A MULTIRESOLUTION COLOR CLUSTERING APPROACH TO IMAGE INDEXING AND RETRIEVAL

Xia Wan and C.-C. Jay Kuo

Integrated Multimedia System Center and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, California 90089-2564
Email:{xiawan,cckuo}@sipi.usc.edu

ABSTRACT

We propose a multiresolution color feature extraction scheme based on octree data structure to achieve efficient and robust image retrieval. With the proposed method, multiple color features, including the dominant color, the number of distinctive colors and the color histogram, can be naturally integrated into one framework. A selective filtering strategy is also described to speed up the retrieval process. Retrieval examples are given to illustrate the performance of the proposed approach.

1. INTRODUCTION

Effective retrieval of image data based on low-level features has received a lot of attention recently. Among them, color features such as global and local color histograms, the mean (i.e. average color) and higher order moments of the histogram, have been widely used to facilitate content based image access. There has been research to improve the discriminating power of color features. For example, the QBIC (Query by Image Content) [1] system supports color feature extraction of manually outlined objects. Evaluation study made by Zhang and Smoliar [3] showed that the fixed size local histogram is computationally simple and efficient in some applications. The method proposed by Stricker and Dimai [4] extracted color features defined in fuzzy regions adaptive to image content.

There are common issues underlying all color-based retrieval methods: the selection of a proper color space in which image colors gives the best discriminant power [5], the use of a proper color quantization scheme to reduce the color resolution, and the development of efficient feature representations to support a robust and flexible query process. In [5], we observed that the computational complexity increases quickly as the resolution of color feature increases. This can be a major problem in applications where the desired performance requires a high resolution of color features and a sophisticated color quantization schemes. We also observed that results of quantized images are very sensitive to the location of quantization boundaries. Similar colors can be quantized into two different bins, which will lead to false misses in the retrieval process.

In this work, we propose a new color feature based on multiresolution color clustering. This color feature is different from our previous work [5] in the sense that it allows natural color clustering according to the content of an image (i.e. image adaptive) rather than a fixed hierarchically structured quantization. We have also developed a set of filtering methods based on the new color feature to facilitate the retrieval process. They include: filtering by the dominant color, by the color depth, and by tree intersection. A combination of these methods allows a prompt access to images in a large image database. Thus, this new color feature extraction scheme provides a balance between the performance in terms of discriminant capability and the computational complexity.

2. MULTIRESOLUTION COLOR CLUSTERING

The target of clustering is to partition a set $\mathcal{X}$ of n samples $\mathbf{x_1}, \cdots, \mathbf{x_n}$ into c disjoint subsets. Since the number c of clusters is unknown, a series of cost functions $J^{(c=i)}$ with $i = 1, 2, 3, \cdots$ have to be calculated to determine its value. Clearly, $J^{(c=i)}$ deceases monotonically with the increment of c. If the samples are naturally grouped into $\hat{c}$ compact and well separated clusters, $J^{(c=i)}$ decreases fast when $i \leq \hat{c}$ and slowly when $i > \hat{c}$. To avoid difficulties in determining c, a multiresolution clustering method is proposed in this section. The proposed scheme consists of two stages. First, we use the octree color quantization to get the initial clustering. Then, we adopt a fast agglomerative hierarchical approach for multiresolution clustering.

Octree is used to represent the color information of

an image. The root node of the octree is related to the entire color space, the 8 children of the root corresponds to the eight subspaces of the entire space, each of the eight nodes can have its own 8 children corresponding to further divided subspaces. Every node of the octree has two attributes describing the color information of the corresponding subspace: the average color $\vec{C}$ representing the mean of pixels, and the pass-number p representing the number of pixels located in the subspace. Fig. 1 illustrate the splitting planes corresponding to the second level of the octree. In actual implementation, we do not have to perform the initialization of the octree in the L*u*v* space by comparing the value of pixels with boundaries defined by the slitting planes. Instead, the RGB representation is used when inserting a color into the octree. However, the average color $\vec{C}$ is calculated with the L*u*v* representation in order to calculate the color distances accurately.

The octree insertion and node merging process are illustrated in Fig. 2 and Fig. 3, respectively. The clustering procedure is summarized as follows.

1. Octree initialization and shrinking

(a) Insert a new pixel in the image from the root to the leaf and update the pass number and the average color of nodes accordingly based on its RGB color representation. Repeat the process until all pixels are inserted.

(b) Get the link of the parents of all leaf nodes, find the node whose pass number is minimum, reduce the children of this node, modify the link. Repeat this step until the number of leaf nodes is less or equal to C ($C = 256$).

2. Multiresolution clustering

(a) Set the average color of the node whose lightness is less than L_0 ($L_0 = 50$) to black.

(b) Perform the following steps for $k = 1, 2, 3$. Find the nearest pair of distinct clusters based on their means. If their distance is less than T_k, merge them. This process is repeated until the distance of all distinct pairs is greater than T_k. $T_1 = 16$, $T_2 = 24$ and $T_3 = 32$ is used in our implementation.

It is worthwhile to mention that we use a divide and conquer method based on the octree structure in finding the nearest pair of clusters to reduce the computational complexity from $O(N^2)$ to $O(N \log N)$, where N is the number of leaf nodes.

3. INDEXING AND RETRIEVAL BASED ON MULTIRESOLUTION CLUSTERING

3.1. Indexable Features and Distance Computation

We can derive a set of interesting features based on the multiresolution color clustering feature to speed up the retrieval. They include the following.

Average color: (stored with 3 bytes)

The average color of the entire image corresponds to that of the root of the octree. The distance between the average colors of the query and target images can be computed via

$$d_{avg}(Q, T) = d_{Euclidean}(\vec{C}_{0,0}^{(Q)}, \vec{C}_{0,0}^{(T)}),$$

Dominant color: (stored with 3 bytes)

The average color of the node with the largest pass number at the coarsest resolution gives the dominant color of an image. The distance between the dominant colors of the query and target images can be computed via

$$d_{dom}(Q, T) = d_{Euclidean}(\vec{C}_{k,m}^{(Q)}, \vec{C}_{k,n}^{(T)}), \quad k = 3,$$

where $\vec{C}_{k,m}^{(Q)}$ and $\vec{C}_{k,n}^{(T)}$ are average colors of dominant nodes of images Q and T at the coarsest resolution, which is 3 in our implementation, respectively.

Color width: (stored with 1 byte)

The number of leaf nodes at the finest resolution (i.e. $k = 1$) is called the color width of a given image. It indicates the richness of colors. The distance based on the color width is defined as:

$$d_{width}(Q, T) = |W^{(Q)} - W^{(T)}|,$$

where $W^{(Q)}$, $W^{(T)}$ are widths of octrees of images Q and T.

Multiresolution color distributions (stored on the average with 124 bytes per image in our test database)

The average color and the pass number of leaf nodes at each resolution lead to a set of multiresolution color distributions. Since the number of clusters and the position of clusters are different from images to images, we have to define the distance between two sets of clustered nodes. Let k represent the resolution level, and $\mathcal{Q}$ and $\mathcal{T}$ denote, respectively, node sets for query image Q and target image T at level k. Nodes from $\mathcal{Q}$ and $\mathcal{T}$ are said to have a match if their distance is less than T_k. Let $\mathcal{M}$ be the set of all matched nodes in $\mathcal{Q}$ and $\mathcal{T}$

at resolution k. Then, the distance can be defined as:

$$d_{Layer(k)}(Q,T) = \sum_{\mathcal{M}_i \in \mathcal{M}} \left(\sum_{m \in \mathcal{M}_i} p_m^{(Q)} - \sum_{\hat{m} \in \mathcal{M}_i} p_{\hat{m}}^{(T)} \right)^2 + \sum_{u \in \mathcal{Q}-\mathcal{M}} p_u^{(Q)} + \sum_{u \in \mathcal{T}-\mathcal{M}} p_u^{(T)}, \quad (2)$$

where $p_m^{(Q)}$ and $p_{\hat{m}}^{(T)}$ are the normalized pass-numbers of nodes m and $\hat{m}$ in the same matched set $\mathcal{M}_i$ and belonging to $\mathcal{Q}$ and $\mathcal{T}$, respectively, and $p_u^{(Q)}$ and $p_u^{(T)}$ are the normalized pass-numbers of nodes u in unmatched sets $\mathcal{Q} - \mathcal{M}$ and $\mathcal{T} - \mathcal{M}$, respectively.

As described above, we need 131 (=3+3+1+124) bytes in total to store the average color, the dominant color, the color width and the multiresolution color distribution for each image in the database. This is about one half of the storage required by the traditional histogram method with 256 quantization bins where 1 byte is used to record the normalized pixel numbers in each bin.

3.2. Retrieval Examples

Each indexing feature mentioned above carries interesting color information of an image. Filtering by a selected set of simple features such as the average color, the dominant color and the color width can be performed first to remove irrelevant images. This is particularly useful, if the query image has a certain prominent features, e.g. a clear dominant color and an unusual color width. Filtering based on the comparison of multiresolution color distributions can be performed at a later stage to refine the candidate image set which contains similar images.

We use several examples below to demonstrate this idea. Our experimental database consists of 2119 images, including natural scenes, animals, plants, architectures and people. Large varieties of our image database prevent the bias on a particular type of images. We consider three image sets, i.e."Skiing", "Stained-glasses", and "sunset' and use one from each image set as the query image.

Retrieval of "Skiing" image

Each image in the "Skiing" image set is dominated by the white tone. The dominant color and the percentage of pixels possessing this color is shown in Table 1. Retrieval by the dominant color alone can promptly get a very small candidate image set.

Retrieval of "Stained-glasses" image

The color width of the query image is 71, which is very large in comparison with most images in the database. The distribution of the color width of images in our test database at the each resolution of clustering can be found in Fig. 4. As seen from the figure, only a small number of images have a large width. Thus, filtering by the color width helps to narrow down the number of candidate images quickly. Filtering by the color width is ideal for images with rich or few distinctive colors.

Retrieval of "Sunset" image

Each image in this set has a dominant color, even though their dominant colors may not be very similar. For example, some images are dominated by dark red while other images are dominant by dark yellow. Thus, the threshold for filtering with the dominant color has to be set a larger value to avoid false misses and, as a result, the candidate image set is still large. Filtering by multiresolution color distributions can be applied to this set of candidate images. Five images within the query image set are retrieved in the list of top eight candidate images filtering with the finest resolution feature. These five images are ranked among the top 77 and 19 images (out of 2119 images) which are similar to the query image with the coarsest and middle resolution features, respectively. Thus, we can perform the distance computation in a smaller image set as the resolution increases. As a result, filtering by multiresolution color features can speed up the retrieval process.

4. CONCLUSIONS

We proposed a new multiresolution color extraction scheme based on octree data structure. Color feature obtained by our scheme is more efficient than the color histogram in several aspects. First, it calculates the color feature of each individual image separately, and only a small number of distinctive colors and their corresponding pass-numbers are used to describe the color feature of the image. Consequently, the color feature of each image is described more effectively with a smaller storage space. Second, there is no rigid quantization boundaries in quantizing similar colors so that we can get a more robust retrieval result with respect to small color differences among images. Third, more color features such as color width, dominant color, average color can be obtained as the byproduct. The retrieval process can be speeded up by combining these features properly.

5. REFERENCES

[1] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient color histogram indexing for quadratic form distance functions," *IEEE trans. on Pattern Recognition and Machine Intelligence*, vol. 17, pp. 729–736, July 1995.

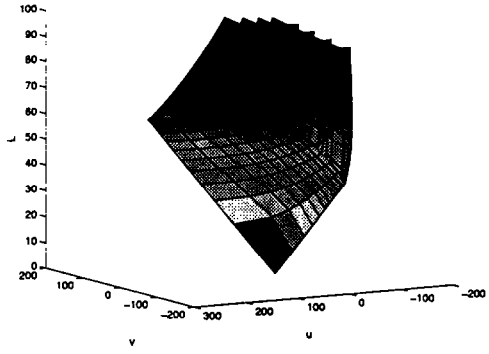


Figure 1: Splitting of the L*u*v* space.

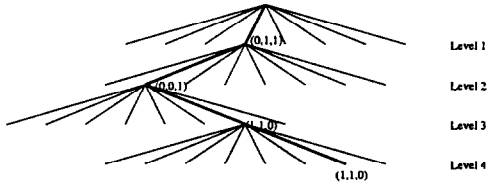


Figure 2: An example of inserting a color point with $R = 53$, $G = 187$, $B = 197$ into the octree.

[2] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[3] H. Zhang, C. L. Y. Gong, and S. Smolia, "Image retrieval based on color features: an evaluation study," in *SPIE Digital Image Storage and Archiving Systems*, vol. SPIE 2606, pp. 212–220, Oct 1995.

[4] M. Stricker, "Color indexing with weak spatial constraints," in *SPIE Storage and Retrieval for Image and Video Databases IV*, vol. 2670, pp. 29–40, Feb. 1996.

[5] X. Wan and C.-C. Kuo, "Color distribution analysis and quantization for image retrieval," in *SPIE Storage and Retrieval for Image and Video Databases IV*, vol. SPIE 2670, pp. 9–16, Feb 1996.

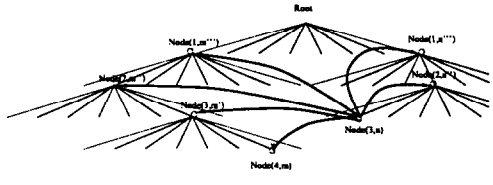


Figure 3: Illustration of the node merging process.

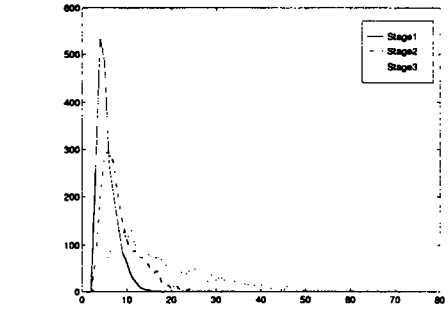
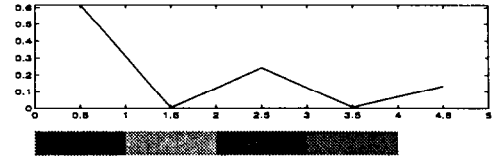
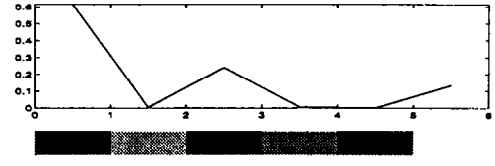


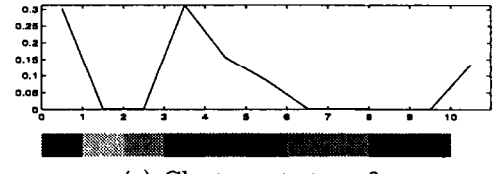
Figure 4: Distribution of the number of clusters at different resolutions.



(a) Clusters at stage 1.



(b) Clusters at stage 2.



(c) Clusters at stage 3.

Figure 5: Color clustering of "Sunset" image with three different resolutions.

Images	Dominant color (Luv)	Pass-Number
Ski_0	(86,-18,-19)	0.9135
Ski_1	(96, -9, -3)	0.9077
Ski_2	(92,-11, -8)	0.8667
Ski_3	(91,-14, -7)	0.9490
Ski_4	(89, -10, -4)	0.9334
Ski_5	(98, -8, -5)	0.8999
Ski_6	(87,-14,-12)	0.9431
Ski_7	(87,-22,-25)	0.9121
Ski_8	(91, -10, -9)	0.9283
Ski_9	(86, -10,-8)	0.8376

Table 1: The dominant color of images in "Skiing" set.