IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing

Efficient Color Edge Detection Using Saturated Color Space

Mohammad Zolfaghari
Department of Communications and Electronics
Faculty of Electrical and Computer Eng., Shiraz University
Shiraz, Iran
zolfaghari63@gmail.com

Mehran Yazdi
Department of Communications and Electronics
Faculty of Electrical and Computer Eng., Shiraz University
Shiraz, Iran
yazdi@shirazu.ac.ir

Abstract—Saturating color spaces aims to approach the image pixels belonging to the same regions chromatically which hereby causes a better segmentation. However a simple saturation may produce spurious edges and disappear some real edges. In this paper, we propose an improved algorithm which preserves much efficiently the real edges and prevent creating spurious ones. Comparing the results obtained by our algorithm with those by the recently proposed method of Chung et al. [7] demonstrates the superiority of our algorithm.

Keywords-CIE lu'v' color space ; image segmentation ; edge detection ; edges preserving.

I. INTRODUCTION

Contrast enhancement or saturating an image can be described as an important factor in image segmentation which can be based on seed determination [1][2], region growing [3] and region merging [4].

Although increasing image contrast results in a better segmentation, losing edges and creating spurious edges may occur in the edge detector process.

A various methods have been proposed for contrast enhancement such as using filters [5], histograms and chromatic features [6]. These methods can successfully deal with contrast enhancement but can poorly preserve edges.

Recently Chung et al. [7] have proposed a method of edge preserving based on color space saturation. Although their method opens a new direction in color space based saturation, it contains some deficiencies that can be improved. In this paper, we propose an improvement in edge preserving algorithm based on saturating CIE Lu'v' color space. The comparisons are made by applying Trahanias edge detector [7] on saturated images obtained by Chung's algorithm and our algorithm which demonstrate the superiority our algorithm.

II. METHOD OF CHUNG ET AL.

A. Image saturating

As indicated in Fig. 1(a), for a pixel like C which is within the R'G'B' triangle in CIE Lu'v' color space [7] and has the adequate distance from W, the maximum amount of saturation can be obtained by moving the pixel $C = (u'_C, v'_C, L)$ along the line segment $\overline{WC}$ and finding the intersection point of the line segments of $\overline{WC}$ and $\overline{R'B'}$ .This intersection point is called $C_S = (u'_{C_S}, v'_{C_S}, L_{C_S})$ and indicates the maximum amount of saturation for the pixel C. Note that this procedure does not affect the brightness of the pixel, so $L_{CS} = L$.

B. Image desaturating

Maximizing the amount of saturation for the image pixels causes two problems; (1) losing edges and creating a number of spurious edges and (2) increasing the intensity of colors and causing that the image may become visually abnormal. Therefore the desaturation process is needed. As can be seen in Fig. 1(b), the others of papers [6] [7] are determined the component of desaturated pixel $C_{ds} = (u'_{c_{ds}}, v'_{c_{ds}}, L_{c_{ds}})$ by using the central gravity law of color mixture as presented in (1):

$$u'_{C_{ds}} = \frac{u'_W \frac{L_W}{v'_W} + u'_{C_s} \frac{L}{v'_{C_S}}}{\frac{L_W}{v'_W} + \frac{L}{v'_{C_S}}} \quad , \quad v'_{C_{ds}} = \frac{L_W + L}{\frac{L_W}{v'_W} + \frac{L}{v'_{C_S}}} \qquad L_{C_{ds}} = L_W + L \tag{1}$$

where $L_W = K\overline{L}$, $\overline{L}$ is the mean of color image luminance and k is a factor to control the image luminance and saturation.

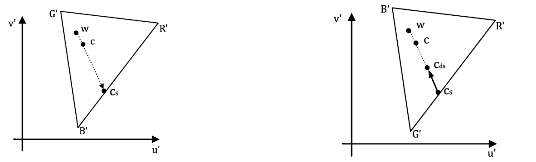
C. Algorithm of Chung et al.

Chung et al. use following steps to perform the procedure of edges preserving.

Step1. Firstly the edges of image in the CIE Lu'v' color space are detected by Trahanias edge detector. There are two cases for each pixel like C; the pixel C is an edge pixel which is denoted as E(C) =1, or the pixel C is not an edge pixel which is denoted as E(C) = 0.

Step2. Each pixel in the saturated image C_s, is transferred to the pixel C_{ds} by (1).

Step3. An 3×3 window (see Fig. 2) is selected for scanning the whole image rows horizontally by



a) Saturation step (b) Desaturation step

Figure 1. Saturation and desaturation steps.

P	P	P
P	C	u
u	u	u

Figure 2. 3× 3 window for scanning image rows.

applying the previous step. In this window the symbol P means that the edge preservation procedure is done upon these pixels, the symbol C shows that the edge preservation procedure is applied on the central pixel of window and the symbol u denotes that the procedure will be done upon these pixels.

In each step of window movement, the case of central pixel should be determined by the edge detector. Four cases can occur: (1) E(C)=1 and after transferring the pixel C to the pixel C_{ds}, $E(C_{ds})$ =1, (2) E(C)=0 and after transferring $E(C_{ds})$=0. In cases 3 and 4, the value of E(C) is Not equal to $E(C_{ds})$. The cases 3 and 4 are noticed as an error.

The color point C_{es} is defined as a color point between the points C and C_s, saturating the color point C without producing an error.

Step4. According to the previous step, if the cases 1 and 2 occur then C_{es}= C_{ds}, otherwise for finding the color point C_{es}, the binary □□□□□□□□□□□□□□□□□□□alternative search algorithm in the next step should be used.

Step5. If E(C) =1 and the pixel C is transferred to the pixel C_{ds} where $E(C_{ds})$=0, then the error occurs. For removing the error, firstly the pixel C_{ds} is transferred to the pixel $C_{t(1)}$ where $E(C_{t(1)})$=1. According to Fig. 3, $\left|\overline{C_{ds}C_{t(1)}}\right| = d$ and the distance d is empirically equal to $\frac{4\left|\overline{C_s C_{ds}}\right|}{5}$. In the next phase, the pixel $C_{t(1)}$ is transferred to $C_{t(2)}$ where $\left|\overline{C_{t(1)}C_{t(2)}}\right| = d/2$. If $E(C_{t(2)})$=1 then the pixel $C_{t(2)}$ is transferred to the pixel $C_{t(3)}$ and C_{es}=$C_{t(3)}$, otherwise it is transferred to the pixel $C_{t'(2)}$ and C_{es}=$C_{t'(2)}$.

The same procedure is applied on the pixel in the right side of the color point C_{ds} and on the closest pixel to the pixel C_{ds} to find C_{es}.

If the above procedure could not find the edge – pixel for the central pixel of the window, then C_{es}=C_{ds} is chosen.

This procedure can be done for the non edge – pixels and they become the edge-pixels.

III. Proposed Algorithm

This section devotes to present our proposed algorithm, but before that, the deficiencies in the previous method of edge preserving are discussed briefly.

(1) According to the Section II.C and Fig. 3, the distances L_1 and L_2 were not consider in the binary alternative search. If the color points $C_{t(3)}$, $C_{t(2)}$ and $C_{t'(2)}$ cannot remove the error, the distances L_1 and L_2 were not consider by the algorithm and then we must choose C_{es}= C_{ds}. However, it is possible that the point which is searched for removing the error, was between the color points C and C_{ds} or between the color points C_{ds} and C_s. (2) Equation (1) which is used for desaturating does not depend on the color point $C = (u'_C, v'_C, L)$ .It causes that some pixels are transferred from a color point like C_s to a color point like C_s between W and C. As shown in Fig. 4, in this way, it is not possible to search on the left side.

In our proposed algorithm, for desaturating the image, according to Fig. 5, the origin of coordinate system is transferred to the point W, then the color point C=(u'_c,v'_c,L) is transferred to the color point $C_{ds} = (u'_{C_{ds}}, v'_{C_{ds}}, L_{C_{ds}})$. In the desaturating process $L_{C_{ds}} = L$, and the components $u'_{C_{ds}}$ and $v'_{C_{ds}}$ are computed by (2) through (4).

$$\sqrt{(u'_{C_{ds}} - u'_{C_s})^2 + (v'_{C_{ds}} - v'_{C_s})^2} = d_1 \tag{2}$$

$$v'_{C_{ds}} = m\, u'_{C_{ds}} \tag{3}$$

$$d_1 = n\,\mathrm{d} \qquad 0 < \mathrm{n} < 1 \tag{4}$$

where d_1 is the rate of desaturation of pixel C, d is the distance between C and C_s, n is the factor which controls the rate of desaturating and m is the slope of the line segment $\overline{WC}$ and $L_{C_{ds}} = L$.

The advantage of the above method for desaturating the image is that we are absolutely sure that all the image pixels move toward the color point C_s by rate of d′. By using this method, the desaturated pixels like C_{ds} dose not place between W and C then the deficiency (2) is solved.

In the next step, our proposed algorithm is described in the following steps:

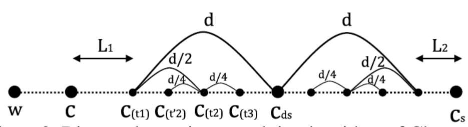


Figure3. Binary alternative search in algorithm of Chung et al. for edges preservation.

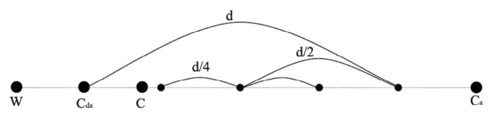


Figure4. Example of a case where a saturated pixel is moved between W and C instead of being placed between C and C_s.

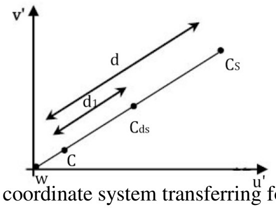


Figure 5. Origin coordinate system transferring for desaturating an image.

Step1. After transferring the original image to the CIE Lu'v' color space, the edge detection is done on the image.

Step2. The image is saturated as in Section II.A and is desaturated by (2) through (4).

Step3. This step is done like the 3rd step in Section II.C.

Step4. If the central pixel of the window is defined as a pixel like C where E(C)=1, and is transferred to C_{ds} where $E(C_{ds})=1$ or in other case if E(C)=0 and the pixel C is transferred to C_{ds} where $E(C_{ds})=0$, for these cases $C_s= C_{ds}$ and else if we have an error and we should go to the next step for finding the color point C_{es}.

Step5. As shown in Fig. 6, the color point C_{ds} should be moved from both sides to find the color point C_{es}. If $\left|\overline{CC_{ds}}\right| = D_1$ and $\left|\overline{C_sC_{ds}}\right| = D_2$, the distances D_1 and D_2 are divided into ten equal intervals experimentally, then the point C_{ds} is moved toward right side until finding a point like $C_{es(r)}$ between C_s and C_{ds}, which removes the error or reaching the point C_s. The same procedure is done in the left side to find a point like $C_{es\,(L)}$ between C_{ds} and C or to reach the point C.

If we reach the color points C and C_s then $C_{es} = C_{ds}$ else if we reach the color points C and $C_{s(r)}$ then $C_{es} = C_{es(r)}$, else if we reach the points C_s and $C_{es(L)}$ then $C_{es} = C_{es(L)}$, other wise the point C_{es} is chosen between $C_{es(L)}$ and $C_{es(r)}$, which is nearest point to the color point C_{ds} .

Step6. All the previous steps are repeated, but in the step 3 a window like in Fig. 7 is used and this window is moved on the image vertically. It causes a remarkable improvement in the edge preserving of our algorithm.

IV. EXPERIMENTAL RESULTS

In this section the results of our proposed algorithm

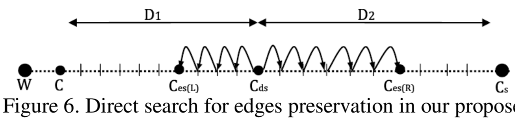


Figure 6. Direct search for edges preservation in our proposed algorithm.

p	p	u
p	c	u
p	u	u

Figure7. 3× 3 window for scanning image columns.

are compared with those of the previous algorithm.

The values of k in (1) and n in (4) are selected empirically so that the means of saturated image pixels obtained by two desaturated algorithms become equal.

We do it to fairly compare two algorithms. The mean of saturated image pixels is defined as m_s. But the value of m_s changes after using edge preservation algorithms and it is notified as m_{eff}. m_s and m_{eff} are computed by (5) :

$$m_s = \frac{1}{p}\sum_{j=1}^{p}\left|\overline{C_J C_{ds_J}}\right| \;;\; m_{eff} = \frac{1}{p}\sum_{j=1}^{p}\left|\overline{C_J C_{es_J}}\right| \qquad (5)$$

where p is the total number of pixels, $\left|\overline{C_J C_{ds_J}}\right|$ is the distance between C and C_{ds} for the pixel J and $\left|\overline{C_J C_{es_J}}\right|$ is the distance between C and C_{es} for the pixel J.

For assessing the results, we compute the differences (E_s) between the number of saturated image edges and the edges of the original image in the CIE Lu'v' color space, the differences (E_{ds}) between the number of desaturated image edges and edges of the original image in the CIE Lu'v' color space and the differences (E_{eff}) between the number of image edges preserved in the algorithm and the edges of the original image. E_s, E_{ds} and E_{eff} represent binary error images where Fig. 8(a) shows the test image Lena . Next the image is saturated and the Trahanias edge detector is applied on it. Figure 8(b) shows E_s where the lost edges and spurious edges or the errors occur by the saturating Process. The saturated image is desaturated by (1) with k =0.3 and m_s =0.0576. The computed E_{ds} is illustrated as a error image in Fig. 8(c). Using the previous edges preservation algorithm, the value of m_{eff} becomes 0.0541 and the error image E_{eff} is obtained and shown in Fig. 8(d).

By selecting n=0.4 and using (2) through (4), the saturated image is desaturated and m_s = 0.0576 is obtained. E_{ds} is shown in Fig. 8(e). We then apply the proposed edges preservation algorithm on Lena image with m_{eff} = 0.0568. The error image E_{eff} for the proposed algorithm is illustrated in Fig. 8(f).

The comparison of Fig. 8(d) and Fig. 8(f) shows that proposed algorithm preserved the edges much more than the previous algorithm, although in the both methods the amount of saturation for each pixels is the same before the edges preservation process.

We also compared the algorithms using another test image of Peppers (see Fig. 9(a)). After saturating, the images are desaturated by previous method and our proposed method. The error image of E_{eff} for the test image is shown in Fig. 9(b) by using the previous edges preservation algorithm and in Fig. 9(c) by using our proposed algorithm.

To evaluate the results quantitatively, we use the rate of improvement in edges preservation, R, which can be computed by (6).

$$R = \frac{E_s - E_{eff}}{E_s} \times 100 \qquad (6)$$

In Tables I and II, the obtained results of mention images are assessed statistically. As can be seen, the proposed algorithm achieves a significantly better rate of improvement in edges preserving for two test images. Some conclusions can be extracted from the presented results: (1) we can obtain the same rate by choosing appropriate k and n, the image is desaturated by two methods to same rate. But after using the edges preserving algorithms, m_{eff} in our algorithm is more than m_{eff} in previous algorithm. (2) The value of E_{ds} shows that the origin coordinate system transferring is a better way than the Central Gravity Law of Color Mixture for desaturating an image. (3) Although our algorithm saturates more the images, the value of E_{eff} shows that the edges preservation in our algorithm is still better than that in the previous algorithm.

V. Conclusion

In this paper, we propose an improvement in the edge preserving of saturating process of color images which causes a better segmentation in practical, because the pixels that are in the same region become more similar chromatically. To do that, firstly the image was saturated in the CIE Lu'v' color space and was desaturated by transferring the origin of coordinate system so that less real edges are lost and also less spurious edges are added in the results. So we can saturate color image more with better edge preserving. The experimental results on well known test images demonstrated the superiority of our proposed algorithm compared to previous one.

References

[1] F.Y. Shih, S. Cheng, “Automatic seeded region growing for color image segmentation”, Image Vis. Comput., pp. 877–886, 2005.

[2] Mehnert, P. Jackway, “An improved seeded region growing algorithm”, Pattern Recogn. Lett., pp. 1065–1071, 1997.

[3] J. Fan, D.K.Y. Yau, A.K. Elmagarmid, W.G. Aref, “Automatic image segmentation by integrating color–edge extraction and seeded region growing”, IEEE Trans. Image Processing, pp. 1454–1466, 2001.

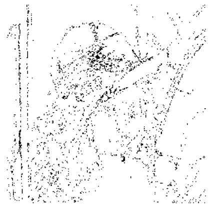
[4] R.C. Gonzalez, R.E. Woods, Digital Image Process, second ed., Prentice Hall, New York, 2002.

[5] S.J. Sangwine and R.E.N. Horne, The color image processing handbook, Chapman and Hall, London, UK, 1998.

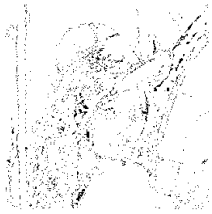
[6] L. Lucchese, S.K. Mitra, “Filtering color images in the xyY color space”, in: Proc. Int. Conf. Image Processing, Vancouver, BC, Canada, pp. 500–503, September 2000.

[7] K. L. Chung, Y. W. Liu and W. M. Yan, “Efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation,” Journal of Visual Communication and Image Representation, pp. 299-310, 2008.

(a) Original image.



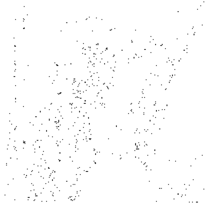
(b) Image of E_s



(c) E_{ds} by the algorithm of Chung et al. in desaturation process.

(d) E_{eff} by the algorithm of Chung et al. in edge preserving process.

(e) E_{ds} by proposed desaturation Method.



(f) E_{eff} by our proposed Preservation algorithm.

Figure 8. Obtained results for Lena image.

(a) Test image Peppers

(b) E_{eff} by the algorithm of Chung et al .

(c) E_{eff} by our proposed algorithm.

Fig. 9. Obtained results for Peppers.

TABLE I. Results of Image Lena with k = 0.3 , n =0.4

Lena image	m_s	E_s	E_{ds}	E_{eff}	m_{eff}	R(%)
Algorithm of Chung et al	0.0576	2960	2746	1865	0.0541	37
Our proposed algorithm	0.0576	2960	1193	591	0.0568	80.04

TABLE II. Results of Image Peppers with k = 0.25, n = 0.3

Peppers image	m_s	E_s	E_{ds}	E_{eff}	m_{eff}	R(%)
Algorithm of Chung et al	0.0182	2443	1799	1202	0.0181	50.8
Our proposed algorithm	0.0182	2443	1032	421	0.0221	82.7