Spatial Color Component Matching of Images

Jianying Hu
Avaya Labs Research
233 Mount Airy Road, Basking Ridge, NJ
jianhu@avaya.com

Efstathios Hadjidemetriou
Computer Science Department
Columbia University, NY
stathis@cs.columbia.edu

Abstract

Color and color neighborhood statistics have been used extensively in image matching and retrieval. However, the effective incorporation of color layout information remains a challenging issue. In this paper we present a novel method for color layout based image matching called Spatial Color Component Matching *(SCCM). First perceptually dominant colors are extracted from an image and are back-projected to segment the image into various areas. Then, each dominant color area, depending on its size, is segmented into a number of spatial units using a multilevel graph partitioning algorithm. Each unit is described in terms of its color and a set of spatial attributes to form a* Spatial Color Component *(SCC). All SCC's form a list that summarizes the color layout information in an image. The distance between two images is then defined by the minimum distance mapping between the two corresponding SCC lists. The algorithm has been evaluated using an image database of wall paper patterns and another database of natural images. It has been judged by human subjects to be highly effective in both cases.*

1. Introduction

Color feature is one of the most widely used visual features in image matching and retrieval. Techniques for color based image matching can be divided into two categories: (1) global color distributions and (2) color layout.

The global color distribution features are relatively simple to calculate and can provide reasonable results in many applications [17, 12, 5]. However, since such techniques do not take into consideration any spatial features, it has limited discriminative power. As a result, much research has been carried out to incorporate the layout of colors, which lead to the second category of techniques. One popular group of techniques within this category can be characterized as augmented or extended histograms. Examples include color correlogram [6], and augmented color histograms [2]. These techniques, however, all rely on pixel level statistics, thus can not capture regional features such as "a green region to the right of a red region".

An alternative to the histogram based techniques are the region based methods. One approach is to divide the whole image into fixed sub-blocks and extract color features from each of the sub-blocks [16, 18]. The drawback of fixed partitioning based methods is that they do not take into account the natural segmentation of an image, thus finer features within a sub-block are lost. Alternative approaches have been developed which segment images into regions based on color or texture, then compute both color and spatial attributes for each region [14, 1]. However these methods were designed to facilitate region or object based queries in image databases. They are not suitable for comparing whole images based on color layout. Another class of algorithms make use of models that encode relative relations among sub-blocks to represent images of various layout types [9, 15]. These algorithms were designed to classify images into a set of *pre-defined* classes, thus can not be easily adapted to compare arbitrary images.

In this paper we propose a new method for comparing arbitrary images based on color layout called *Spatial Color Component Matching* (SCCM). For each image, first *perceptually dominant colors* are extracted and back-projection is used to segment the image into *dominant color areas*. Each dominant color area is then spatially divided into a number of subregions using a graph partitioning algorithm. The color along with a group of spatial attributes are used to describe these subregions called *spatial color components* (SCC). The distance between any two images is then computed by finding the minimum distance between the corresponding SCC lists.

2. Dominant Color Area Extraction

We first transform a color image from the RGB space into the L^*a^*b color space to take advantage of the perceptually more uniform nature of the L^*a^*b space [19] . The set of all possible colors are then quantized to a subset defined by a compact color codebook [10].

In the next step, a statistical method is used to identify and remove colors of speckle noise. The method is based on the observation that human beings tend to ignore isolated spots of a different color that are randomly distributed among a consistent color. Such isolated colors are identified through the computation of *Neighborhood Color Histogram Matrices* [5] and then mapped to the surrounding consistent color. The colors are then sorted in order of decreasing area percentage it occupies in the image. A color is removed if it occupies less then 2% of the image and its ranking in the sorted list is larger than 3. At this point all remaining colors are considered dominant colors in the image. A pixel with non-dominant color is assigned the closest dominant color.

After dominant color extraction, a back-projection procedure is used to segment the image into *dominant color areas*. The procedure simply assigns all pixels with the same color to the same area. Note that a dominant color area differs from a conventional image region in that it is not necessary connected. Figure 1 (a) and (b) show an image and its dominant color areas.

(a) (b) (c)

Figure 1. (a): An image; (b): Its dominant color areas; (c): The six SCC's computed from the largest dominant color area as explained in Section 3, shown in pseudo-color.

3. Spatial Color Components

While the set of dominant color areas provide a compact representation of the main color layout characteristics of an image, they are cumbersome to compare directly. The reason is that although each area has a uniform color, the spatial characteristics such as size, location, and spatial distribution can vary widely within an area, making it difficult to match different color areas. Our solution to this problem is to further partition each dominant color area into subregions that are equal-size in terms of area percentage. The number of subregions assigned to each area is proportional to the total size of the area. As a result, for each image we obtain a fixed number of equal-size subregions called *spatial color components* (SCC). Each SCC is then characterize by its color as well as its spatial attributes. The distance between two images can than be computed efficiently by matching the two corresponding lists of SCC's.

Clearly, the crucial step in this approach is the partitioning algorithm. Intuitively, the partitioning should be guided by the layout of the pixels in the area and result in spatially "tight" clusters. More formally, the partitioning needs to satisfy the following criteria: 1) the subregions are as close to equal-size as possible; 2) it results in least possible amount of disconnectedness within a subregion; and 3) it maximizes cross region distances. This poses a complex combinatorial optimization problem, which can be formulated as a k-way graph partitioning problem.

The k-way graph partitioning problem is the problem of partitioning the vertices of a graph into k roughly equal parts, such that the total weight associated with edges connecting vertices in different parts is minimized [8]. Although the problem itself is NP-complete, much research has been carried out to find approximate solutions that can be computed efficiently. One of the latest set of algorithms, the METIS package developed by Karypis *et al.* [7], achieves a run time linear to the number of edges in the graph using a novel multilevel recursive partitioning scheme. METIS also provides support for eliminating spurious non-contiguous subregions, which makes it ideal for our application. In the following we explain how METIS is applied to solve our partitioning problem.

Suppose we have a dominant color area which is to be partitioned into k SCC's. First a sub-sampling procedure is performed (when necessary) to bring the number of pixels in the area to below a predetermined value (e.g., 1000). Let the resulting number of pixels in the area be n, we create a weighted graph $\mathcal{G}\{\mathcal{V}, \mathcal{E}\}$, where the set of vertices $\mathcal{V} = \{v_1, ..., v_n\}$ contains one vertex for each pixel in the area and the set of edges $\mathcal{E} = \{e_{ij}, i = 1, ..., n; j = i+1, ..., n\}$ contains one edge for every pair of vertices. The weight w_{ij} assigned to edge e_{ij} is then defined as: $w_{ij} = D_{max} - d_{ij}$, where d_{ij} is the Euclidean distance between the pixels represented by vertices v_i and v_j, and D_{max} is the maximum pixel-wise distance within the area. The subroutine pmetis from the METIS package is then applied to this weighted graph and the resulting k partitions form the spatial color components. The computation time for the whole partitioning process is linear to the number of pixels in the area after sub-sampling.

Figure 1(c) illustrates the six SCC's extracted from the largest dominant color area shown in Figure 1(b), in pseudo-color.

4 Distance Computation

In order to compare two images represented as lists of SCC's, we first need to define the distance between two SCC's. Clearly, both color and spatial features should be considered in defining such distance. Since each SCC is uniform in color, the color feature is simply the color of the component in L^*a^*b form. In selecting the spatial features, the goal is to compose a small set of features that captures the most important spatial characteristics of a set of pixels. We have arrived at four spatial features: the centroid, which indicates the location of the SCC; the minimum moment of inertia and its orientation, which together indicate the mass distribution of the SCC; and the average size of connected components within the SCC, which is a measure of the "scatteredness" of the SCC.

Let S_u and S_v represent two different SCC's, the following are definitions of the features and the corresponding distances.

- Color $C_u(L_u, a_u, b_u)$: color coordinates in L^*a^*b space. Distance: $D_c(u,v) = \sqrt{(L_u - L_v)^2 + (a_u - a_v)^2 + (b_u - b_v)^2}$.
- Centroid Position $P_u(\bar{x}_u, \bar{y}_u)$: horizontal and vertical coordinates for the centroid. Distance: $D_p(u,v) = \sqrt{(\bar{x}_u - \bar{x}_v)^2 + (\bar{y}_u - \bar{y}_v)^2}$.
- Minimum Moment of Inertia M_u, as defined in [4]. Distance: $D_m(u,v) = |M_u - M_v|$.
- Orientation of minimum moment of inertia A_u, as defined in [4]. Distance: $D_a(u,v) = min\{|A_u - A_v|, \pi - |A_u - A_v|\}$.
- Average region size S_u: average size of connected components within the SCC. Distance: $D_s(u,v) = |S_u - S_v|$.

In order to combine the above distances, each distance is first normalized by its mean computed over a large set of images. Denote the normalized distance with $\tilde{D}$, the overall distance between two spatial color components S_u and S_v is defined as:

$$D(u,v) = \sum_{\alpha \in \{c,p,m,a,s\}} w_\alpha \tilde{D}_\alpha(u,v)$$

where α is the suffix indicating the feature type and w_α is the weight assigned to the corresponding feature. Currently the weights are determined empirically and are assigned the following values: $w_c = 0.8; w_p = 0.1; w_m = 0.02; w_a = 0.04; w_s = 0.04$. Not surprisingly, w_c is by far the largest weight, reinforcing the observation from previous subjective experiments indicating that color is the most important factor in judging image similarity [11].

Given the distance between two SCC's described above, the distance between two images is then defined as following. First a fixed number, N, SCC's are computed for each image. Then the optimal mapping between the SCC lists is obtained. The optimal mapping between two SCC lists of equal length is defined as the one-to-one mapping that minimizes the total pair-wise distance. It can be computed efficiently using a publicly available implementation [13] of Gabow's weighted graph matching algorithm with $O(N^3)$ complexity[3]. The sum of distances defined by this optimal mapping is then taken as the distance between the two images.

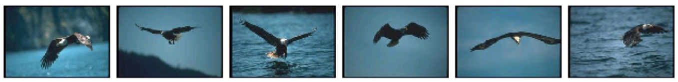
5. Experimental Results

We tested our algorithm on two databases and compared it to the widely used color correlogram method [6]. One of our databases contains 884 stock photo images of size 192×128 from the Corel Photo Collection, the other contains 336 images of scanned wall paper samples of size 200×153. The Corel database contains images from 12 different categories (*airshow, bald eagle, cheetah, desert, elephant, field, firework, mountain, polar bear, sailboat, sunrise, and tiger*), each containing different number of images. In order to avoid bias towards any particular category, 24 images were randomly selected from each category to be query images, resulting in a total of 288 query images. The wall paper database is not divided into specific categories thus all 336 of its images are used as query images. For each query image, similar images were retrieved from within the database using both color correlogram and the proposed SCCM method and the top matching images were saved. A 99 color codebook for the L^*a^*b space [10] was used in both methods. For the SCCM method, 20 SCC's were computed for each image.

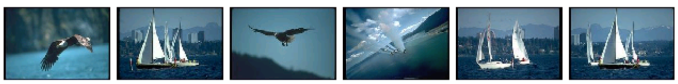
A web based graphical user interface has been constructed to compare the results from the two algorithms through subjective experiments. In each experiment, a query image is randomly selected from all pre-computed query images, then the retrieval results using two different algorithms are presented on the screen in two separate horizontal lists, each ordered by decreasing similarity. The user is asked to select the list that better matches his/her own judgment of similarity. A "NONE" option can be chosen when the user feels that there is no clear winner between the two. This process is repeated until the user has made 5 valid (i.e., other than "NONE") choices within each database. In the interface each list is only labeled as "List 1" or "List 2" and the order of the two algorithms is randomized for each query image so that no bias is given to either one.

This GUI was used to compare the proposed SCCM method to the color correlogram method [6]. A total of 12 independent subjects (6 man, 6 women) participated in the experiments. All but one of the subjects favored SCCM more in the 10 choices made, the remaining one subject had a 5-5 split between the two algorithms. Overall, out of the $12 \times 10 = 120$ valid choices made, SCCM was perceived to be better in 74% of the cases, which clearly demonstrates the effectiveness of the SCCM method.

Figure 2 shows a query example from the Corel database. The top list shows the retrieved images using SCCM, the bottom list shows the retrieved images using the color correlogram method. Within each list the left most image is the query image and the top 5 matching images are displayed in order of decreasing similarity to the query image from left to right. Figure 3 shows a query example from the wall paper database, in the same format.



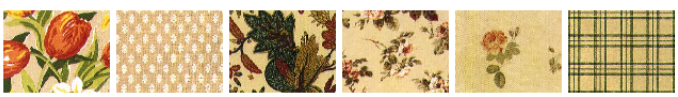
(a)



(b)

Figure 2. Retrieval results for a query image from the Corel database. (a): SCCM; (b): Color correlogram.

(a)



(b)

Figure 3. Retrieval results for a query image from the wall paper database. (a): SCCM; (b): Color correlogram.

6. Summary

We presented a new method for comparing pairs of images based on color layout, called Spatial Color Component Matching (SCCM). An image-based spatial partitioning scheme is used to represent an image as a list of spatial color components (SCC). The distance between two images is computed by first finding the minimum-distance mapping between the corresponding SCC lists and then summing up the the distances between the resulting SCC pairs. Subjective experiments using both artificial and natural images demonstrated the effectiveness of this approach. Future work will be directed towards developing indexing schemes based on this new distance metric, and investigating other possible spatial descriptors such as texture measures.

References

[1] S. Belongis, C. Carson, H. Greenspan, and J. Malik. Color- and texture- based image segmentation using EM and its application to content-based image retrieval. In *ICCV'98*, Bombay, India, Jan. 1998.

[2] Y. Chen and E. Wong. Augmented image histogram for image and video similarity search. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 523–532, San Jose, Jan. 1999.

[3] H. Gabow. *Implementation of algorithms for maximum matching on nonbipartite graphs*. PhD thesis, Stanford University, 1973.

[4] B. Horn. *Robot Vision*. McGraw-Hill, New York, 1986.

[5] J. Hu and A. Mojsolovic. Optimal color composition matching of images. In *15th ICPR*, volume 4, pages 47–51, Barcelona, Spain, Sept. 2000.

[6] J. Huang, S. Kumar, M. Mitra, W. Zhu, and R. Zabih. Image indexing using color correlogram. In *CVPR'97*, San Juan, Puerto Rico, June 1997.

[7] G. Karypis. Metis: Family of multilevel partitioning algorithms. *http://www-users.cs.umn.edu/ karypis/metis*.

[8] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1), 1998.

[9] P. Lipson, E. Grimson, and P. Sinha. Configuration based scene classification and image indexing. In *CVPR'97*, pages 1007–1013, Puerto Rico, June 1997.

[10] A. Mojsolovic and J. Hu. Extraction of perceptually important colors and similarity measurement for image matching. In *ICIP2000*, Vancouver, Canada, Sept. 2000.

[11] A. Mojsolovic, J. Kovacevic, J. Hu, R. J. Safranek, and K. Ganapathy. Matching and retrieval based on the vocabulary and grammar of color patterns. *IEEE Trans. Image Processing*, 9(1):38–54, Jan. 2000.

[12] W. Niblack, R. Berber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, and P. Yander. The QBIC project: Quering image by content using color, texture and shape. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 172–187, San Jose, Jan. 1994.

[13] E. Rothberg. Solver for the maximum weight matching problem. *ftp://dimacs.rutgers.edu/pub/netflow/matching/weighted/solver-1*.

[14] J. Smith and S. Chang. Tools and techniques for color image retrieval. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 1630–1639, San Jose, 1996.

[15] J. Smith and S. Chang. Multi-stage classification of images from features and related text. In *4th DELOS Workshop*, Pisa, Italy, August 1997.

[16] M. Stricker and A. Dimai. Color indexing with weak spatial constraints. In *SPIE Storage and Retrieval for Image and Video Databases*, pages 29–39, San Jose, 1996.

[17] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[18] Y. Tao and W. Grosky. Spatial color indexing: a novel approach for content-based image retrieval. In *ICMCS*, volume 1, pages 530–535, 1999.

[19] G. Wyszecki and W. S. Stiles. *Color science: concepts and methods, quantitative data and formulae*. John Wiley and Sons, New York, 1982.