademark and flag image databases in [1] it might be necessary to find exact color matches, it is not required in general.

The paper is organized as follows. Section II describes the dominant color descriptor. Section III describes the indexing and search method. Section IV provides experimental results, and Section V concludes with discussions.

II. DOMINANT COLOR DESCRIPTOR

The local color feature extraction starts first with color image segmentation. For image segmentation, we use the edgeflow algorithm [12]. Color clustering is performed on each segmented region to obtain its representative colors (see Section II-A).

After clustering, only a small number of colors remain and the percentages of these colors are calculated. Each representative color and its corresponding percentage form a pair of attributes

TABLE III
PRECISION AND RECALL FOR $K = 50$

Corel photo # of the query region	Prec. (hist.)	Prec. (new)	Prec. (SVD)	Recall (hist.)	Recall (new)	Recall (SVD)
114001	0.820	0.820	0.440	0.562	0.562	0.301
115015	0.600	0.420	0.040	0.625	0.438	0.042
117041	0.760	0.660	0.400	0.594	0.516	0.312
124043	0.800	0.740	0.160	0.500	0.463	0.100
127000	0.720	0.540	0.340	0.537	0.403	0.254
131072	0.500	0.360	0.240	0.556	0.400	0.267
132002	0.600	0.420	0.120	0.526	0.368	0.105
133006	0.740	0.620	0.220	0.536	0.449	0.159
141008	0.720	0.660	0.360	0.621	0.569	0.310
143028	0.540	0.500	0.480	0.540	0.500	0.480
144001	0.660	0.620	0.540	0.485	0.456	0.397
150008	0.740	0.780	0.060	0.507	0.534	0.041
151096	0.820	0.800	0.720	0.500	0.488	0.439
152006	0.900	0.840	0.340	0.417	0.389	0.157
158034	0.740	0.780	0.220	0.468	0.494	0.139
162026	0.800	0.660	0.460	0.571	0.471	0.329
164002	0.560	0.440	0.220	0.549	0.431	0.216
169049	0.640	0.840	0.120	0.438	0.575	0.082
171010	0.720	0.720	0.380	0.444	0.444	0.235
176090	0.400	0.600	0.100	0.426	0.638	0.106
178060	0.480	0.500	0.180	0.429	0.446	0.161
184038	0.800	0.800	0.820	0.533	0.533	0.547
194060	0.760	0.760	0.280	0.487	0.487	0.179
196032	0.240	0.500	0.260	0.240	0.500	0.260
258080	0.680	0.740	0.100	0.453	0.493	0.067
average	0.676	0.647	0.309	0.497	0.474	0.227

that describe the color characteristics in an image region. The dominant color descriptor F is defined to be

$$F = \{\{c_i, p_i\},\ i = 1, \cdots, N\} \tag{1}$$

where N is the total number of color clusters in the image region, c_i is a 3-D color vector, p_i is its percentage, and $\sum_i P_i = 1$. Note that N can vary from region to region.

A. Color Clustering

We use the color clustering method proposed in [6]. In this clustering, the pixel color values are vector quantized using a modified generalized Lloyd algorithm (GLA). Colors are represented in the perceptually uniform CIE LUV color space. The distortion D in each cluster is given by

$$D_i = \sum_n v(n)\|x(n) - c_i\|^2, \qquad x(n) \in C_i \tag{2}$$

where c_i is the centroid of cluster C_i, $x(n)$ is the color vector at pixel n, and $v(n)$ is the perceptual weight for pixel n. The perceptual weights are calculated from the local pixel statistics to account for the fact that human visual perception is more sensitive to changes in smooth regions than in textured regions as described in [6]. The update rule for this distortion metric is derived to be

$$c_i = \frac{\sum v(n)x(n)}{\sum v(n)},\ x(n) \in C_i. \tag{3}$$

An agglomerative clustering [7] is performed on the cluster centroids to further merge close clusters such that the minimum distance between two centroids exceeds a preset threshold T_d. The final quantized image is obtained by assigning each pixel to its closest cluster centroid.

Since the complexity of this color clustering algorithm is high, other faster color clustering methods can also be used to extract the dominant region colors. One possibility is to compute the traditional color histogram feature first. The representative colors can then be obtained by merging the color histogram bins. The disadvantage with this approach is that the representative colors are not as accurate as the colors extracted directly from the image pixels.

B. Color Similarity

The colors $\{c_i\}$ and the corresponding percentage of pixels p_i form the color descriptor as given by (1). Let $F_1 =$

$\{\{c_i, p_i\}, i = 1, \ldots, N_1\}$ and $F_2 = \{\{b_j, q_j\}, j = 1, \ldots, N_2\}$ be two color feature descriptions. The distance between F_1 and F_2 is given by

$$D^2(F_1, F_2) = \sum_{i=1}^{N_1} p_1^2 + \sum_{j=1}^{N_2} q_j^2 - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{i,j}p_i q_j \quad (4)$$

where $a_{i,j}$ is the similarity coefficient between colors c_i and b_j,

$$a_{i,j} = \begin{cases} 1 - d_{i,j}/d_{\max}, & d_{i,j} \le T_d \\ 0, & d_{i,j} > T_d \end{cases} \quad (5)$$

where $d_{i,j}$ is the Euclidean distance between color c_i and b_j

$$d_{i,j} = \|c_i - b_j\| \quad (6)$$

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

The above distance measure can be shown to be equivalent to the well-known quadratic color histogram distance measure [9]

$$D_h^2(H_1, H_2) = (H_1 - H_2)^T A (H_1 - H_2) \quad (7)$$

where H_1 and H_2 are traditional histogram vectors of length N_h, and the coefficients of matrix A are $a_{i,j}$. In fact, if the number of color bins in the histogram vector N_h is large enough such that all the representative colors are color bins of the histogram method, a color histogram vector can be constructed using the percentage values p_i. Ignoring all the zero entries and rewriting the quadratic distance gives

$$D_h^2(H_1, H_2) = \sum_{i=1}^{N_1}\sum_{k=1}^{N_1} a_{i,k}p_i p_k + \sum_{j=1}^{N_2}\sum_{l=1}^{N_2} a_{j,l}q_j q_l - \sum_{i=1}^{N_1}\sum_{j=1}^{N_2} 2a_{i,j}p_i q_j. \quad (8)$$

During clustering, the minimum distance between two cluster centroids can be set to T_d as discussed above. Noting that

$$a_{i,k} = \begin{cases} 1, & i = k \\ 0, & i \ne k \end{cases}, \quad \text{and } a_{j,l} = \begin{cases} 1, & j = l \\ 0, & j \ne l \end{cases} \quad (9)$$

it is easy to show that $D_h(H_1, H_2) = D(F_1, F_2)$.

III. Search and Retrieval

Each object or region in the database is represented using the dominant color descriptor as defined in (1). Typically, three to four colors provide a good characterization of the region colors. Given a query image, similarity retrieval involves searching the database for similar color distributions as the input query. Since the number of representative colors is small, one can first search the database for each of the representative colors separately, and then combine the results. Fig. 1 illustrates the basic indexing and search scheme.

Searching for individual colors can be done very efficiently in a 3-D color space. We consider here only fixed range queries where T_d [see (5)] limits the search range. While one can use well-known tree structures such as the R*-tree [2] and SS tree [24], a one-layer representation with a uniform space partitioning designed specifically for the given range is more effective [18]. One approach is to partition the color space into a finite number of cubics of the same size. However, such rectangular grid structures are not the most efficient for spherical range queries.

A. D_3^* Lattice Structure

The D_3^* lattice has certain desirable properties in sphere packing [5] that are relevant to spherical range queries

1) accuracy: D_3^* lattice gives the minimal total mean squared quantization errors among all the lattice structures in the 3-D space;
2) efficiency: D_3^* lattice has the most optimal covering of the 3-D space, i.e., the ratio of the volume of the smallest sphere that covers the Voronoi cell to the volume of the cell is minimum for the D_3^* lattice.

The structure of a 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. For example, (0, 0, 0), (1, 1, 3), and (2, 10, 20) belong to the D_3^* lattice. These points can be scaled and shifted to have desired lattice point intervals and locations. There are two important parameters in the lattice design: ρ, the minimum radius of a sphere that can cover the Voronoi cell, and ρ', the maximum radius of a sphere that the Voronoi cell can cover. Fig. 2 illustrates these parameters for the 2-D case. For the basic D_3^* lattice, $\rho'_{basic} = \|(0, 0, 0) - (1, 1, 1)\|/2 = 0.866$. The calculation of ρ, however, is a slightly more complicated. In [5], the ratio between these two variables is provided, i.e. $\rho'/\rho = 0.7747$. The value of ρ is set during the lattice design. One can calculate the scaling factor s that scales the basic D_3^* lattice using the following relationship:

$$s = 0.7747\rho/\rho'_{basic}. \quad (10)$$

For example, given the lattice points (1, 1, 1) and (2, 0, 0), the corresponding scaled lattice points are (s, s, s) and ($2s$, 0, 0), respectively.

Given an arbitrary (query) point c in the 3-D space, the corresponding nearest lattice point b_c can be easily computed as follows. Let

$$b_e = round\left(\frac{c}{L}\right) \cdot L \quad (11)$$

and

$$b_o = round\left(\frac{c - L/2}{L}\right) \cdot L + L/2 \quad (12)$$

where $L = 2s$, and L is the step size along the coordinate axes. For example, $L = 2$ for the basic lattice. The origin is assumed to be a lattice point. Note that b_e is the nearest even lattice point and b_o is the nearest odd lattice point to the given point c. Let $b = \{b_e, b_o\}$. Then

$$b_c = \arg\min_b(\|b - c\|). \quad (13)$$

For a range query, one need to compute all the lattice points that are within a sphere of radius R centered at a given point in the 3-D color space. For this, first compute all the lattice points in the cubic of size $2R$ by $2R$ centered at the query point. This can

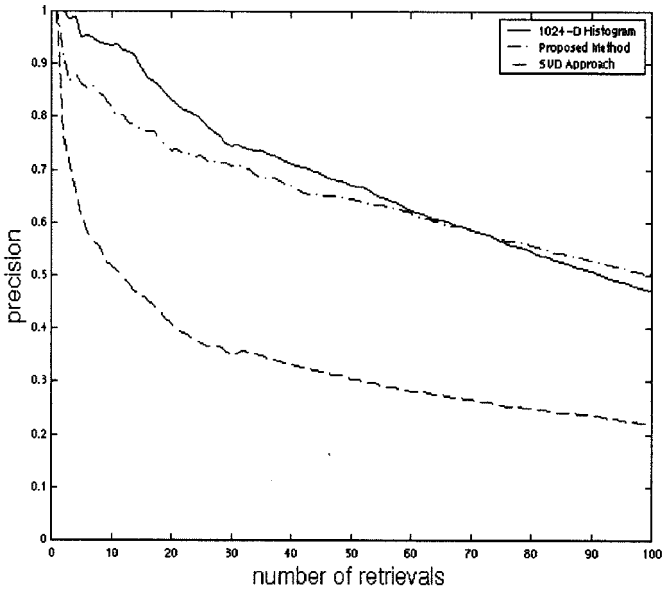

Fig. 4. Average precision versus number of retrievals.

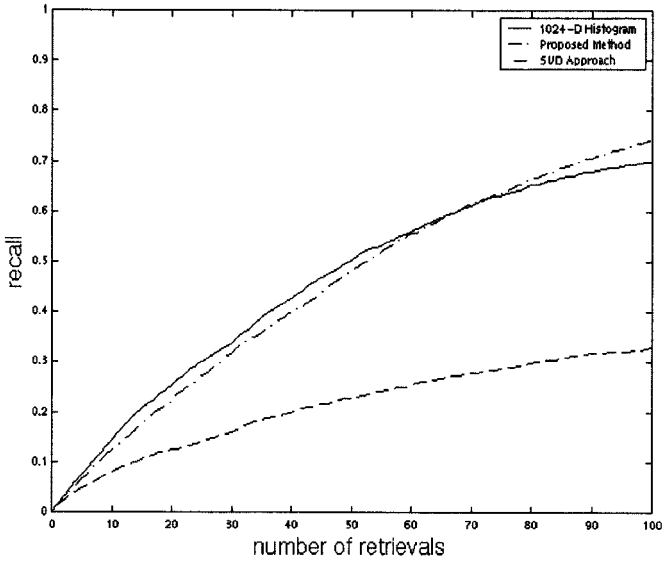

Fig. 5. Average recall versus number of retrievals.

be done using eqs. (11)–(13). The lattice points that are outside the range R are then eliminated by computing the Euclidean distance to the query point.

B. Indexing

In building the database, each region color is assigned to its nearest lattice point. Table I shows an example of the data structure of an indexing node. The image region ID is a unique integer label that identifies each region in the database. The entries in each index node are sorted by these ID numbers. After the indexing node of the new entry is found, its region ID is compared to the list of sorted region IDs in the node and is inserted into the right location in the sorted list. The proposed indexing scheme allows the database to be dynamic, which means that insertions and deletions of database entries are straightforward and without the need to reconstruct the entire index structure of the database.

During the search process, however, in addition to the nearest lattice point to the query color, other nearby lattice points are also to be considered because the query color can lie near the boundaries of the lattice cells. Fig. 3 illustrates this for the two-dimensional (2-D) case 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 denote the minimum radius of a sphere that can cover a Voronoi cell, as shown in Fig. 3. Note that $R = r + \rho$.

Since $R > r$ and an indexing node contains all the entries in its Voronoi cell, a part of the search space does not contain any relevant matches. For example, for the 2-D case shown in Fig. 3 the actual search space includes all the shaded areas. For a given search range r, the value of ρ in the lattice design is important to the retrieval performance. A small value of ρ means that the actual search space is only slightly larger than the desired search space, and therefore most of the accessed indexing nodes are relevant. However, there is a trade-off because the number of the indexing nodes increase as ρ decreases and the indexing itself becomes less efficient. The number of indexing nodes accessed per query color is $O(R^3/\rho^3)$ and does not directly depend on the database size.

The lattice structure can also have more than one layer of representation. If a Voronoi cell is dense with too many entries, its space can be further divided into a set of subcells. This results in a hierachical lattice structure. Hierachical lattice structures have been used in VQ-based image coding [15]. There should be multiple fixed search ranges, one for each level. A careful design of the hierachical structure could improve retrieval efficiency. However, in the experiments we use only a single layer representation.

C. Search Procedure

The complete search procedure includes the following steps.

Step 1) For each query color, find the matching regions that contain similar colors by using the lattice indexing structure. To quickly eliminate some false matches, a threshold T_p is set for the difference between the query percentage p_i and the retrieved percentage q_j. A matching region is eliminated if the following condition is not satisfied:

$$|p_i - q_j| < T_p. \tag{14}$$

Step 2) Join the matching results from all the query colors and eliminate all the false matches. Regions that satisfy the following two conditions are considered as the final retrieval candidates:

$$\sum_i p_i \geq T_t \qquad \text{and} \qquad \sum_j q_j \geq T_t \tag{15}$$

and where i and j index the matched colors. Partially matched entries in the database are quickly eliminated. T_t is set to 0.6 in the experiments.

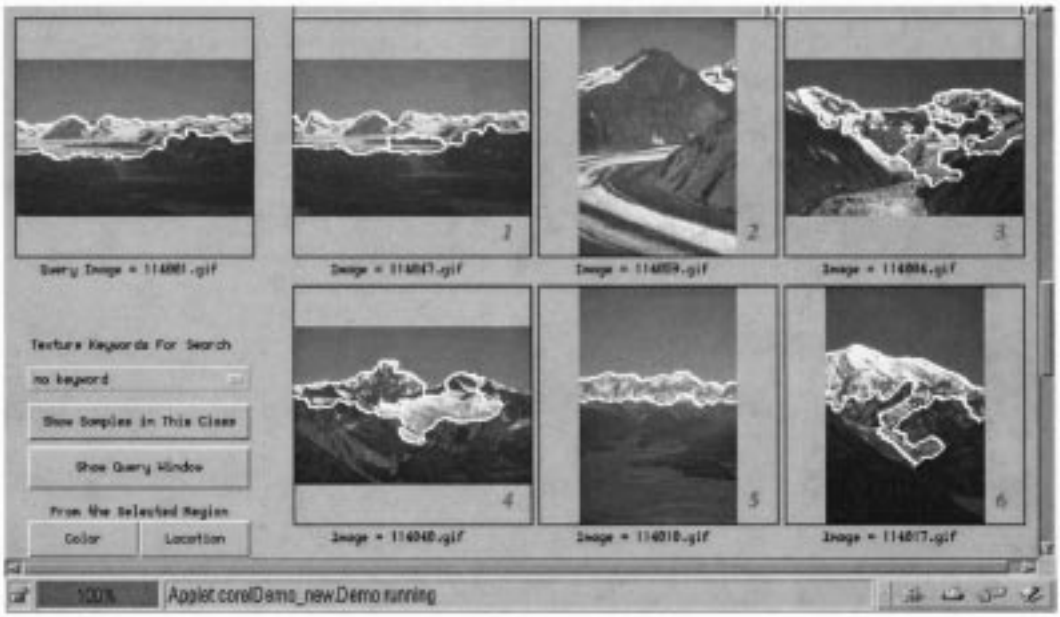

Fig. 6. Example of region-based image search using the dominant color descriptor. The query is the snowcap on the mountain. Top six retrievals are shown.

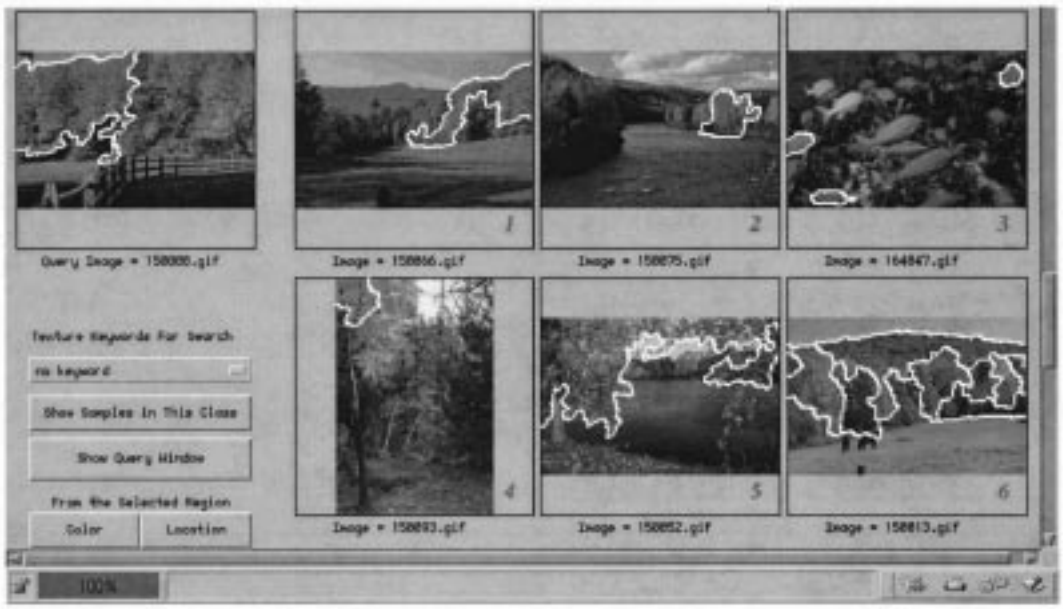

Fig. 7. Example of region-based image search using the dominant color descriptor. The query is the foliage in autumn. Top six retrievals are shown.

Step 3) Calculate the distances between the retrievals and the query and rank them in order. For the indexing and the distance measure to be consistent, the desired search radius r should equal T_d, the maximum distance for two colors to be considered similar, as defined in Section III.

Step 4) If a range query is issued, all of the matching candidates with distances smaller than the given range are returned. If an N nearest neighbor query is issued, the top N candidates are returned.

Step 1) and Step 2) quickly eliminate a large number of false matches. Step 3) involves more expensive calculations, but only for the final retrievals in the database. Overall, the computational complexity of the search procedure is low. Note that the number of random disk accesses is $O(mn)$, where n is the number of colors in the query, typically 3–5, m is the number of nodes accessed per query color. Note that the traditional color histogram is difficult to index as the complexity of indexing grows rapidly with the number of dimensions. In contrast, the dominant color descriptor is indexed in the low-dimensional 3-D color space.

IV. Experimental Results

The dominant color descriptor is tested on a database of 2500 color images from COREL. After the segmentation, more than

26 000 regions are obtained. Among them, 25 image regions containing a variety of colors and color combinations are chosen as queries. To best characterize the color information, all of the processing is done in the perceptually uniform CIE LUV color space. The parameter values used in the experiments are $\rho = 6.708$, $r = 13.416$, $R = 20.124$, $T_p = 0.5$, $T_t = 0.5$. Table II summarizes some of the experimental data. On average there are 3.5 representative colors per region after clustering. Four numbers are needed to represent each dominant color, three for the color value and one for the percentage value. Thus, on average the proposed dominant color descriptor requires only 14 numbers. The average number of nodes accessed per query is a small fraction (134.9/1553 = 8.7%) of the indexing nodes.

We compare the retrieval performance of the dominant color descriptor to the traditional color histogram descriptor. The proposed method can be seen as a variation of the histogram approach in terms of the color feature vector and the distance measure used. A 1024-D color histogram feature vector is extracted from each image region in the database. Using this histogram representation, the top 100 matches are obtained for each of the query image regions. Note that this requires 1024 numbers per region, as compared to only 14 numbers (on average) per region in the proposed method. For comparison, we computer the SVD on the quadratic matrix A as suggested in [9] and obtain a 14-D transformed color histogram vector. Again, an exhaustive search of the database is performed to find the top 100 retrievals using this SVD approach.

Before the evaluation, subjective testing is done to determine the relevant matches in the database to the query image regions. The top 100 retrievals from both the histogram and the proposed approaches are marked by five subjects to decide whether they are indeed visually similar in color. Those marked by at least three out of the five subjects are considered as relevant matches and the others are considered as false matches. Because it is impractical to go through the entire database to find all the relevant matches for the queries, the union of relevant retrievals from the two methods is used as the approximate "ground truth" to evaluate the retrieval accuracy. The retrieval accuracy is measured by precision and recall

$$Precision(K) = C_K/K \quad \text{and} \quad Recall(K) = C_K/M \quad (16)$$

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 precision and recall values for $K = 50$ are listed in Table III. The average precision and recall curves are plotted in Figs. 4 and 5. It can be seen from the table and the figures that the proposed method achieves good results in terms of the 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. Figs. 6 and 7 show two example retrievals. One query is the dark-blue and white colored snowcap on the mountain. The other is the mixed red, yellow and green colored foliage in autumn. The top six retrievals in both examples show a good match of colors. More examples and a demonstration of the region-based image search using the proposed color indexing method can be found at http://maya.ece.ucsb.edu/Netra/index2.html.

V. Conclusion

In this work, a dominant color representation for image regions is proposed. The dominant color descriptor consists of the representative colors in the region and their distribution. A similarity measure is defined for the proposed color descriptor and is shown to be equivalent to the quadratic color histogram distance measure. An efficient color indexing scheme for image retrieval using this color descriptor is presented. Experimental results show that the proposed method is fast and effective.

There are some limitations of the proposed retrieval system. One problem is that the approach is still based on low-level visual features and hence the retrieved matches do not necessarily correspond to any high-level semantics. Another limitation of the system is that it does not handle the spatial relationship between regions in the image. More work is needed to address these issues.

References

[1] G. P. Babu, B. M. Mehtre, and M. S. Kankanhalli, "Color indexing for efficient image retrieval," *Multimedia Tools Applicat.*, vol. 1, pp. 327–348, Nov. 1995.

[2] N. Beckmann, H. Kriegel, R. Schneider, and B. Seeger, "The R^*-tree, an efficient and robust access method for points and rectangles," in *Proc. ACM SIGMOD Int. Conf. Management Data*, 1990, pp. 322–331.

[3] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry Algorithms and Applications*. Berlin, Germany: Springer, 1997.

[4] G. Cha and C. Chung, "Multi-mode indices for effective image retrieval in multimedia systems," in *Proc. IEEE Multimedia Computing Systems*, 1998, pp. 152–159.

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

[6] Y. Deng, C. Kenney, M. S. Moore, and B. S. Manjunath, "Peer group filtering and perceptual color quantization," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4, 1999, pp. 21–24.

[7] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1970.

[8] A. Guttman, "R-trees: A dynamic index structure for spatial search," in *Proc. ACM SIGMOD Int. Conf. Management Data*, 1984, pp. 47–57.

[9] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 729–736, July 1995.

[10] J. Huang, S. R. Kumar, M. Mitra, W. Zhu, and R. Zabih, "Image indexing using color correlograms," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1997, pp. 762–768.

[11] ISO/IEC JTC1/SC29/WG11/N2461, "MPEG-7 requirements document,", Atlantic City, NJ, Oct. 1998.

[12] W. Y. Ma and B. S. Manjunath, "Edge flow: A framework of boundary detection and image segmentation," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, 1997, pp. 744–749.

[13] ——, "NeTra: A toolbox for navigating large image databases," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, 1997, pp. 568–571.

[14] W. Y. Ma and H. Zhang, "Benchmarking of image features for content-based retrieval," in *Proc. IEEE 32nd Asilomar Conf. Signals, Systems, Computers*, vol. 1, 1998, pp. 253–257.

[15] D. Mukherjee and S. K. Mitra, "Vector set-partitioning with successive refinement Voronoi lattice VQ for embedded wavelet image coding," in *Proc. IEEE Int. Conf. Image Processing*, 1998.

[16] G. Pass and R. Zabih, "Histogram refinement for content based image retrieval," in *Proc. IEEE Workshop Applications Computer Vision*, 1996, pp. 96–102.

[17] Y. Rubner, C. Tomasi, and L. J. Guibas, "A metric for distributions with applications to image databases," in *Proc. 6th Int. Conf. Computer Vision*, 1998, pp. 59–66.

[18] H. Samet, *The Design and Analysis of Spatial Data Structures*. Reading, MA: Addison-Wesley, 1990.
[19] J. Smith and S.-F. Chang, "Tools and techniques for color image retrieval," *Proc. SPIE*, vol. 2670, pp. 2–7, 1996.
[20] M. A. Stricker and M. Orengo, "Similarity of color images," in *Proc. SPIE Storage Retrieval Still Image Video Databases IV*, vol. 2420, 1996, pp. 381–392.
[21] M. Stricker and A. Dimai, "Color indexing with weak spatial constraints," in *Proc. SPIE Storage Retrieval Still Image Video Databases IV*, vol. 2670, 1996, pp. 29–40.
[22] M. J. Swain and D. H. Ballard, "Color indexing," *Int. J. Comput. Vis.*, vol. 7, no. 1, pp. 11–32, 1991.
[23] X. Wan and C. J. Kuo, "A multiresolution color clustering approach to image indexing and retrieval," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signals Processing*, vol. 6, 1998, pp. 3705–3708.
[24] D. A. White and R. Jain, "Similarity indexing with the SS-tree," in *Proc. Int. Conf. Data Engineering*, 1996, pp. 516–523.
[25] H. Zhang, Y. Gong, C. Y. Low, and S. W. Smoliar, "Image retrieval based on color features: An evaluation study," in *Proc. SPIE Digital Image Storage Archiving Systems*, vol. 2606, 1995, pp. 212–220.

Yining Deng (S'95–M'99) received the B.S. degree in electrical engineering from the Cooper Union for the Advancement of Science and Art, New York, in 1995, and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Santa Barbara, in 1996 and 1999, respectively.

He is currently with Hewlett-Packard Laboratories, Palo Alto, CA.

B. S. Manjunath (S'88–M'91) received the B.E. degree (with distinction) in electronics from Bangalore University, Bangalore, India, in 1985, the M.E. degree (with distinction) in systems science and automation from the Indian Institute of Science in 1987, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1991.

He joined the Electrical and Computer Engineering Department, University of California, Santa Barbara, in 1991. His current research interests include multimedia databases, digital libraries, image processing and computer vision. He is also an active participant in the ISO/MPEG-7 standardization.

Dr. Manjunath was a recipient of the national merit scholarship (1978–1985) and was awarded the university gold medal for the best graduating student in electronics engineering in 1985 from the Bangalore University. He is currently an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and was a Guest Editor for the January 2000 special issue on image and video processing for digital libraries.

Charles Kenney was born in Washington, DC, in 1950. He received the B.Sc., M.A., and Ph.D. degrees from the University of Maryland, College Park, in 1973, 1976, and 1979, respectively.

From 1979 to 1981, he taught mathematics at California State College, Bakersfield, and subsequently was a Numerical Analyst with the Naval Weapons Center, China Lake, CA. Since 1987, he has worked half his time at the Naval Weapons Center and half his time as a Research Engineer with the Electrical and Computer Engineering Department, University of California, Santa Barbara. From 1981 to 1990, his primary research was in the numerical linear algebra related to control theory, especially the solution of large Riccati problems via Pade approximation theory and the matrix sign function. His work in this area has led to efficient parallel algorithms with accurate condition estimation procedures. His work for the Navy has concentrated on differential equations associated with supersonic flow over missile bodies, shock waves in detonations, thermal ablation of rocket motor linings, electromagnetic propagation in antenna horn arrays, and high-speed payout of optical fiber for missile guidance. For the last four years, his research has been in the field of image processing using partial differential equations.

Michael S. Moore (S'97) received the B. S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1990 and the M.S. degree from the University of California, Santa Barbara, in 1996, where he is currently pursuing the Ph.D. degree in image processing.

He spent five years with the Naval Nuclear Propulsion Program, Washington, DC.

Hyundoo Shin received the B.S. degree in applied physics from Columbia University, New York, in 1983 and the M.S. and Ph.D. degrees in applied mathematics from Brown University, Providence, RI, in 1990.

From December 1990 to August 1992, he was a Postdoctoral Associate with Yale University, New Haven, CT, and from September 1992 to January 1994, was a Research Associate at Brown University. He joined Samsung Electronics as a Senior Researcher in February 1994. He is currently a Principal Research Engineer with the Digital Media System Laboratory, Samsung Electronics, Seoul, Korea.