Comparison of Color Edge Detection Techniques in Various Color Spaces

Ravneet Kaur Sidhu

Student, Masters of Technology, Department of Computer Science
Punjabi University, Patiala, Punjab, India

ABSTRACT

Since edge forms the outline of an object, edge detection is an important research task in the field of image processing. Many image processing applications require the edge detection as a beginning stage of an image processing. Edge detection of gray-scale and color images gives different results. Color images contain more information than gray-scale images. So, color edge detection is an important method. There are many color spaces such as RGB, HSV, YC_bC_r and $L^*a^*b^*$. In this paper, we applied different color edge detection techniques on each color space. The results of different edge detection techniques in different color spaces have been analyzed for their accuracy and computational speed. Experiments show that $L^*a^*b^*$ color space gives the best edge details with least computational cost using vector-valued edge detection technique in comparison to vector-valued and traditional edge detection tecnhiques in all other color spaces.

Keywords: color edge detection, edge thinning technique, HSV color space, $L^*a^*b^*$color space, RGB color space, YC_bC_r color space, threshold selection.

1. INTRODUCTION

Edge detection is a process of detecting the presence and location of local changes of intensity in an image. Edges are the meaningful discontinuities of an image and form an outline of an object. Edge representation reduces the amount of data to be processed, preserving the important information about the shapes of objects. Edges are used in various image-processing techniques like object identification, stereo analysis, region segmentation, watermarking etc. [1]. Most of the images today, are color images. Color is a powerful descriptor as humans can distinguish between thousands of color shades in comparison to only few gray shades. So, use of color in image processing is highly motivated. Ninety percent of the edges are same in gray-scale and in color images. Still there are ten percent of the edges that are not detected in gray-scale images which can be important for certain vision tasks [2]. This difference comes from the change of the color. Color images provide more edge information as compared to gray scale images. In gray-scale images, a scalar gray-level is assigned to each pixel of the image; while in color images, a color vector is assigned to each pixel.

No detailed study has been done yet for different edge techniques in different color spaces. Different approaches have been used for color edge detection in [3] and [4] and it was observed that the results from HSV color space are better than YC_bC_r and YIQ; and RGB and YIQ color spaces, respectively. $L^*a^*b^*$ color space has not been covered. This paper presents the various color edge detection techniques in different color spaces such as RGB, HSV, YC_bC_r and $L^*a^*b^*$ and shows the color space that best detects the edges with least computational cost.

2. COLOR SPACES

Color spaces such as RGB, HSV, $L^*a^*b^*$, YUV, YC_bC_r etc., provide the description of the color. Color images are in RGB color space. RGB color space specifies the color using three components: red, green and blue. It is ideally suited for hardware implementation. HSV color space specifies the color using hue, saturation and value. This model decouples the value (intensity) component from the color carrying information (hue and saturation) in a color image. It is the ideal tool for describing colors and is used in many image-processing algorithms. $L^*a^*b^*$ color space is mainly used for high-quality photographic applications.

The three components; luminosity, hue and saturation; are relative with respect to a reference white point $C_{ref} = \left(X_{ref}, Y_{ref}, Z_{ref}\right)$[5]. YC_bC_r color space is an international standard used for digital television and image compression. The luminance component Y is separated from the two chroma components [6].

Conversion of RGB color space into HSV color space. The conversion of R, G and B into H, S and V components is as follow [7]:

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G' \end{cases} \quad (1)$$

with $\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\right\}$

$$S = 1 - \frac{3}{(R+G+B)}[min(R,G,B)], \quad (2)$$

$$V = \frac{1}{3}(R + G + B). \quad (3)$$

Conversion of RGB color space into $L^*a^*b^*$ color space. The conversion of R, G and B into L^*, a^* and b^* components is as follows [5]:

$$\left.\begin{aligned} L^* &= 116 \cdot Y' - 16 \\ a^* &= 500 \cdot (X' - Y') \\ b^* &= 200 \cdot (Y' - Z') \end{aligned}\right\}, \tag{4}$$

With $$X' = f_1\left(\frac{X}{X_{ref}}\right), Y' = f_1\left(\frac{Y}{Y_{ref}}\right), Z' = f_1\left(\frac{Z}{Z_{ref}}\right), \tag{5}$$

$$f_1(c) = \begin{cases} c^{\frac{1}{3}} & for\ c > 0.008856 \\ 7.787 \cdot c + \frac{16}{116} & for\ c \leq 0.008856 \end{cases}, \tag{6}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{7}$$

Reference white point is: $(X_{ref}, Y_{ref}, Z_{ref}) = (0.950456, 1.000000, 1.088754)$.

Conversion of RGB color space into YC_bC_r color space. The conversion of RGB into YC_bC_r is as follow [6]:

$$\begin{pmatrix} Y \\ C_b \\ C_r \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \tag{8}$$

3. COLOR EDGE DETECTION TECHNIQUE

Edge detection basically consists of the following steps: Smoothing (to suppress noise while preserving true edges), detecting edges (through some edge detection method), thinning (to narrow down the ridges), and finally thresholding (to produce the binary edge map). This paper presents the edge detection on noise-free images, so smoothing is not required.

Edge Detection Technique. Edge detection measures the gradient magnitude at the pixel location. The gradient is the vector pointing in the direction of maximum rate of change of function $f(x, y)$. One of the technique for computing gradient is to extend the gray-scale edge detection to color images. The other technique is to extend the gradient to vector functions.

Traditional Edge Detection Technique. The first order derivatives are used for edge detection using the magnitude of the gradient. For a function $f(x)$, the gradient at coordinates (x, y) is defined as [7]:

$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}. \quad (9)$$

The magnitude of above vector at (x, y) is denoted as:

$$M(x,y) = mag(\nabla f) = \sqrt{g_x^2 + g_y^2}) \, . \quad (10)$$

-1	-2	-1
0	0	0
1	2	1
-1	-2	-1

-1	0	1
-2	0	2
-1	0	1
-1	0	1

Fig. 1. Sobel Operator masks: (a) x-direction mask and (b) y-direction mask.

Approximations to g_x and g_y can be derived using the masks in Fig. 1(a) and (b). Magnitude is computed as in Eq. 10.

Vector-valued Edge Detection Technique: The gradient is applicable to scalar functions and this concept cannot be applied to vector functions. So another technique used for extending this concept to vector functions is Di Zenzo's vector method [7,8].

Let r, g and b be the unit vectors along the $R, G \ and \ B$ axes in RGB color space; then the vectors for the full image$f(x, y)$ can be defined as:

$$u = \frac{\partial R}{\partial x}\vec{r} + \frac{\partial G}{\partial x}\vec{g} + \frac{\partial B}{\partial x}\vec{b} \text{ and, } v = \frac{\partial R}{\partial y}\vec{r} + \frac{\partial G}{\partial y}\vec{g} + \frac{\partial B}{\partial y}\vec{b}. \quad (11)$$

$g_{xx}, g_{yy} \ and \ g_{xy}$ are defined in terms of the dot product of these vectors, as follows:

$$g_{xx} = u \cdot u = u^T u = \left(\frac{\partial R}{\partial x}\right)^2 + \left(\frac{\partial G}{\partial x}\right)^2 + \left(\frac{\partial B}{\partial x}\right)^2, \quad (12)$$

$$g_{yy} = v \cdot v = v^T v = \left(\frac{\partial R}{\partial y}\right)^2 + \left(\frac{\partial G}{\partial y}\right)^2 + \left(\frac{\partial B}{\partial y}\right)^2, \quad (13)$$

$$g_{xy} = u \cdot v = u^T v = \left(\frac{\partial R}{\partial x}\right)\left(\frac{\partial R}{\partial y}\right) + \left(\frac{\partial G}{\partial x}\right)\left(\frac{\partial G}{\partial y}\right) + \left(\frac{\partial B}{\partial x}\right)\left(\frac{\partial B}{\partial y}\right). \quad (14)$$

The direction of maximum rate of change is given by:

$$\theta(x, y) = \left(\frac{1}{2}\right)\tan^{-1}\left(\frac{2g_{xy}}{g_{xx} - g_{yy}}\right). \quad (15)$$

The value of maximum rate of change is

$$G(x,y) = \left\{\frac{1}{2}\left[\left(g_{xx} + g_{yy}\right) + \left(g_{xx} - g_{yy}\right)\cos 2\theta + 2g_{xy}\sin 2\theta\right]\right\}^{\frac{1}{2}}. \qquad (16)$$

For HSV, YC_bC_r and $L^*a^*b^*$ color spaces, $R, G\ and\ B$ components in the above algorithm are replaced by $H, S\ and\ V$; Y, C_b and C_r ; and L^*, $a^*\ and\ b^*$, respectively. In case of HSV color space, the cylindrical coordinate system is transformed into Cartesian coordinate system, using $X = S\cos(H), Y = S\sin(H)\ and\ Z = V$ and then used for gradient computation [9]. This transformation enables the vector-gradient to be applied on XYZ space which is orthogonal.

Thinning Technique. Edge map produced by the color edge detection technique contains thick edges. So, next step is to thin those edges. This can be done by considering the orientation of the gradient vector. We define the edge direction from the angle $\theta(x,y)$computed in Eq. 15. We use the different angle ranges for edge direction determination [7]. After finding the direction of the edge gradient, the two neighbors of the central pixel, in the corresponding direction, are compared with the central pixel. The value of the central pixel is suppressed, if its value is less than at least one of its two neighbors, else the value is retained.

Threshold Technique. Threshold technique is very important in edge detection as the accuracy of the results depends upon the choice of threshold parameters. It basically produces a binary image from the gray-scale image. It discards the weak pixels. An Optimum Global Thresholding Using Ostu's Method [7] is used to obtain the threshold value that separates the main edges from the insignificant edge pixels. The threshold value obtained from this method is used to classify the pixels of the gray-scale image obtained from thinning technique, into two classes, and thereby forming a binary image.

Table 1. Various approaches used for edge detection.

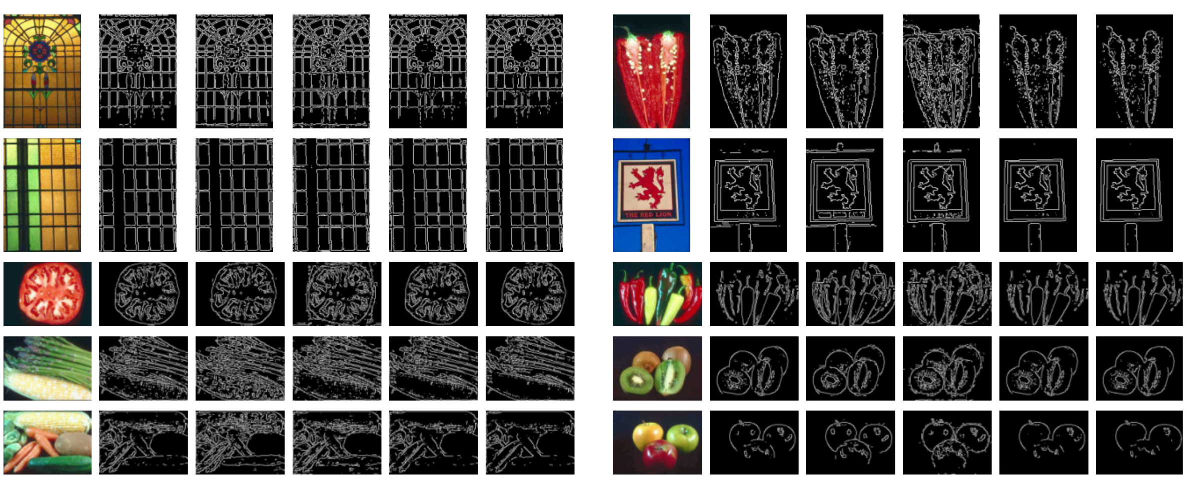
Approach	Technique
A	Vector-valued edge detection technique in RGB color space.
B	Vector-valued edge detection technique in $L^*a^*b^*$ color space.
C	Vector-valued edge detection technique in HSV color space.
D	Traditional edge detection technique on V component in HSV color space.
E	Traditional edge detection technique on Y component in YC_bC_r color space.

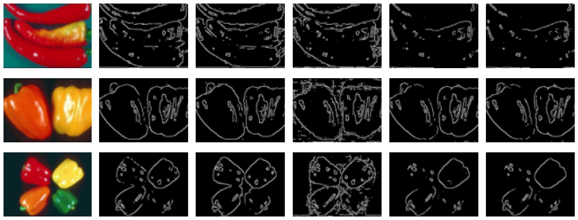
4. EXPERIMENTS AND RESULTS

In this section, we demonstrate the performance of various approaches used for edge detection in different color spaces, in terms of accuracy and computational speed. The implementation of the various edge detection approaches has been done in Microsoft Visual C++ 6.0 under Windows 7.0 on a PC with 2.26 GHz CPU and 3 GB RAM. The four color spaces used for implementation and analysis are: RGB, HSV, YC_bC_r and $L^*a^*b^*$. The various approaches used for edge detection are shown in Table 1. 15 RGB images, from Corel 5000 dataset [14], are used as test images. The test images and edge detail results can be viewed in Fig. 2, 3. The computational speed for 15 test images for various approaches can be viewed in Table 2.

Experiments show that approach B gives better edge details than approach C on 13 images with a little computation advantage of 0.049 millisec. Even the results of approach A are better than approach C on 9 images with an advantage of 0.189 millisec. In case of approach A and B, on 12 images approach B shows better results than approach A with an extra overhead of 0.14 millisec. Thus, approach B gives the best edge details followed by approach A and C. Vector-valued edge detection technique gives better edge details than the traditional edge detection technique. Approach D and E give almost same edge details for all test images with approach D having computation advantage of 0.048 millisec over approach E.

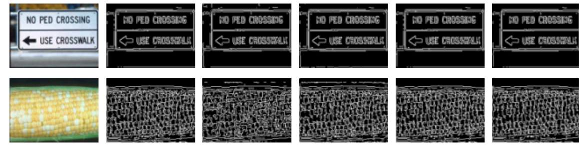
Although, the computational time requirement for traditional edge detection approaches is significantly lower than the vector-valued edge detection approaches, but they provide very less edge detail. With vector-valued edge detection technique, approach B provides the best edge details with computational advantage of 0.049 millisec over the approach C and computational overhead of 0.14 millisec. over approach A.





(a) (b) (c) (d) (e) (f)

Fig. 2. (a) Original Images, (b) Approach A, (c) Approach B, (d) Approach C,
(e) Approach D, (f) Approach E.



(a) (b) (c) (d) (e) (f)

Fig. 3. (a) Original Images, (b) Approach A, (c) Approach B, (d) Approach C,
(e) Approach D, (f) Approach E.

Table 2. Computational time taken for 15 test images.

Approach	**Computational time (in millisec.)**
A	1.201
B	1.341
C	1.39
D	0.498
E	0.546

5. CONCLUSION

The different approaches: vector-valued and traditional edge detection techniques, are analyzed in the four color spaces: RGB, HSV, YC_bC_r and $L^*a^*b^*$. The results are based on subjective evaluation of edge detail and computational speed. It is observed that vector-valued edge detection technique gives the best edge details in $L^*a^*b^*$ color space with least computational cost in comparison to the edge details provided by vector-valued and traditional edge detection techniques in other color spaces.

REFERENCES

[1] Koschan A., Abidi M., *Detection and classification of edges in color images.* IEEE Signal Processing Magazine 2005; 22(1):64-73.

[2] Koschan A., *A comparative study on color edge detection*. Proc. 2nd Asian conf. on Computer Vision ACCV'95, Singapore, 1995; 3:574-578.

[3] Chauhan P., Shahabade R. V., *Edge detection comparison on various color spaces using histogram equalization.* International Journal of Advanced Computational Engineering and Networking, June 2013; 1(4).

[4] Jeon G., *Measuring and comparison of edge detectors in color spaces.* International Journal of Control and Automation 2013; 6(5): 21-30.

[5] Burger W., Burge M. J., *Principles of Digital Image Processing: Core Algorithms.* Springer, 2009.

[6] Burger W., Burge M. J., *Principles of Digital Image Processing: Fundamental Techniques.* Springer, 2009.

[7] Gonzalez R. C., Woods R. E., *Digital Image Processing*. Pearson, Third Edition, 2008.

[8] Zenzo S. D., *A note on the gradient of a multi-image, Computer Vision, Graphics, and Image Processing.*1986; 33(1):116-125.

[9] Liu G., Yang J., *Content-based image retrieval using color difference histogram*. Pattern Recognition 2013; 46:188-198.

[10] Information on http://www.ci.gxnu.edu.cn/cbir/Dataset.aspx.