

COLOR IMAGE EDGE DETECTION USING CLUSTER ANALYSIS

Hai Tao , Thomas S. Huang

Beckman Institute, University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA,
E-mail:tao@ifp.uiuc.edu

ABSTRACT

A color image edge detection algorithm is proposed in this paper based on the idea that use global color information to guide local gradient computation. Major chromatic components of an image are first extracted through cluster analysis. According to these color clusters, a set of linear chromatic transforms are generated. An appropriate chromatic transform is chosen for each pixel to maximize the gradient magnitude. In this way, edges are treated as transitions from one cluster to another. The algorithm is implemented and experimental results for real color images are included.

1. INTRODUCTION

Color perception is an important part of the image cognition process. Under certain circumstances, human vision system chooses color rather than shapes and textures as the major discrimination attribute [1][4]. For this reason, color image edge detection algorithms have received more attention recently and several algorithms have been proposed [2] [3] [5] [6] [7]. In some of these algorithms, the given color space is first transformed into other spaces in which the coordinates match the perceptual attributes of colors, e.g. HSI space [2] [3] or YT₁T₂ [5] space. In these spaces, components are relatively independent to each other. Traditional edge detectors can be adopted in each channel and the results are recombined in ways that make use of both color and luminance information [2] [5]. Fuzzy logic approach is applied to make the algorithms robust to noise[3]. Other methods also use local color vector distribution statistics to correctly detect edges against noise[7].

The key idea in this proposed algorithm is to detect edges based on statistic characteristics of each individual color image. A un-supervised clustering algorithm is applied to find major clusters in the color space. For a color edge pixel, its neighboring pixels tend to belong to different clusters. The best linear projection which maximizes the

inter-cluster distance will be used to re-combine the edge magnitudes in individual channels. Section 2 of this paper will explain this approach in more details. Simulation results on real images are presented in section 3. Conclusions and future works can be found in section 4.

2. THE APPROACH

For a given color image, its monochrome version is derived by extracting the luminance component for each pixel. A linear combination of the R, G, and B components is usually used. Then, edges can be extracted from this monochrome image by using gradient based algorithms. An alternative approach is to detect edges in each individual channel and recombine the edge magnitudes linearly for each pixel. The above two methods are equivalent as long as the edge detector is a linear operator. This is straight-forward by considering the following formula:

$$I' = 0.30I'_r + 0.59I'_g + 0.11I'_b \quad (1)$$

$$= (0.30I_r + 0.59I_g + 0.11I_b)' \quad (2)$$

where I'_r, I'_g, I'_b and I_r, I_g, I_b denote the gradients and the intensities in each color channel respectively. This process can also be viewed as the projection of multi-dimensional gradient vectors to scalar values. By using the same linear recombination, some gradient vectors with significant magnitudes can be project to small values if the projection direction is not at the same direction of the gradient vectors. For example, in Figure 1, the edge magnitude between an orange($(r, g, b) = (1.0, 0.4, 0.0)$) color region and a cyan($(r, g, b) = (0.0, 0.72, 1.0)$) color region virtually equals 0 if equation 1 is applied. If several possible projections are derived for a given color image by statistic analysis, then for each pixel, the gradient vector is projected in a way which results the maximum edge value. The color information can be used efficiently in this way. In Figure 1, when linear combination $I' = (I'_r, I'_g, I'_b) \cdot u^t, u = (0.69, -0.22, -0.69)$ is used, the edge can be detected efficiently.

To derive these projection vectors, clustering algorithm is first applied in RGB space. Each pixel of the color image is treated as a vector in the RGB space. Each cluster may contains pixels in regions of similar color. The edges are usually detected around the boundaries of different regions. Suppose two regions R_1 and R_2 belong to clusters C_1 and

This work was supported in part by Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003, and in part by Joint Services Electronics Program Grant ONR N00014-96-1-0129.

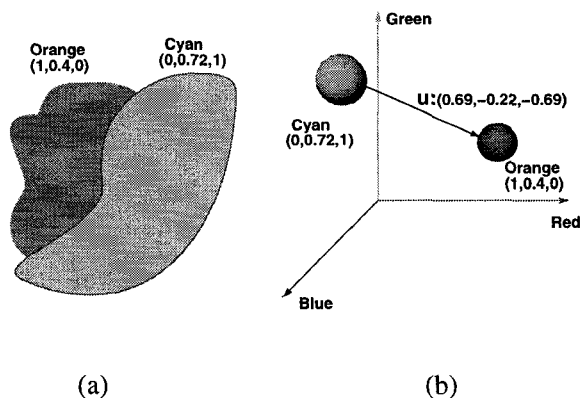


Figure 1: (a) edge between two color regions (b) u - the best linear combination.

C_2 . The centers of these two clusters are v_1 and v_2 . The gradient vector of a pixel around the boundary between R_1 and R_2 is likely to have the same direction of $v_1 - v_2$. Then $v_1 - v_2$ is a good choice of projection vector for edges between R_1 and R_2 . For a given color image, if n clusters are found, then there are $n(n-1)/2$ possible projection vectors. For each pixel of the image, its gradient vector is projected to all these $n(n-1)/2$ projection vectors and the maximum value is adopted. Obviously, this method emphasizes the edges between detected color clusters.

When an image contains many color clusters, the number of projection vectors will be too large to handle. Two solutions for this problem are setting maximum number of clusters for each image and merging similar projection vectors.

3. SIMULATION RESULTS

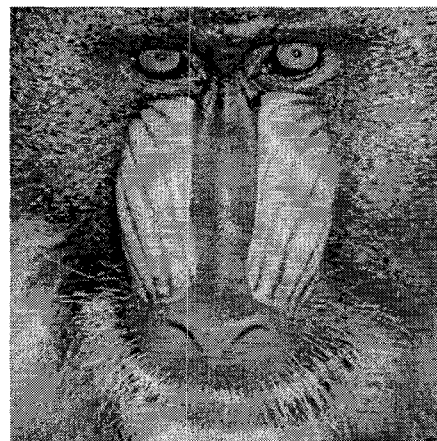
The proposed algorithm has been implemented and applied to several real images. The gradient information in each color channel (R, G or B) is calculated using Sobel operator. The maximum number of clusters allowed for each image is 6. Each projection vector is normalized to length 0.67, which is the length of standard projection vector $[0.30, 0.59, 0.11]$. That standard projection vector is also included in the optimal projection vectors. So, the number of possible projection vectors is $6 \times (6-1) + 1 = 31$.

Merging process is applied to reduce this number. At the beginning, each vector is assigned a weight 1.0. If the distance between two vectors is smaller than a certain threshold, they are merged and represented by a normalized weighted average vector. The weight for this new vector is the sum of the two vectors' weights which are merged. This process continues until the distance between any two vectors is larger than the threshold.

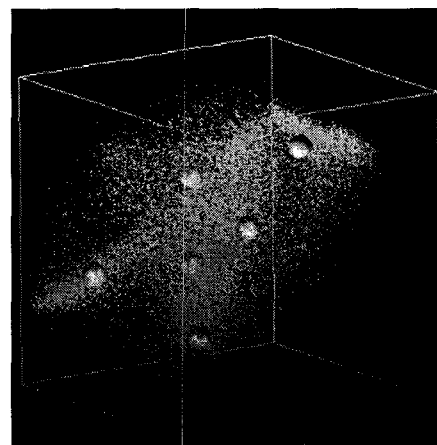
It is obvious that the resulted edge magnitude is larger than that of the standard method at any pixel because a

set of better vectors are used for recombination. A factor of 0.6 is multiplied to reduce the edge magnitude and this also suppress insignificant edge details.

The original *Mandrill* image is shown in Figure 2a. In Figure 2b, each point in the RGB 3D space represents a pixel and each sphere represents a color cluster. There are 6 clusters in this image and two of them represent the blue region and the red region in the picture. The results of the proposed edge detector is shown in Figure 3b and the result of applying Sobel edge detector on gray *Mandrill* image is shown in Figure 3a.



(a)



(b)

Figure 2: *Mandrill*: (a) original image (b) clustering in RGB 3D space.

It is observed that the proposed method significantly enhances the edges between major color regions. The proposed algorithm can also work with other color representation and monochrome image edge detectors.

4. CONCLUSIONS AND FUTURE WORKS

The advantage of the proposed algorithm is that the global information of a given color image is exploited to guide the local edge detector. Use of color information is enabled by having multiple projection vectors. The underlying assumption is the image contains several color regions. Some problems related to this scheme include the computation complexity caused by the clustering process. This problem is partially solved by sub-sampling the original color image. When relative low noises appear in color images, the



(a)



(b)

Figure 3: *Mandrill*: (a) Sobel edge detection on gray *Mandrill* image (b) edge detected using proposed method.

resulted clusters are approximately the same which means the projection vectors can be generated robustly. Problems may arise when the noise affects the clustering process seriously.

One of the future work is to increase the regularity of the detected edges. This can be achieved by restricting the neighboring gradient vectors to similar projection vectors.

5. REFERENCES

- [1] J. T. Allen and T. Huntsberger. Comparing color edge detection and segmentation methods. In *IEEE SOUTH-EASTCON'89*, pages 722–728, 1989.
- [2] T. Carron and P. Lambert. Color edge detector using jointly hue, saturation, and intensity. In *IEEE Int. Conf. on Img. Proc. '94*, pages 977–981, 1994.
- [3] T. Carron and P. Lambert. Fuzzy color edge extraction by inference rules - quantitative study and evaluation of performances. In *IEEE Int. Conf. on Img. Proc. '95*, pages 181–184, 1995.
- [4] Anil K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Englewood Cliffs, 1989.
- [5] R. Nevatia. A color edge detector and its use in scene segmentation. *IEEE Trans. Syst. Man Cybern.*, 7:820–826, 1977.
- [6] M. A. Abidi R. A. Salinas, C. Richardson and R. C. Gonzalez. Data fusion: color edge detection and surface reconstruction through regularization. *IEEE Trans. Ind. Electron.*, 43(3):355–363, 1996.
- [7] P. E. Trahanias and A. N. Venetsanopoulos. Vector order statistics operators as color edge detectors. *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 26(1):135–143, 1996.