Spectrally Layered Color Indexing

Guoping Qiu[1,2] and Kin-Man Lam[2]

[1] School of Computer Science, The University of Nottingham,
qiu@cs.nott.ac.uk
[2] Center for Multimedia Signal Processing, The Hong Kong Polytechnic University
enkmlam@eie.polyu.edu.hk

Abstract. Image patches of different roughness are likely to have different perceptual significance. In this paper, we introduce a method, which separates an image into layers, each of which retains only pixels in areas with similar spectral distribution characteristics. Simple color indexing is used to index the layers individually. By indexing the layers separately, we are implicitly associating the indices with perceptual and physical meanings thus greatly enhancing the power of color indexing while keeping its simplicity and elegance. Experimental results are presented to demonstrate the effectiveness of the method.

1 Introduction

An effective, efficient, and suitable representation is the key starting point to building image processing and computer vision systems. In many ways, the success or failure of an algorithm depends greatly on an appropriately designed representation. In the computer vision community, it is a common practice to classify representation schemes as low-level, intermediate-level and high level. Low-level deals with pixel level features, high level deals with abstract concepts and intermediate level deals with something in between. Whilst low level vision is fairly well studied and we have a good understanding at this level, mid and high level concepts are very difficult to grasp, certainly extremely difficult to represent using computer bits. In the signal processing community, an image can be represented in the time/spatial domain and in the frequency/spectral domain. In contrast to many vision approaches, signal processing is more deeply rooted in mathematical analysis. Both time domain and frequency domain analysis technologies are very well developed, see for example many excellent textbooks in this area, e.g., [8]. A signal/image can be represented as time sequence or transform coefficients of various types, Fourier, Wavelet, Gabor, KLT etc. These coefficients often provide a convenient way to interpreting and exploiting the physical properties of the original signal. Exploiting well-established signal analysis technology to represent and interpret vision concepts could be a fertile area for making progress.

Content-based image and video indexing and retrieval have been a popular research subject in many fields related to computer science for over a decade [1]. Of all the challenging issues associated with the indexing and retrieval tasks, "retrieval relevance" [7] is probably most difficult to achieve. The difficulties can be explained

in a number of aspects. Firstly, relevance is a high level concept and is therefore difficult to describe numerically/using computer bits. Secondly, traditional indexing approaches mostly extract low-level features in a low-level fashion and it is therefore difficult to represent relevance using low-level features. Because low-level features can bear no correlation with high level concepts, the burden has to be on high-level retrieval strategies, which is again hard. One way to improve the situation is to develop numerical representations (low-level features) that not only have clear physical meanings but also can be related to high level perceptual concepts. Importantly as well, the representations have to be simple, easily to compute and efficient to store.

There is apparent evidence to suggest that human vision system consist of frequency sensitive channels [6]. In other words, when we see the visual world, we perform some forms of frequency analysis among many other complicated and not yet understood processing. Following the frequency analysis argument, it can be understood, that when a subject is presented an image in front of her, she will "decompose" the image into various frequency components and processing each component with different processing channels (presumably in a parallel fashion). It is convenient to view such a process as decomposing the image into different layers, each layer consists of an image the same size as the original one, but only a certain frequency components are retained in each layer, i.e., a band-pass filtered version of the original image. On each layer, only those grid positions where the pixels has a certain "busyness" will have values other grid positions will be empty. It is to be noted that the notion of layered representation was used in [9] as well, but [9] dealt with motion and high level concept and is not related to what we are proposing in this paper.

By decomposing an image into spectrally separated layers, we have applied the concept to the development of simple indexing features for content-based image indexing and retrieval (simplicity and effectiveness is an important consideration in this paper). The organization in the rest of the paper is as follows. In section 2, we present the idea. Section 3 presents an algorithm. Section 4 presents experimental results and section 5 concludes our presentation.

2 The Idea

We are interested in developing efficient and effective image indexing features for content-based image indexing and retrieval. Ideally, the indexing features should be chosen in such a way that simple retrieval methods, such as computing simple metric distance measures in the feature space will produce good results. It is also well known that simple low-level features will bear little correlation with perceptual similarities if only simplistic distance measures are used. Our basic idea is to associate simple low-level features, such as color, with perceptual and physical meanings.

Color is an effective cue for indexing [3] which is well know. Because of its simplicity and effectiveness, it is attractive feature. However, it is also well know simple usage of color, i.e., color histogram, has significant drawbacks [1]. Researchers have tried various ideas of combining color with other features for indexing, see e.g., [1]. Our idea is to group colors according to their associated physical and perceptual properties.

The well-known opponent color theory [2] suggests that there are three visual pathways in the human color vision system and the spatial sharpness of a color image depends mainly on the sharpness of the light dark component of the images and very little on the structure of the opponent-color image components. In terms of perceptual significance of an image, the sharpness or roughnessof an image region determines its perceptual importance. In other words, if two areas of an image contain the same color, then the difference/or similarity of the regions are separated/identified by their spatial busyness.

Digital signal processing researchers have developed a wealth of technologies to analyze physical phenomena such as sharpness/roughnessof a signal/image. The most effective way is frequency analysis, technologies ranging from FIR filter to filter banks are well studied [8]. A busy/sharp area is associated with higher frequency components, and a flat area has lower frequency distributions. A busy area may be associated with textured surfaces or object boundaries, a flat area may be associated with backgrounds or interior of an objects. Therefore a red color in a flat area may signify a red background or large red objects with flat surface, and a similar red color in a busy area may be indications of red colored textured surface or red object boundaries. Based on these observations and reasoning, we propose a spectrally layered approach to image indexing. A schematic is illustrated in Fig. 1.

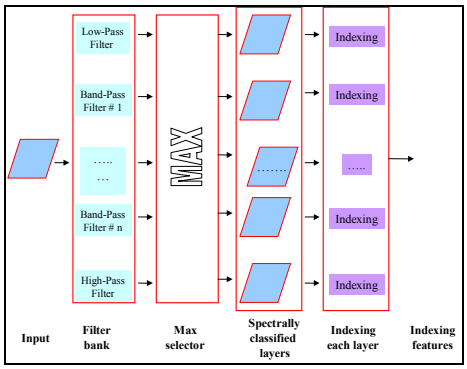

Fig. 1. Schematic of spectrally layered image indexing

Let $\boldsymbol{x}$ be the input image array, $\boldsymbol{h_k}$ be the impulse response of a band-pass filter (including the low-pass and high pass filters). Then the output of each band pass filter is

$$y_k(i,j) = x(i-l, j-m) * h_k(l,m) \tag{1}$$

Where * denotes convolution. For each pixel position, the MAX selector will identify the filter that produces the largest output, which is used to form the spectrally classified images. Let L_k be the kth layer image corresponding to the kth filter, then

$$L_k(i,j) = \begin{cases} x(i,j), \text{ if } y_k(i,j) = MAX\left(y_1(i,j), y_2(i,j), \cdots, y_n(i,j)\right) \\ Empty, \quad \text{Othewise} \end{cases} \tag{2}$$

Indexing is then performed on each layer to obtain the indexing feature vector, $\boldsymbol{I_f}$, for the image

$$I_f = \{I_f(L_1), I_f(L_2), \cdots, I_f(L_n)\} \tag{3}$$

To summarize therefore, an image is first passed through a filter bank (each filter of the filter bank covers a specified spectral bandwidth). The output of the filter bank is used to classify the pixels. Pixels in an area with similar spatial frequencies are then retained on the same layer. In each layer, which contains only those pixels in areas with similar frequency distributions, is used to form it's own index. The aggregation of the feature indices from all the layers then forms the overall index of the image. In this way, we have effectively classified the images according to the frequency contents of the image areas and indexing them separately. When such a strategy is used to match two images, the features from areas of similar spatial roughness are matched. That is, we will be matching flat area features in one image to the features in the flat areas of another image, and similarly, busy area features will be mapped to busy area features. When simple image features such as color is used for indexing such strategy should work very effectively, and we introduce an implementation in the next section.

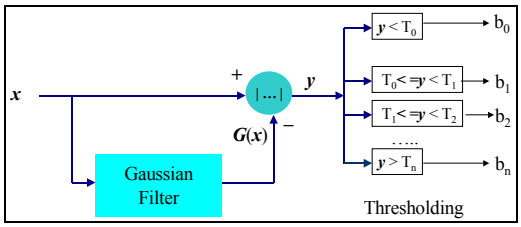

Fig. 2. A simple layer classification method.

3 An Algorithm

One of the consideration factors in our current work is efficiency. Although it is possible to introduce various features to index each layer, we present a simple and yet effective algorithm for the implementation of the scheme of the last section. The task involved two aspects. The first is how to implement the filter bank scheme, the second is how to index each layer.

Filter bank is a well-studied area in image processing. However, we realize that the spectra classification can be implemented in a variety of ways. The essence is to classify pixels in area with similar frequency characteristic into the same layer. We here present a simple non-filter bank based spectral classification method. This is illustrated in Fig. 2. Notice that only the ***Y*** component of the image is used in the classification process. This is because the sharpness of the image is mostly contained in this component. An image ***x*** is low-pass filtered first by a Gaussian kernel. This low-passed version is then subtracted from the original image. The difference is then rectified (i.e., taking the absolute values). Multiple thresholds are then applied. The binary images are obtained as (4) and the layers are formed as in (5)

$$b_k(i,j)=\begin{cases}1, & \text{if } T_{k-1} <= \left|x(i,j)-G(x(i,j))\right| < T_k \\ 0, & otherwise\end{cases} \quad (4)$$

$$L_k(i,j)=\begin{cases} x(i,j), & \text{if } b_k(i,j)=1 \\ Empty, & \text{Othewise}\end{cases} \quad (5)$$

Clearly, **y** is a Laplacian image. The magnitude of the image indicates the sharpness of the image area. A busy area, which contains many sharp changes will result in large differences between the original image and the Gaussian smoothed image. A flat area will result in small difference between the original and the Gaussian image. Therefore the pixel magnitude in the Laplacian is an indication of the roughnessof the image area sounding the pixel. The scheme therefore effectively classifies pixels in areas with similar roughnessinto the same layer.

Once we have classified pixels into different layers, we can use the simple color histogram [3] to index each layer. Let $\boldsymbol{H_k}$ be the histogram of the $\boldsymbol{L_k}$, then the indexing feature of image $\boldsymbol{x}$ is expressed in (6) and we call (6) layered color indexing (***LCI***).

$$H(x)=\{H_k\}, k=0,1,...n \quad (6)$$

Fig. 3. Top left original image. Top right Laplacian. Bottom row, layers form with different thresholding. The black pixels are empty area of the layer.

An explanation of (6) is in order. Instead of building one histogram, we are building multiple color histograms for an image, each taking colors from pixels in areas with similar sharpness. It is therefore immediate that (6) will be more powerful than the original simple color histogram. Because we not only index the color, but also the colors are associated with their surface roughness. Fig. 3 shows an example image. It is seen here the pixels of the Tiger's skin actually have similar roughness. Therefore such representation will allow not only matching the entire contents of the image, but also allow different weightings being given to different layers depending on the user's requirements. In this particular example, if a user is more interested in retrieving tigers, then more weighting can be given to the features of the last layer.

Some reader may be concerned about the complexity of our method. It is in fact very simple. The filtering and classification (thresholding) can be done very fast. The dimension of the color histogram does not have to be high. We implemented our

experiments with 4-layer and 64-color LCI (256-dimension histogram, which has similar complexity to other state of the art methods, e.g., color correlogram [3]) and we have observed very good performances. We report detail experiments in the next section.

4 Experimental Results

We used a subset of 5000 images from the Corel color photo collection in our experiment. To build the database, we used 4 layers and a 64-color quantizer to build the layered color indexing. Each image was therefore represented by a 256-cell histogram. The Gaussian filter of Burt and Aldenson [5] was used (the coefficients were: 0.05, 0.25, 0.4, 0.25, 0.05). The threshold values were chosen empirically, however, we observed that the performance was not very sensitive to small variations of threshold values. We first did a rough statistics on the histograms of a few hundreds of Laplacian images. We found the values are mostly concentrated in 0 – 18 interval. The thresholds used in the results below were y < 6, 6 <= y < 12, 12 <= y < 18, y > 18. To compare the similarity of two histograms, a L_1-norm relative distance metric [4] was used, which is defined as: Let $\boldsymbol{X} = (x_1, x_2, \ldots x_n)$, $\boldsymbol{Y} = (y_1, y_2, \ldots y_n)$, then the L_1-norm relative distance between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$D(X,Y) = \sum_i \frac{|x_i - y_i|}{1 + x_i + y_i} \tag{7}$$

We performed experiments to evaluate the retrieval precision and recall performance. As a comparison, we have also implemented color correlogram (CC) which has a D = {1, 3, 5, 7}. The same 64 colors were used in all experiments. Therefore the complexity of the two methods was the same and the colors used were exactly the same.

4.1 Retrieval Precision

In this experiment, we collected 324 query images, each had a unique answer. The results of our method and that of the color correlogram's are shown in table 1. Based on the same color and the same complexity, LCI clearly outperformed CC.

Table 1. Image retrieval precision results. The table should be interpreted as: For CC method, 112 queries (out of 328) found their unique answers in the first rank, 181 queries found their unique answers within the first 10 returns etc. The average rank of all 328 answers is 118. Similar interpretation applies to the LCI method.

	Ranks				
Methods	1	<=10	<=30	<=50	Mean Rank
LCI	139	226	258	272	69
CC	112	181	216	233	118

4.2 Recall

In this experiment, we collected 99 query images, each has a set of hand labeled "correct" answers. The number of correct answers for different queries ranging from 3 to 30. Let $\boldsymbol{Q}_i$ be the ith query image and let $\boldsymbol{Q_i}(1)$, $\boldsymbol{Q_i}(2)$ … $\boldsymbol{Q_i}(N_i)$ be the N_i "correct" answers to the query $\boldsymbol{Q}_i$. We define the following accumulated recall measures

$$A\mathrm{Re}call(l) = \sum_i \left(\left| \{Q_i(j) \mid rank(Q_i(j)) < l\} \right| / N_i \right) \quad (8)$$

ARecall(l) is a weighted score of how many correct answers are returned in the first l positions, accumulated over all queries. If there are more correct answers, the weighting is lower (propotional to the inverse of the number of correct ansers). It is therefore a measure of recall performance. A higher value of ARecall indicates a better performance. A similar measure was used in [4], but instead of measing each indicidul query, we measure the accumulated performance for many query, a fairer measure. Fig.4 shows the recall performance of LCI and CC. It is seen that LCI performed better. Some examples of retrieved images are shown in Fig. 5.

5 Concluding Remarks

In this paper, we have introduced a new method for color indexing. It can be considered as an extension to the classic color indexing. The method significantly enhances the power of color indexing but at the same time retains its simplicity and elegance. The concept introduced in section 2 can be extended to other indexing features as well. Only a simple implementation algorithm is presented in this paper which has been shown to perform excellently and better than a state of the art technique with similar complexity.

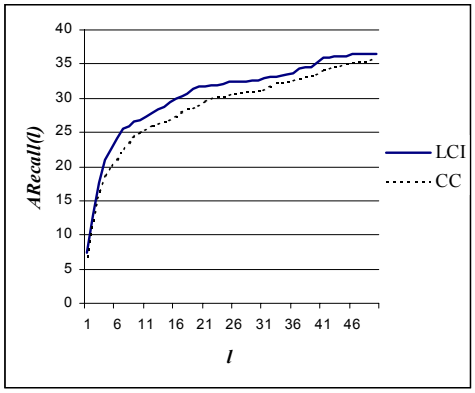

Fig. 4. Recall performance of the LCI and CC methods.

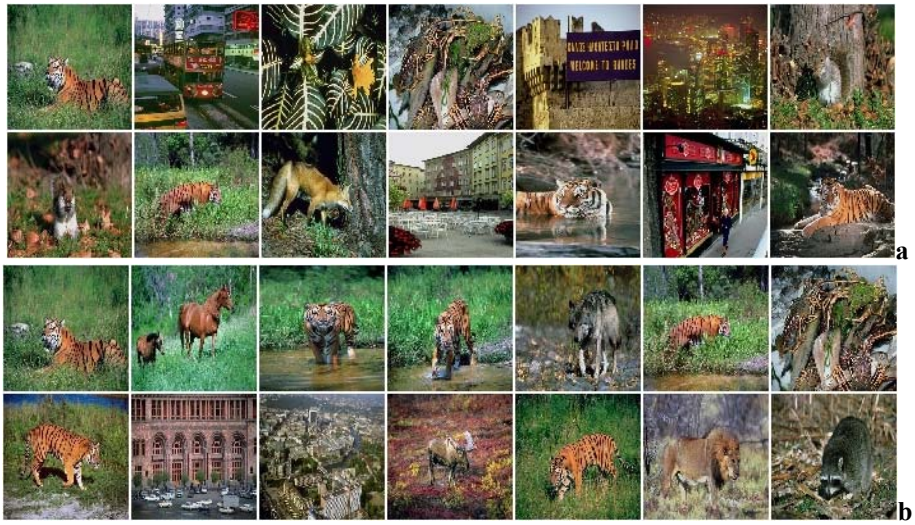

Fig. 5. Examples of retrieved images. (a) results of CC method. (b) results of LCI method. The first image is the query.

References

1. 1. A. W. M. Smeulders et al, "Content-based image retrieval at the end of the early years", IEEE Trans PAMI, vol. 22, pp. 1349 - 1380, 2000
2. P. K. Kaiser and R. M. Boynton, Human Color Vision, Optical Society of America, Washington DC, 1996
3. M. Swain and D. Ballard, "Color Indexing", International Journal of Computer Vision, Vol. 7, pp. 11-32, 1991
4. J. Huang et al, “Spatial color indexing and applications”, International Journal of Computer Vision, pp. 245 - 268, 1999
5. P. J. Burt and E. H. Adelson, “The Laplacian pyramid as a compact image code”, IEEE Trans. Commun., vol. 31, pp. 532 – 540, 1983
6. Blakemore and F. W. Campbell, “On the existence of neurones in the human visual system selectively sensitive to the orientation and size of retinal images”, Journal of Psychology, vol. 204, pp. 237 – 260, 1969
7. Y. Rui et al, “Relevance feedback: A power tool for interactive content-based image retrieval”, IEEE Trans. CSVT, 1998, pp. 644 - 655
8. A. N. Akansu and R. A. Haddad, **Multiresolution signal decomposition**, Academic Press, 1992
9. J. Wang and E. H. Aldelson, “Representing moving images with layers” IEEE Trans on Image Processing, 1994, pp. 625 - 637