



## Efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation

Kuo-Liang Chung<sup>a,\*</sup>, Wei-Jen Yang<sup>b</sup>, Wen-Ming Yan<sup>b</sup>

<sup>a</sup> Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, 10672 Taiwan, ROC

<sup>b</sup> Department of Computer Science and Information Engineering, National Taiwan University, No. 1, Section 4, Roosevelt Road, Taipei, 10617 Taiwan, ROC

### ARTICLE INFO

#### Article history:

Received 17 July 2006

Accepted 22 February 2008

Available online 13 March 2008

#### Keywords:

CIE color model

Color contrast enhancement

Color edge detection

Color histogram moment

Color image segmentation

Color saturation and desaturation

Edge-preservation effect

Seed-based region growing approach

### ABSTRACT

In this paper, a new and efficient edge-preserving algorithm is presented for color contrast enhancement in CIE Lu'v' color space. The proposed algorithm not only can enhance the color contrast as the previous algorithm does, but also has an edge-preservation effect. In addition, the spurious edge points occurred due to the color contrast enhancement can be well reduced using the proposed algorithm. This is the first edge-preserving algorithm for color contrast enhancement in color space. Furthermore, a novel color image segmentation algorithm is presented to justify the edge-preservation benefit of the proposed color contrast enhancement algorithm. Based on some real images, experimental results demonstrate the advantages of color contrast enhancement, edge-preservation effect, and segmentation result in our proposed algorithm.

© 2008 Elsevier Inc. All rights reserved.

### 1. Introduction

The purpose of color contrast enhancement is to enhance a color image such that the enhanced color image is more colorful than the original color image from the viewpoint of human visual system [8,21,22]. Previously, many efficient algorithms for color contrast enhancement have been successfully developed. Based on reducing color ordering approach [2], Zaharescu et al. [31] presented a color contrast enhancement algorithm. Based on the curvelet transform approach [4,26], Starck et al. [25] presented an efficient algorithm for color contrast enhancement. Recently, a two-step approach, namely the saturation step and the desaturation step, was proposed for color contrast enhancement [13,20]. In Lucchese et al.'s algorithm [13], they considered the chromaticity diagram [11,14]. In Pei et al.'s algorithm [20], the considered color domain is the modified chromaticity diagram, i.e. the CIE Lu'v' color space [11]. In [20], Pei et al. also developed some efficient methods to the restoration of Chinese paintings.

Among these previously published color contrast enhancement algorithms, although the enhanced color image has good color contrast enhancement effect, some degree of edge-loss may happen. Due to the edge-loss side effect, some further color image processing

tasks, such as color image segmentation and object recognition, may be degraded. The main motivations of this research are twofold: (1) presenting a new algorithm to come to a compromise between the edge-preservation effect and the color contrast enhancement effect and (2) presenting a novel color image segmentation algorithm to justify the edge-preservation benefit in some application.

In this paper, a new edge-preserving algorithm for color contrast enhancement is presented. Our proposed algorithm has both advantages of edge-preservation effect and color contrast enhancement. Our proposed algorithm consists of three steps: in the first step, a saturation operation is performed to maximize the color contrast effect. In order to speed up the first step, a new history-aid strategy is presented to determine the most possible side of the color gamut triangle in the CIE Lu'v' color space. In the second step, a desaturation operation is performed to enrich the colorful degree. The above two steps are similar to the previous color contrast enhancement algorithms in [13,20]. In the third step, an edge-preservation operation is performed to preserve the edge information while keeping the color contrast enhancement effect as much as possible. In addition, the spurious edge points occurred due to the color contrast enhancement can be well reduced using the proposed algorithm. Some experiments are carried out to demonstrate that our proposed algorithm has a good compromise between the edge-preservation effect and the color contrast enhancement. This is the first edge-preserving algorithm for color contrast enhancement in color space. Finally, a novel color image segmentation algorithm is presented to justify the application of edge-preservation effect. In our proposed color image segmentation algorithm,

\* Corresponding author. Fax: +886 2 27301081.

E-mail address: [k.l.chung@mail.ntust.edu.tw](mailto:k.l.chung@mail.ntust.edu.tw) (K.-L. Chung).

<sup>1</sup> Supported by the National Science Council of ROC under Contracts NSC96-2221-E-011-102-MY3, NSC96-2221-E-011-026, and NSC96-2219-E-001-001.

we have better segmentation results on our obtained enhanced image when compared to those on the previous obtained enhanced image without edge-preservation.

The remainder of this paper is organized as follows: in Section 2, first the notion of CIE Lu'v' color space is introduced and then the previous color edge detector, which will be used in our proposed color contrast enhancement algorithm, is described. In Section 3, our proposed edge-preserving algorithm for color contrast enhancement and the relevant speedup strategy are presented. In Section 4, a novel color image segmentation algorithm is presented. In Section 5, some experimental results are carried out to demonstrate advantages of color contrast enhancement, edge-preservation effect, and segmentation result in our proposed algorithm. Finally, some conclusions are addressed in Section 6.

**2. Preliminaries**

Before presenting our proposed color contrast enhancement algorithm, this section introduces two backgrounds, namely the notion of CIE Lu'v' color space [11] and the previously published color edge detector by Trahanias and Venetsanopoulos [29]. The two backgrounds will be used in Section 3.

*2.1. The CIE Lu'v' color space*

Suppose the input color image is an RGB color image. First, the transformation from the RGB color space to the CIE Lu'v' color space is described. The relevant transformation can be expressed by

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

In Eq. (1), the component Y is the L component in the CIE Lu'v' color space. Therefore, the CIE Lu'v' color space can be called the CIE Yu'v' color space. Based on the values of X, Y, and Z, the two components u' and v' can be obtained by the following equation:

$$u' = \frac{4X}{X + 15Y + 3Z}, \quad v' = \frac{9Y}{X + 15Y + 3Z} \quad (2)$$

By Eqs. (1) and (2), the RGB color space can be transformed into the CIE Lu'v' color space.

Considering only the two components u' and v', Fig. 1 depicts the so called spectral locus of the u'v' chromatic diagram and the spectral locus is depicted by the exterior curve. Within the spectral locus, the triangle area denotes the color space which can be displayed by the CRT monitor. This triangle area is commonly called the color gamut

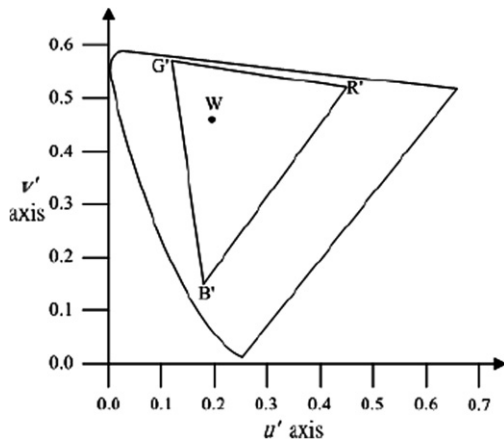


Fig. 1. The CIE u'v' chromatic diagram.

triangle. The three corners of the color gamut triangle are denoted by the three points R', G', and B' which are corresponding to (u'\_{R'}, v'\_{R'}) = (0.4507, 0.5229), (u'\_{G'}, v'\_{G'}) = (0.1250, 0.5625), and (u'\_{B'}, v'\_{B'}) = (0.1754, 0.1579). The interior point W = (u'\_W, v'\_W) = (0.1978, 0.4683) is defined as the white point [11]. Since the colors lain around the white point W are regarded as achromatic colors, it is infeasible to enhance these colors which are very near to the point W.

*2.2. Color edge detector*

In this subsection, the color edge detector by Trahanias and Venetsanopoulos [29] is described and it will be used in our proposed edge-preserving algorithm for color contrast enhancement although some other color edge detectors [23,28,32] can also be considered.

Suppose the window mask used in the color edge detector is of size w × w and the w<sup>2</sup> color pixels covered by the window mask are denoted by the set P = {P<sub>1</sub>, P<sub>2</sub>, ..., P<sub>w<sup>2</sup></sub>}. Usually, w is selected to 3 or 5. The color contrast or difference expressed in the CIE Lu'v' color space is more fruitful than that in the RGB color space [11]. Thus, instead of measuring the color difference in RGB color space, we estimate the color difference in the CIE Lu'v' color space. The three color components of each color pixel are denoted by P<sub>i</sub> = (u'\_i, v'\_i, Y<sub>i</sub>).

Based on the vector order statistic and the R-ordering concept [3], the color edge detector [29] consists of the following three steps:

- Step 1:* Sum up the color distances between each color pixel P<sub>i</sub> and the other color pixels covered by the window mask. For color pixel P<sub>i</sub>, 1 ≤ i ≤ w<sup>2</sup>, the resulting distance is given by d<sub>i</sub> = ∑\_{k=1}^{w<sup>2</sup>} ||P<sub>i</sub> - P<sub>k</sub>||, i = 1, 2, ..., w<sup>2</sup>, where || · || represents an appropriate vector norm.
- Step 2:* Sort these w<sup>2</sup> distances d<sub>1</sub>, d<sub>2</sub>, ..., and d<sub>w<sup>2</sup></sub>. Suppose these sorted w<sup>2</sup> ascending distances are d<sub>i(1)</sub>, d<sub>i(2)</sub>, ..., and d<sub>i(w<sup>2</sup>)</sub> for 1 ≤ i(1), i(2), ..., i(w<sup>2</sup>) ≤ w<sup>2</sup>. Among these w<sup>2</sup> indices, P<sub>i(1)</sub> is the color pixel with the minimal distance d<sub>i(1)</sub>; P<sub>i(w<sup>2</sup>)</sub> sometimes can be viewed as the outlier pixel in the w<sup>2</sup> color pixels.
- Step 3:* Based on the robustness consideration, compute the minimum vector dispersion (MVD) which is given by

$$MVD = \min_j \left\{ \left\| P_{i(w^2-j+1)} - \sum_{m=1}^n \frac{P_{i(m)}}{n} \right\| \right\}, \quad j = 1, 2, \dots, k; \quad k, n < w^2.$$

In [29], k and n are selected to 3 or 4 empirically. When the value of MVD is greater than the specified threshold, the central pixel of the concerned w × w subimage is determined to be an edge pixel; otherwise, it is determined to be a non-edge pixel.

**3. The proposed edge-preserving algorithm for color contrast enhancement**

This section presents our proposed novel algorithm which can come to a compromise between the edge-preservation (including spurious edge-reduction) consideration and the color contrast enhancement. In what follows, we first describe the main concepts used in our proposed algorithm, and then a speedup strategy is given to improve the proposed algorithm.

*3.1. The main concept*

As mentioned above, our proposed edge-preservation algorithm for color contrast enhancement has two considerations, namely: (1) keeping the edge information after enhancing the color image

in the CIE Lu'v' color space and (2) reducing the created spurious edges due to the side effect of the color contrast enhancement. For simplifying the exposition, in what follows, we only focus on the presentation of keeping the edge information after enhancing the color image in the CIE Lu'v' color space. In what follows, we will point out how to reduce the created spurious edges due to the side effect of the color contrast enhancement.

In order to keep the edge information in the original RGB color image as the edge-preservation reference, given an RGB color image  $I$ , the edge map is first obtained by using the previous edge detector which has been introduced in Section 3.2. For exposition, the obtained edge map is expressed by  $I_E$  where  $I_E(x, y) = 1 (= 0)$  means the color pixel  $I(x, y)$  is an edge (non-edge) pixel. Our proposed color contrast enhancement algorithm mainly consists of three steps, namely: the saturation step, the desaturation step, and the edge-preservation step.

3.1.1. The saturation and desaturation steps

Here, the saturation step and the desaturation step are described. The two steps are adopted from the previous algorithms [13,20].

Given an input color pixel  $C = (u'_C, v'_C, Y)$  as shown in Fig. 2(a), in the saturation step, the color pixel  $C$  moves forward along the line WC. When the color pixel  $C$  intersects the line segment B'R', the intersection point is called the maximally saturated color  $C_s$ . Because the saturation step affects only chromatic components  $u'$  and  $v'$ ,  $C$  and  $C_s$  are associated with the same brightness component  $Y$  where  $C_s$  is represented by  $(u'_{C_s}, v'_{C_s}, Y)$ .

When all color pixels of a color image have been maximally saturated, the image appears rather unnatural because the displayed colors are only confined at the boundary of the gamut triangle. Therefore, the desaturation step is proceeded to overcome the chromatic information reduction problem in the saturation step. Fig. 2(b) illustrates the concept of desaturation step in order to enrich the colorful degree of the color image. Instead of maximizing the color contrast by using the color point  $C_s$ , the modified color point  $C_{ds}$  is used according to the Center of Gravity Law of Color Mixture [11]. The three components of the modified color point  $C_{ds} = (u'_{C_{ds}}, v'_{C_{ds}}, Y_{C_{ds}})$  are determined by

$$u'_{C_{ds}} = \frac{u'_W \frac{Y_W}{Y} + u'_{C_s} \frac{Y}{Y}}{\frac{Y_W}{Y} + \frac{Y}{Y}}, \quad v'_{C_{ds}} = \frac{v'_W \frac{Y_W}{Y} + v'_{C_s} \frac{Y}{Y}}{\frac{Y_W}{Y} + \frac{Y}{Y}}$$

$$Y_{C_{ds}} = Y + Y_W$$

where  $Y_W = k\bar{Y}$ ;  $\bar{Y}$  is the mean luminance of the color image and  $k$  is a factor to control the luminance.

Since the determined modified color point  $C_{ds}$  may generate some degree of edge-loss or add spurious edge points, in next step, an efficient strategy is presented to solve the edge-loss problem occurred in the previous color contrast enhancement method

although the proposed strategy can be slightly modified to solve the edge-addition problem. The proposed strategy can keep a good compromise between the color contrast enhancement and the edge-preservation consideration.

3.1.2. The edge-preservation step

When  $I_E(x, y) = 1$ , i.e. the original color pixel  $I(x, y)$  is an edge pixel, it is necessary to examine whether the edge information of the original color pixel has been lost after performing the saturation and desaturation steps. When  $I_E(x, y) = 0$ , it is necessary to examine whether the mapped color pixel becomes a spurious edge pixel after performing the saturation and desaturation steps.

Fig. 3 is used to depict the concept of our proposed edge-preservation step. After performing the saturation and desaturation steps, instead of enhancing the color contrast by using  $C_{ds}$  which may lose the edge information, conceptually the edge-preserving color point  $C_{es} = (u'_{C_{es}}, v'_{C_{es}}, Y_{C_{es}})$  is determined to come to a compromise between the edge-preservation and the color contrast enhancement. Here the determined color point  $C_{es}$  is the final enhanced color point not only preserving the edge information in the RGB color space inherited from the corresponding pixel, but also enhancing the color contrast as maximal as possible. How to determine the enhanced color point  $C_{es}$  from the modified color point  $C_{ds}$  will be described in the following paragraphs.

Suppose the window mask used in the adopted color edge detector is of size  $3 \times 3$ . According to row-major scanning order, the sub-image covered by the window mask is illustrated in Fig. 4 where the symbol  $P$  denotes the enhanced color pixel which has been processed by using our proposed edge-preserving algorithm for color contrast enhancement; the symbol  $C$  denotes the current color pixel being processed and the symbol  $U$  denotes the color pixel to be processed. The previous color edge detector is applied to the subimage as shown in Fig. 4 to determine whether the current color pixel is an edge pixel or not. For convenience, let " $E(C) = 1$ " denote that the current color pixel is an edge pixel; otherwise, let " $E(C) = 0$ " denote that the current color pixel is a non-edge pixel.

For the current color pixel, if the condition  $E(C) = 1$  holds, the previous color edge detector is used to determine whether the current color pixel  $C_{ds}$  is an edge pixel or not. If the current color pixel  $C_{ds}$  is an edge pixel,  $C_{ds}$  is the final enhanced color and we perform the assignment  $C_{es} = C_{ds}$ ; otherwise, the enhanced color point  $C_{es}$  should be determined further by the help of the modified color point  $C_{ds}$ .

Suppose the modified color point  $C_{ds}$  is not an edge pixel. Fig. 5 depicts the binary alternative search direction in our proposed edge-preservation step to determine the enhanced color point  $C_{es}$ , where  $d$  denotes the first movement distance and is set to  $\frac{\|C_s C_{ds}\|}{5}$ , empirically. By running the previous color edge detector, if the current color pixel  $C_{t(1)}$  is an edge pixel, where  $\|C_{t(1)} C_{ds}\| = d$ , we move it to the next color pixel  $C_{t(2)}$  which is the near point of

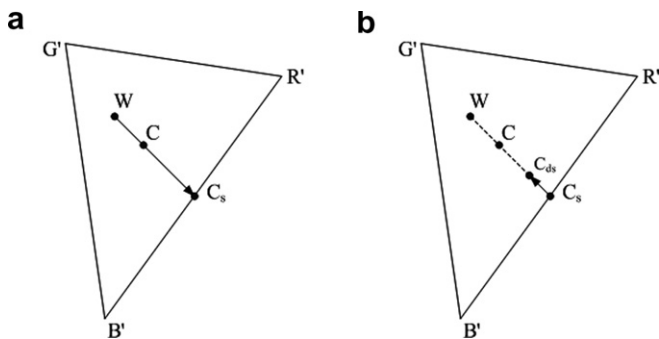


Fig. 2. The first two steps in our proposed algorithm. (a) The saturation step. (b) The desaturation step.

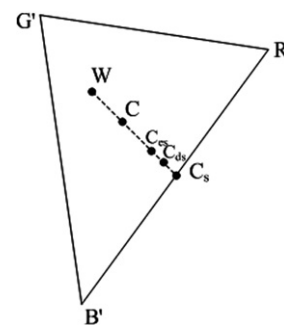


Fig. 3. The depiction of our proposed edge-preserving step.

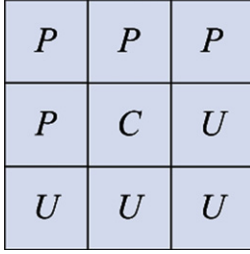


Fig. 4. One  $3 \times 3$  subimage covered by the window mask.

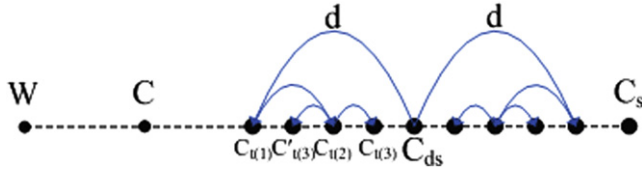


Fig. 5. Binary alternative search direction in edge-preservation step.

$C_{ds}$  and examine whether the color pixel  $C_{l(2)}$  is an edge pixel or not, where  $\|C_{l(1)}C_{l(2)}\| = \frac{d}{2}$ . If  $C_{l(2)}$  is an edge pixel, we move it to the pixel  $C_{l(3)}$ ; otherwise, we move it to the pixel  $C_{l(3)}$ . The binary alternative search in the edge-preservation step is repeated until finding an edge pixel which near the point  $C_{ds}$  as close as possible or the number of testing color pixels along the binary alternative search path is over the specified bound.

According to the above way, the color edge pixel is found at the right side of  $C_{ds}$ . Finally, we select the minimum one from the two found edge pixels at both side of  $C_{ds}$  as the enhanced color point  $C_{es}$ . Consequently, it comes to a compromise between the color contrast effect and the edge-preservation effect. For finding the enhanced color point  $C_{es}$ , our proposed binary alternative search scheme has 40% execution-time improvement ratio ( $= \frac{T_{LAS} - T_{BAS}}{T_{LAS}}$ ), where  $T_{LAS}$  denotes the time required in linear alternative search and  $T_{BAS}$  denotes the time required in binary alternative search, when compared to the linear alternative search scheme.

For the current color pixel, if the condition  $E(c) = 0$  holds, after running the previous color edge detection, on the mapping color point  $C_{ds}$ , we further test whether the mapped color point becomes a spurious edge point or not. If yes, the binary alternative search is repeated until finding the non-edge pixel which is nearest to  $C_{ds}$ ; if no, the mapped point is set to  $C_{ds}$ .

If we do not take the edge-preservation into consideration, ideally the modified color point  $C_{ds}$  is the good choice for the purpose of color contrast enhancement. In order to make the distance between the enhanced color point  $C_{es}$  and the modified color point  $C_{ds}$  as small as possible, the search direction of our approach is not always toward the white point.

After performing the previous edge-preservation step for all color pixels within the gamut triangle, given a color pixel  $(u', v', Y)$  within the gamut triangle, the mapped color pixel  $(r, g, b)$  in the RGB color domain can be obtained by first solving the following equation:

$$\begin{aligned} X &= x(X + Y + Z) \\ Y &= Y \\ Z &= z(X + Y + Z) \end{aligned} \quad (3)$$

where  $x = \frac{9u'}{6u' - 16v' + 12}$ ,  $y = \frac{4v'}{6u' - 16v' + 12}$ ,  $z = 1 - x - y$ , and  $X + Y + Z = \frac{Y}{y}$ . Next, by solving the inverse of Eq. (1), the obtained values of  $X$ ,  $Y$ , and  $Z$  in Eq. (3) can be transferred into the values of  $r$ ,  $g$ , and  $b$ .

Based on the original Peppers image (see Fig. 6(a)), Fig. 6(b) illustrates the enhanced color image obtained by using the previ-

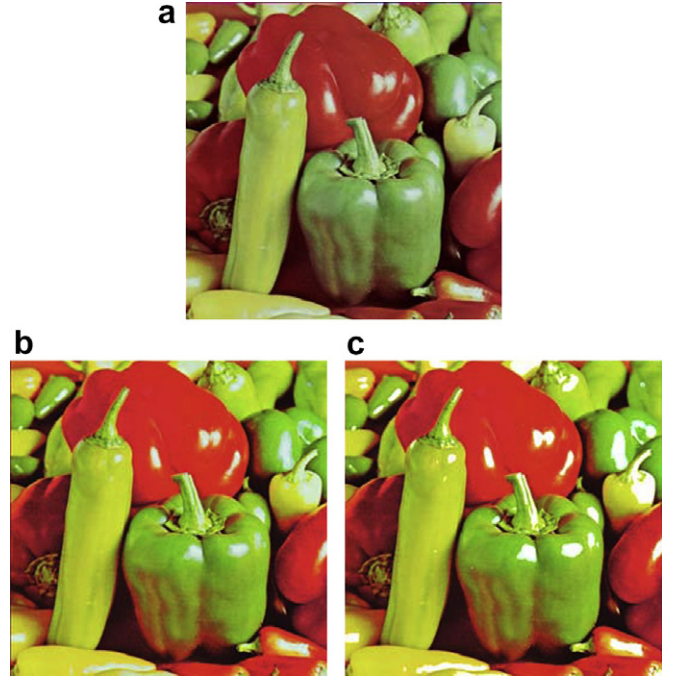


Fig. 6. The color contrast enhancement results for Peppers image. (a) Original image. (b) The enhanced image by using the previous algorithm. (c) The enhanced image by using our proposed algorithm.

ous color contrast enhancement algorithm. Fig. 6(c) illustrates the enhanced color image obtained by using our proposed color contrast enhancement algorithm. It is observed that both enhanced color images, Fig. 6(b) and (c), have the same color contrast enhancement effect and they look more colorful than that in Fig. 6(a). Besides the color contrast enhancement effect, our proposed algorithm has a good compromise between the edge-preservation effect and the color contrast enhancement effect. This good compromise will be demonstrated and evaluated in Section 5. Especially, the enhanced color image obtained by using our proposed edge-preserving algorithm plays a good input role for some applications, such as the color image segmentation.

### 3.2. The speedup strategy

In this subsection, a speedup strategy is presented to improve the computational effort in the saturation step of our proposed color contrast enhancement algorithm.

In order to quickly obtain the maximally saturated color  $C_s$  when giving the input color pixel  $C$ , an  $O(1)$ -time intersection point determination strategy is presented. Considering only the two chromatic components  $u'$  and  $v'$ , first assume the intersection point between line  $\overline{WC}$  and the gamut triangle is on the segment  $\overline{B'R'}$  (see Fig. 2(a)) where the two end points  $B'$  and  $R'$  have the chromatic values  $(u'_{B'}, v'_{B'})$  and  $(u'_{R'}, v'_{R'})$ , respectively. It is known that  $W = (u'_W, v'_W)$  is the white point. According to this assumption, we have the following two equalities:

$$\begin{aligned} u'_W + \tau(u'_C - u'_W) &= u'_{R'} + \lambda(u'_{B'} - u'_{R'}) \\ v'_W + \tau(v'_C - v'_W) &= v'_{R'} + \lambda(v'_{B'} - v'_{R'}). \end{aligned} \quad (4)$$

Eq. (4) can be rewritten by

$$\begin{aligned} \tau(u'_C - u'_W) - \lambda(u'_{B'} - u'_{R'}) &= u'_{R'} - u'_W \\ \tau(v'_C - v'_W) - \lambda(v'_{B'} - v'_{R'}) &= v'_{R'} - v'_W. \end{aligned} \quad (5)$$

After solving Eq. (5), if the values of  $\tau$  and  $\lambda$  satisfy  $\tau \geq 1$  and  $0 \leq \lambda \leq 1$ , the above assumption, i.e. the intersection point between



line  $\overrightarrow{WC}$  and the gamut triangle, is true. If so, the chromatic components  $u'_{C_s}$  and  $v'_{C_s}$  of the maximally saturated color  $C_s$  can be computed by the following equation:

$$\begin{aligned} u'_{C_s} &= u'_W + \tau(u'_C - u'_W) \\ v'_{C_s} &= v'_W + \tau(v'_C - v'_W). \end{aligned}$$

Otherwise, if the values of  $\tau$  and  $\lambda$  violate the conditions,  $\tau \geq 1$  and  $0 \leq \lambda \leq 1$ , we try to examine next assumption: assume the intersection point between line  $\overrightarrow{WC}$  and the gamut triangle is on the segment  $\overrightarrow{B'G'}$ , and so on.

Besides the above strategy to speedup the determination of point  $C_s$  in saturation step, now our proposed history-aid strategy is presented to predict the most possible side of the gamut triangle such that the intersection point of  $\overrightarrow{WC}$  and the gamut triangle is on that side. In general, the intersection point between  $\overrightarrow{WC}$  and the gamut triangle may be on the side  $\overrightarrow{R'B'}$ ,  $\overrightarrow{B'G'}$ , or  $\overrightarrow{G'R'}$ . Heuristically, each side of the gamut triangle must be examined one-by-one to test whether the intersection point is on that side or not. However, due to the color locality property, especially in smooth regions, moving the point  $C$  forward along the line  $\overrightarrow{WC}$  will usually intersect the same side which has been determined in the iteration. If the proposed history-aid strategy does not work, the other two sides of gamut triangle are further checked. Experimental results show that the proposed  $O(1)$ -time intersection point determination strategy and the history-aid strategy have a good computation-saving effect and in average 20%  $\left( = \frac{\text{old time} - \text{new time}}{\text{old time}} \times 100\% \right)$  execution-time improvement ratio is obtained when compared to the heuristic approach for determining the side of gamut triangle. Besides the  $u'v'$  chromatic domain in the gamut triangle, luminance ( $Y$ ) channel may be considered in our proposed history-aid strategy.

Based on the four testing images, the Pepper image, F14 image, Table Tennis image, and Akyio image (see Section 5), in average, the total execution-time required in the previous color contrast enhancement algorithm to map the RGB color domain into the CIE  $Lu'v'$  domain is 0.149 s. Based on the same testing images, in average, the total execution-time required in our proposed edge-preserving algorithm without considering the binary alternative search strategy and the history-aid strategy is 0.788 s. Employing the two speedup strategies in our proposed edge-preserving algorithm is 0.347 s. In summary, the time required in any one of the above three concerned algorithms is much less than 1 s. Once the previous enhanced color images and our obtained enhanced color images have been obtained, they can be reused in any color image operations such as edge detection, segmentation, etc. Based on the same testing images, experimental results demonstrate that in average, the execution-time required in color image segmentation (see next section) is more than 3 s. Therefore, the time required in color contrast enhancement could be ignored when compared to the ones in color image operations.

#### 4. Color image segmentation application

In this section, our proposed novel color image segmentation algorithm, which uses the enhanced color image obtained in Section 3 to be the input image, is presented. In Section 5, some experiments will be carried out to demonstrate the edge-preserving benefit in our proposed color image segmentation algorithm.

Our proposed color image segmentation algorithm consists of two major phases, namely the seed-based region growing phase and the merging phase. Maybe some other color segmentation algorithms [30,12,6] also have better segmentation results on our obtained enhanced color image when compared to those on the previous obtained enhanced color image without edge-preservation.

#### 4.1. The seed-based region growing phase

Our proposed seed-based region growing phase is modified from the previous ones in [1,7,16,24]. In what follows, two major stages of the seed-based region growing phase, namely the seed determination stage and the region growing stage, are presented.

##### 4.1.1. The seed determination stage

The seeds used in the seed-based region growing phase are the homogeneous pixels of the input color image and are selected when their homogeneity levels are greater than the threshold. The homogeneity level of one pixel in the image is determined by its standard deviation and discontinuity of the subimage covered by the  $w \times w$  window mask. Let  $C_{ij}$  denote the color information of pixel  $P_{ij}$  at location  $(i,j)$  in the image and the window mask be centered at  $(i,j)$ . The standard deviation of each component of  $P_{ij}$  can be obtained by

$$\sigma_{C_{ij}} = \sqrt{\frac{1}{w^2} \sum_{m=i-\frac{(w-1)}{2}}^{i+\frac{(w-1)}{2}} \sum_{n=j-\frac{(w-1)}{2}}^{j+\frac{(w-1)}{2}} (C_{mn} - \mu_{ij})^2}$$

where  $C_{ij} \in \{Y, u', v'\}$  and

$$\mu_{ij} = \sum_{m=i-\frac{(w-1)}{2}}^{i+\frac{(w-1)}{2}} \sum_{n=j-\frac{(w-1)}{2}}^{j+\frac{(w-1)}{2}} C_{mn}.$$

The discontinuity of pixel  $P_{ij}$  is measured by its neighboring edge information. By Eq. (3), the calculated value of MVD is assigned to  $v_{T_{ij}}$ , i.e.

$$v_{T_{ij}} = \text{MVD}.$$

Let

$$\begin{aligned} \sigma_{P_{ij}} &= \frac{\sigma_{T_{ij}}}{\sigma_{\max}} \\ v_{P_{ij}} &= \frac{v_{T_{ij}}}{v_{\max}} \end{aligned}$$

where  $\sigma_{T_{ij}} = \sigma_Y + \sigma_{u'} + \sigma_{v'}$ ,  $\sigma_{\max} = \text{Max}\{\sigma_{T_{ij}}\}$ , and  $v_{\max} = \text{Max}\{v_{T_{ij}}\}$ . It is easy to check that the values of  $\sigma_{P_{ij}}$  and  $v_{P_{ij}}$  are within the range  $[0,1]$ . When  $\sigma_{P_{ij}}$  and  $v_{P_{ij}}$  are large, the pixel  $P_{ij}$  could be defined to be a non-homogenous pixel. Therefore, the homogeneity level of pixel  $P_{ij}$  is defined as

$$H_{P_{ij}} = 1 - (\sigma_{P_{ij}} \times v_{P_{ij}}).$$

When the value of  $H_{P_{ij}}$  for pixel  $P_{ij}$  at location  $(i,j)$  is greater than the specified threshold  $T_H$ , the pixel  $P_{ij}$  is selected to a seed which will be used in the second stage of this phase. Let  $T_H = 0.99$ , the blue color pixels shown in Fig. 7 are the determined seeds in the enhanced color Peppers image as shown in Fig. 6(b).

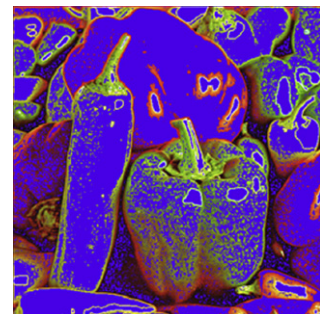


Fig. 7. The determined seeds as shown in blue colors of the enhanced Peppers image.

#### 4.1.2. The region growing stage

After determining the seeds from the enhanced color image, the region growing stage is followed to gather the relevant pixels for each region. The region growing stage consists of the following four steps:

- Step 1:** Connect the adjacent seeds and assign the same region number to them. That is, adjacent seeds are treated as a large seed of the image.
- Step 2:** Examine the connected neighbors of the seeds and mark the pixels which are not seeds. Then, compute the color contrast between each marked pixel and its adjacent regions, each region only consisting of seeds. The color contrast level between the marked pixel  $P_m$  and its adjacent region  $R_i$  is measured by
- $$C(P_m, R_i) = \frac{\sqrt{(Y_{P_m} - \bar{Y}_{R_i})^2 + (u'_{P_m} - \bar{u}'_{R_i})^2 + (v'_{P_m} - \bar{v}'_{R_i})^2}}{\sqrt{Y_{P_m}^2 + u'^2_{P_m} + v'^2_{P_m}}}$$
- where  $\bar{Y}_{R_i}$ ,  $\bar{u}'_{R_i}$ , and  $\bar{v}'_{R_i}$  are the mean values of the three CIE components of region  $R_i$ .
- Step 3:** Select the minimum one among these calculated color contrast levels and its associated  $P_m$  and  $R_i$ . Then, put the associated marked pixel  $P_m$  into the region  $R_i$  and remark the connected neighbors of  $P_m$ .
- Step 4:** According to the color contrast level measure defined in Step 1, update the color contrast level between the updated region  $R_i$  and its adjacent marked pixels. Then, repeat Step 3 until all pixels in the image have found their regions.

The above seed-based region growing phase may over-segment the enhanced color image. After performing the above seed-based region growing phase on the enhanced Peppers image as shown in Fig. 6(b), Fig. 8 illustrates the related over-segmented result. Due to this over-segmentation problem, it is necessary to merge the similar regions in the over-segmented image further. In next subsection, our proposed merging phase is presented to alleviate this over-segmentation problem. In our proposed merging phase, the fruitful edge information preserved in the enhanced color image, which has been obtained by using our proposed edge-preserving algorithm for color contrast enhancement, can result in a more desirable segmentation result.

#### 4.2. The merging phase

This subsection presents our proposed novel merging phase. Let  $R_i$  and  $R_j$  be two adjacent regions. For measuring the difference between the two adjacent regions  $R_i$  and  $R_j$ , we first compute the dif-



**Fig. 8.** The over-segmented result after performing the seed-based region growing phase on the enhanced Peppers image in Fig. 6(b).

ference between the centroid and the geometric center of  $R_k$  ( $= R_i \cup R_j$ ), the color moment difference between  $R_i$  and  $R_j$ , and the boundary edge information between  $R_i$  and  $R_j$ . Since the enhanced color image obtained by our proposed edge-preserving algorithm for color contrast enhancement has more true edge information, we believe that our proposed merging phase used in color image segmentation does inherit the edge-preserving benefit and it will lead to better color image segmentation result.

The centroid of  $R_k$  ( $= R_i \cup R_j$ ) [8] is given by

$$(C_{R_k,x}, C_{R_k,y}) = \left( \frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

where  $m_{10} = \sum \sum_{(x,y) \in R_k} x f(x,y)$ ,  $m_{01} = \sum \sum_{(x,y) \in R_k} y f(x,y)$ , and  $m_{00} = \sum \sum_{(x,y) \in R_k} f(x,y)$ ;  $f(x,y)$  denotes the relevant three components of the enhanced color image. The geometric center of  $R_k$  is given by

$$(GC_{R_k,x}, GC_{R_k,y}) = \left( \frac{m_{10}^{CC}}{m_{00}^{CC}}, \frac{m_{01}^{CC}}{m_{00}^{CC}} \right)$$

where  $m_{10}^{CC} = \sum \sum_{(x,y) \in R_k} x f^0(x,y)$ ,  $m_{01}^{CC} = \sum \sum_{(x,y) \in R_k} y f^0(x,y)$ , and  $m_{00}^{CC} = \sum \sum_{(x,y) \in R_k} f^0(x,y)$ . From the above defined centroid and geometric center, the difference between the centroid and geometric center of  $R_k$  is measured by

$$D_{GC}^C(R_i, R_j) = \|(GC_{R_k,x}, GC_{R_k,y}) - (C_{R_k,x}, C_{R_k,y})\|. \quad (6)$$

The larger the difference between the centroid and geometric center of  $R_k$  is, the more improper the merge of  $R_i$  and  $R_j$  is.

Besides the difference between the centroid and geometric center of  $R_k$ , the color difference between  $R_i$  and  $R_j$  is also employed to be a factor in the proposed merging phase. Based on the color domain histogram moments [15,18,27], our proposed color domain difference between  $R_i$  and  $R_j$  is given by

$$CD(R_i, R_j) = \|\text{CC}_{R_i} - \text{CC}_{R_j}\| \quad (7)$$

where  $\text{CC}_R = (\text{CC}_{R,u'}, \text{CC}_{R,v'}) = \left( \frac{M_{00}^R}{M_{00}^R}, \frac{M_{01}^R}{M_{00}^R} \right)$ ,  $M_{pq}^R = \sum \sum_{(u',v') \in R} u'^p v'^q D_H^R(u', v')$ , and  $D_H^R(u', v')$  denotes the number of pixels in the RGB color image where these related pixels are mapped to the same region  $R$  in  $(u', v')$ -domain. In Eq. (7),  $\text{CC}_R = (\text{CC}_{R,u'}, \text{CC}_{R,v'})$  denotes the color domain centroid of region  $R$ ;  $M_{pq}^R$  denotes the color histogram moment of region  $R$ ;  $D_H^R(u', v')$  denotes the relevant two-dimension histogram.

Besides Eqs. (6) and (7), the boundary edge information between  $R_i$  and  $R_j$  [17] is definitely used to measure the difference of two adjacent regions in the proposed merging phase. It has been defined that the edge map obtained from the enhanced color image  $I$  is denoted by  $I_E$  where  $I_E(x,y) = 1$  ( $= 0$ ) means the color pixel  $I(x,y)$  is an edge (non-edge) pixel. Let  $B_{(R_i, R_j)}$  be the set of pixel locations which constitute the positions of boundary between two regions  $R_i$  and  $R_j$ . The ratio of the edge information occupied on the boundary can be expressed by

$$\gamma_E(R_i, R_j) = \frac{\sum_{(x,y) \in B_{(R_i, R_j)}} I_E(x,y)}{|B_{(R_i, R_j)}|} \quad (8)$$

where  $|B_{(R_i, R_j)}|$  denotes the number of pixels on the boundary  $B_{(R_i, R_j)}$ .

By Eqs. (6)–(8), when the values of  $D_{GC}^C(R_i, R_j)$ ,  $CD(R_i, R_j)$  and  $\gamma_E(R_i, R_j)$  are larger, the two adjacent regions  $R_i$  and  $R_j$  are more different from each other. Integrating the above three difference measures, the difference between two adjacent regions  $R_i$  and  $R_j$  can be defined by

$$D_T(R_i, R_j) = w_1 \times \frac{D_{GC}^C(R_i, R_j)}{D_{GC}^C(\text{max})} + w_2 \times \frac{CD(R_i, R_j)}{CD(\text{max})} + w_3 \times \gamma_E(R_i, R_j) \quad (9)$$

where  $D_{GC(max)}^C = \text{Max}_{i \neq j} \{D_{GC}^C(R_i, R_j)\}$  and  $CD_{(max)} = \text{Max}_{i \neq j} \{CD(R_i, R_j)\}$ ;  $\frac{D_{GC}^C(R_i, R_j)}{D_{GC(max)}^C}$ ,  $\frac{CD(R_i, R_j)}{CD_{(max)}}$ , and  $\gamma_E(R_i, R_j)$  are within the range  $[0, 1]$ ; empirically, the three weights are set to  $w_1 = 1$ ,  $w_2 = 1.8$ , and  $w_3 = 2.5$ .

Based on the above description, in summary, our proposed merging phase consists of the following three steps:

- Step 1: For each region  $R_i$ , consider its all adjacent regions  $R_j$ 's and compute all the differences  $D_T(R_i, R_j)$ s by Eq. (9).
- Step 2: Select the minimum difference  $D_T(R_i, R_j)$  among these calculated  $D_T(R_i, R_j)$ s obtained from Step 1. When the selected minimal  $D_T(R_i, R_j)$  is greater than the specified threshold or the current segmented image only contains a region, output the segmentation result and stop this phase. Otherwise, go to Step 3.
- Step 3: Merge  $R_i$  and  $R_j$  into a region. After merging  $R_i$  and  $R_j$ , by Eq. (9), update all the related differences  $D_T(R_i, R_k)$ s and  $D_T(R_j, R_k)$ s for all the concerned two adjacent region-pairs  $(R_i, R_k)$  and  $(R_j, R_k)$ . Go to Step 2.

After presenting two phases of our proposed color segmentation algorithm, next section will illustrate the segmentation result by running our proposed color segmentation algorithm on our obtained enhanced color image and compare it with the segmentation result by running our proposed color segmentation algorithm on the enhanced color image obtained by the previous color contrast enhancement algorithm.

### 5. Experimental results

In this section, some experimental results are demonstrated to show that our proposed algorithm has a good compromise between the edge-preservation effect and the color contrast enhancement effect when compared to the previous algorithm [13,20]. Besides, some experiments are carried out to demonstrate the edge-preservation benefit of our proposed color image segmentation algorithm when running it on the enhanced color image obtained by the proposed color contrast enhancement algorithm. For convenience, the enhanced color image obtained by the previous color contrast enhancement algorithm is called the previous obtained enhanced color image; the enhanced color image obtained by our proposed color contrast enhancement algorithm is called our obtained enhanced color image.

Fig. 9(a) illustrates the edge map of the original Peppers image shown in Fig. 6(a). The edge map of the previous obtained enhanced color Peppers image is illustrated in Fig. 9(b). The edge map of our obtained enhanced color Peppers image is illustrated in Fig. 9(c) and it is observed that our obtained edge map of each pepper is quite similar to that in the original edge map (see Fig. 9(a)). However, the edge map in Fig. 9(b) is some different from the original edge map. Quantitative demonstrations (see Figs. 11, 12, and Tables 1 and 2) will be given later to explain why Fig. 9(c) is better than Fig. 9(b). Besides the color contrast enhancement effect existed in the previous color contrast enhancement algorithm and our proposed one (see Fig. 6(b) and (c)), the edge-preservation effect of our proposed color contrast enhancement algorithm is better than that of the previous enhancement algorithm.

After demonstrating the color contrast enhancement benefit of both algorithms and the edge-preservation benefit of our proposed algorithm over the previous algorithm, let us take F14 color image (see Fig. 10(a)) as the second testing image. Fig. 10(b)–(f) demonstrate the edge map of the original color F14 image after running the previous color edge detector, the previous obtained enhanced color image, the edge map of the previous obtained enhanced

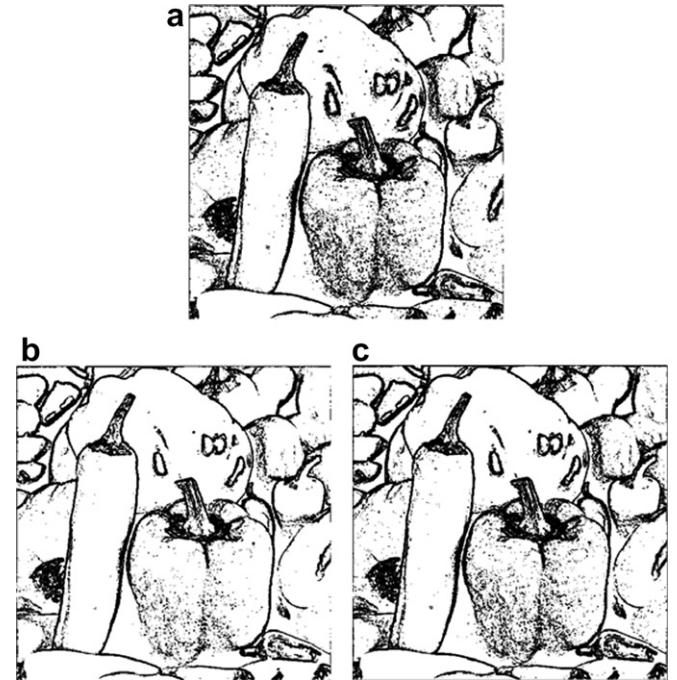


Fig. 9. The edge maps of color Peppers images. (a) The edge map of original image. (b) The edge map of the previous obtained enhanced image. (c) The edge map of our obtained enhanced image.

Table 1  
Edge-loss ratios for the concerned two color contrast enhancement algorithms

	Previous algorithm	Our proposed algorithm
Peppers (%)	31.17	9.34
F14 (%)	31.69	12.92
Table Tennis (%)	23.93	4.75
Akiyo (%)	18.72	5.10
Average (%)	24.78	7.59
Improvement ratio (%)	69.37	

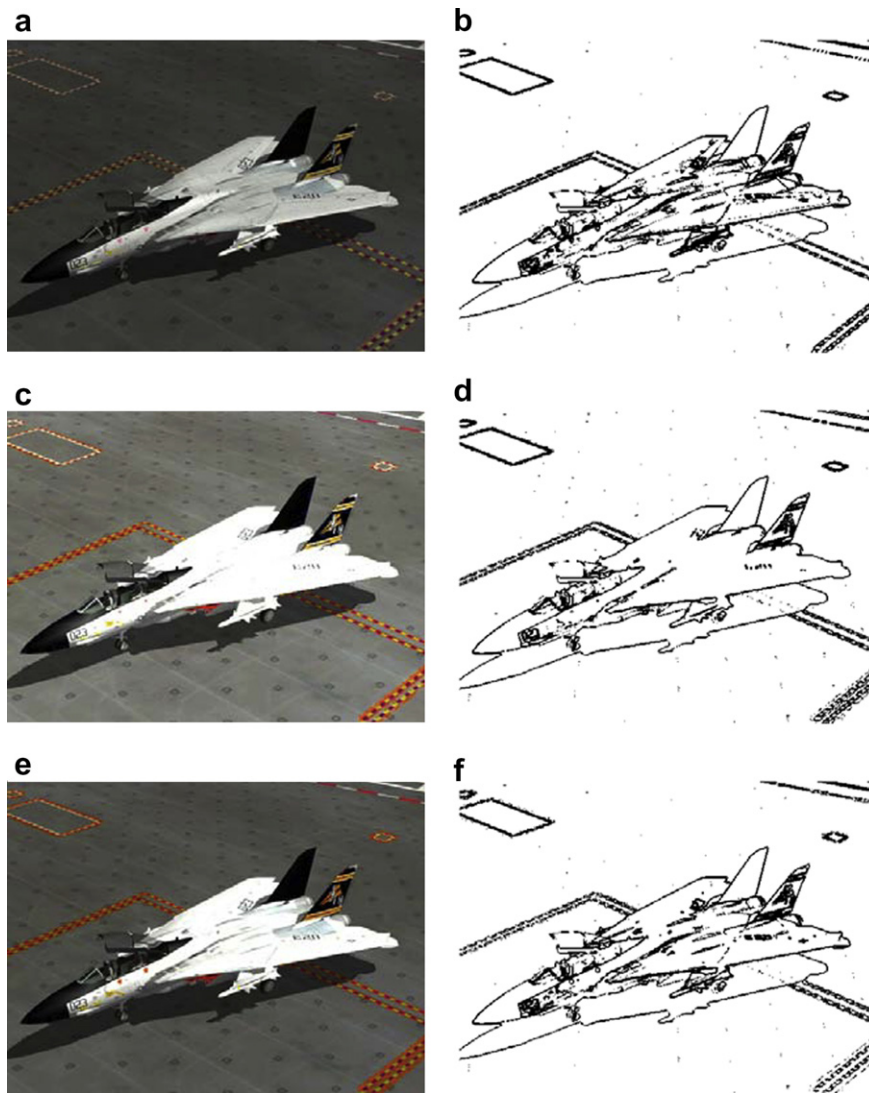
Table 2  
Edge-add ratios for the concerned two color contrast enhancement algorithms

	Previous algorithm	Our proposed algorithm
Peppers (%)	10.18	8.92
F14 (%)	7.08	5.15
Table Tennis (%)	11.77	11.63
Akiyo (%)	15.30	14.27
Average (%)	11.08	9.99
Improvement ratio (%)	9.84	

image, our obtained enhanced color image, and the edge map of our obtained enhanced image, respectively. Similar to the performance evaluation for color Peppers image, experimental results for color F14 image also reveal that the previous and our proposed color contrast enhancement algorithms make the original F14 image more colorful. In addition, our proposed enhancement algorithm has better edge-preservation effect when compared to the previous enhancement algorithm.

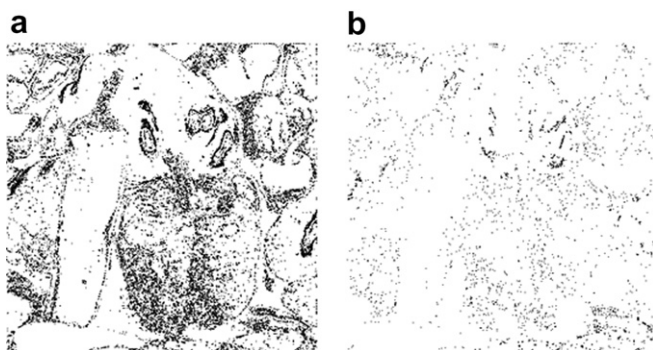
Next, we adopt the set of edge-losing pixels and the set of edge-adding pixels to demonstrate the edge-preservation effect of our proposed color contrast enhancement algorithm. It has been defined that the edge map of the original image is expressed by  $I_E$  and  $I_E(x, y) = 1$  ( $= 0$ ) denotes that the color pixel  $I(x, y)$  is an edge (non-edge) pixel. Let  $I_{EE}$  denote the edge map when running the previous color edge detector on the related enhanced color image.





**Fig. 10.** The enhancement demonstration for color F14 images. (a) Original color F14 image. (b) The edge map of image (a). (c) The previous obtained enhanced image. (d) The edge map of image (c). (e) Our obtained enhanced image. (f) The edge map of image (e).

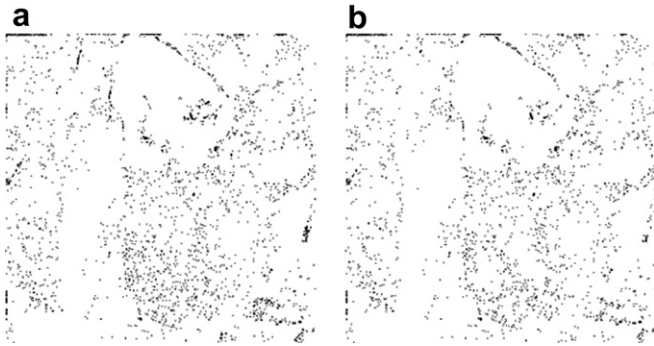
The set of edge-losing pixels in the resulting edge map is denoted by  $N_{el}$ . Each pixel in the set of edge-losing pixels must satisfy  $I_E(x, y) = 1$  and  $I_{EE}(x, y) = 0$ . On the other hand, one pixel is called the edge-losing pixel when the pixel is an edge pixel in the original color image, but the pixel at the same position in the enhanced image is a non-edge image pixel. Alternatively, the set of edge-adding pixels in the resulting edge map is denoted by  $N_{ea}$ , and each pixel in the set of edge-adding pixels must satisfy  $I_E(x, y) = 0$  and  $I_{EE}(x, y) = 1$ . Fig. 11(a) and (b) illustrate the set of edge-losing pixels  $N_{el}$  for the edge map of the previous obtained enhanced color Peppers image (see Fig. 9(b)) and the edge map of our obtained enhanced color Peppers image (see Fig. 9(c)), respectively. Fig. 12(a) and (b) illustrate the set of edge-adding pixels  $N_{ea}$  for the edge map of the previous obtained enhanced color Peppers image and the edge map of our obtained enhanced color Peppers image, respectively. By Figs. 11 and 12, our obtained enhanced color image has less edge-losing pixels and edge-adding pixels when compared to the previous obtained enhanced color image. Then, the set of edge-losing pixels  $N_{el}$  for the edge map of the previous obtained enhanced color F14 image (see Fig. 10(d)) and the edge map of our obtained enhanced color F14 image (see Fig. 10(f)) are illustrated in Fig. 13(a) and (b), respectively; the set of edge-adding pixels



**Fig. 11.** The set of edge-losing pixels  $N_{el}$  for color Peppers image. (a)  $N_{el}$  for the previous obtained enhanced color Peppers image. (b)  $N_{el}$  for our obtained enhanced color Peppers image.

$N_{ea}$  for the edge map of the previous obtained enhanced color F14 image and the edge map of our obtained enhanced color F14 image are illustrated in Fig. 14(a) and (b), respectively. From the above demonstrations, our proposed color contrast enhancement





**Fig. 12.** The set of edge-adding pixels  $N_{ea}$  for color Peppers image. (a)  $N_{ea}$  for the previous obtained enhanced color Peppers image. (b)  $N_{ea}$  for our obtained enhanced color Peppers image.

algorithm has better edge-preservation effect when compared to the previous color contrast enhancement algorithm.

In order to quantify the edge-preservation effect of our proposed color contrast enhancement algorithm, the edge-loss ratio and edge-addition ratio are defined. The edge-loss ratio  $R_{el}$  is defined as

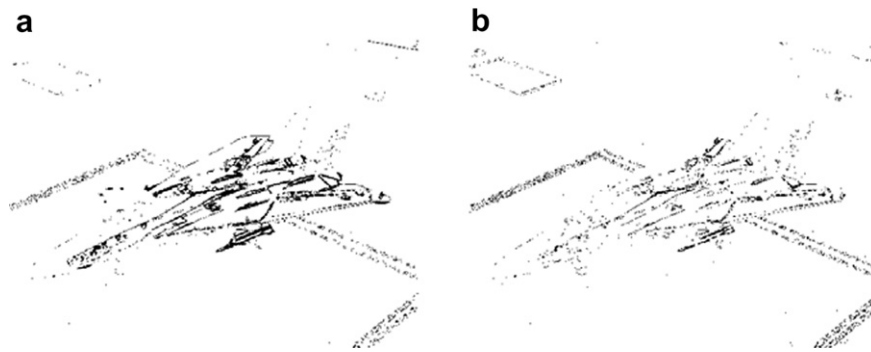
$$R_{el} = \frac{|N_{el}|}{|I_E(x,y) = 1|}$$

where  $|N_{el}|$  denotes the number of edge-losing pixels in the resulting edge map;  $|I_E(x,y) = 1|$  denotes the number of edge pixels in the original image. Contrary to the defined edge-loss ratio, the edge-addition ratio  $R_{ea}$  is defined as

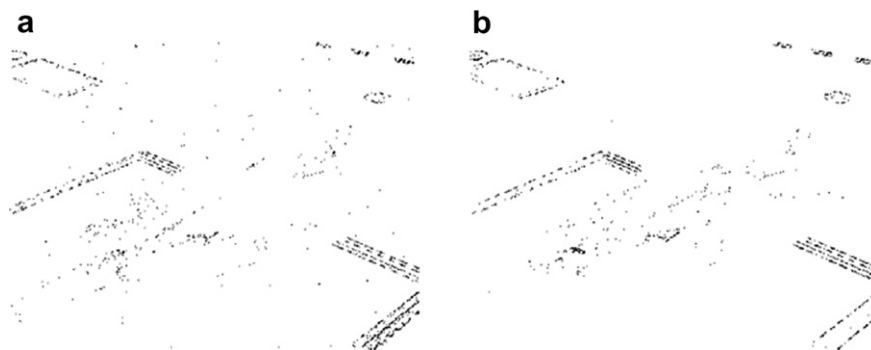
$$R_{ea} = \frac{|N_{ea}|}{|I_E(x,y) = 1|}$$

where  $|N_{ea}|$  denotes the number of edge-adding pixels in the resulting edge map. To compare the edge-loss ratios between the concerned two algorithms, four testing images are used. Besides the color Peppers image and the color F14 image, the other two testing images, the color Table Tennis image and the color Akiyo image as shown in Fig. 15(a) and (b), respectively, are used. Based on the four testing images, Table 1 demonstrates the edge-loss ratios for the concerned two algorithms. For both enhancement algorithms, in average, our proposed enhancement algorithm only has 7.59% edge-loss ratio, but the edge-loss ratio of the previous enhancement algorithm is 24.78%. The edge-loss improvement ratio of our proposed color contrast enhancement algorithm over the previous enhancement algorithm is 69.37%. Table 2 demonstrates the edge-addition ratios for the concerned two algorithms. Based on four testing images, the average edge-addition ratio of our proposed enhancement is 9.99% and that of the previous enhancement algorithm is 11.08%. The edge-addition improvement ratio of our proposed color contrast enhancement algorithm over the previous enhancement algorithm is 9.84%

Further, we take the color Peppers image to depict the visual edge-preserving effect. Fig. 16(a) illustrates the segmentation result when running our proposed segmentation algorithm on the original image. Fig. 16(b) illustrates the magnified subimage cut from the left portion of one red pepper in Fig. 16(a) and the edge map of Fig. 16(b) is illustrated in Fig. 16(c). Fig. 16(d) and (e) illustrate the segmentation results when running our proposed segmentation algorithm on the previous obtained enhanced color image and running the same proposed segmentation algorithm on our obtained enhanced color image which has the edge-preserving effect, respectively. Fig. 16(f) illustrates the magnified subimage cut from the left portion of one red pepper in Fig. 16(d) and the edge map of Fig. 16(f) is illustrated in Fig. 16(g). The magnified subimage cut from the same portion of the same red pepper in



**Fig. 13.** The set of edge-losing pixels  $N_{el}$  for color F14 image. (a)  $N_{el}$  for the previous obtained enhanced color F14 image. (b)  $N_{el}$  for our obtained enhanced color F14 image.



**Fig. 14.** The set of edge-adding pixels  $N_{ea}$  for color F14 image. (a)  $N_{ea}$  for the previous obtained enhanced image. (b)  $N_{ea}$  for our obtained enhanced image.

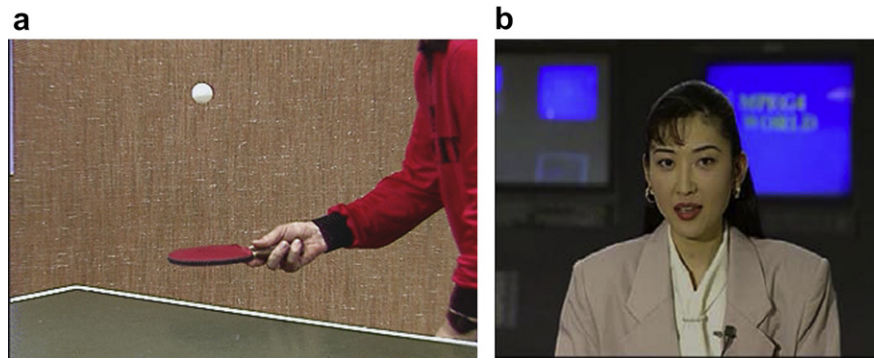


Fig. 15. The other two testing images. (a) The color Table Tennis image. (b) The color Akiyo image.

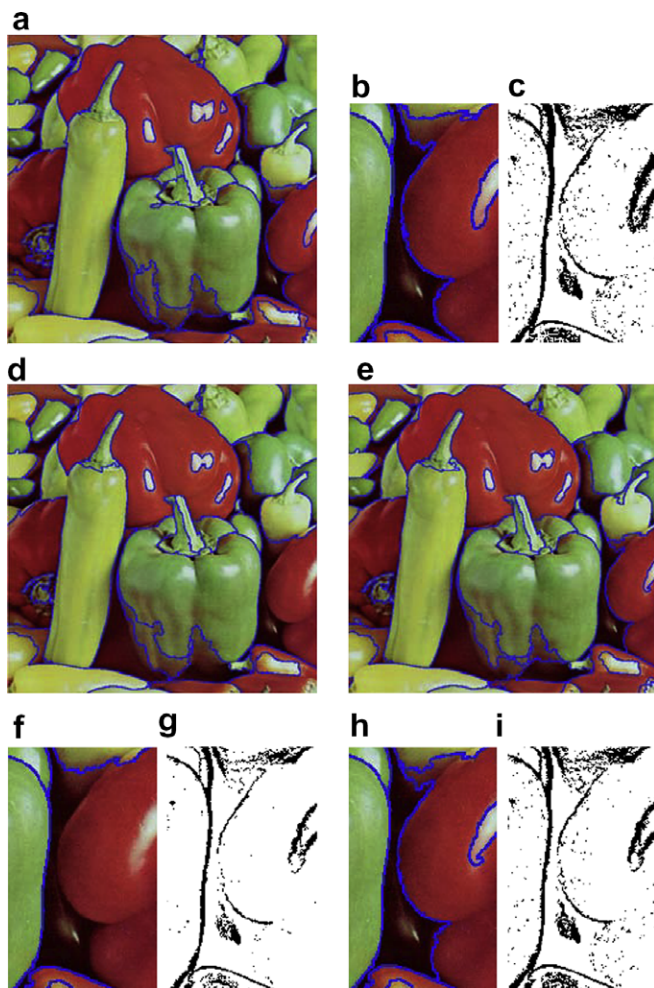


Fig. 16. The segmentation comparison for original Peppers image and two kinds of enhanced Peppers images. (a) The segmentation result when running the proposed segmentation algorithm on original image. (b) One magnified subimage extracted from (a). (c) The edge map of (b). (d) The segmentation result when running the proposed segmentation algorithm on the previous obtained enhanced color image. (e) The segmentation result when running the proposed segmentation algorithm on our obtained enhanced color image. (f) One magnified subimage extracted from (d). (g) The edge map of (f). (h) The magnified subimage extracted from (e). (i) The edge map of (h).

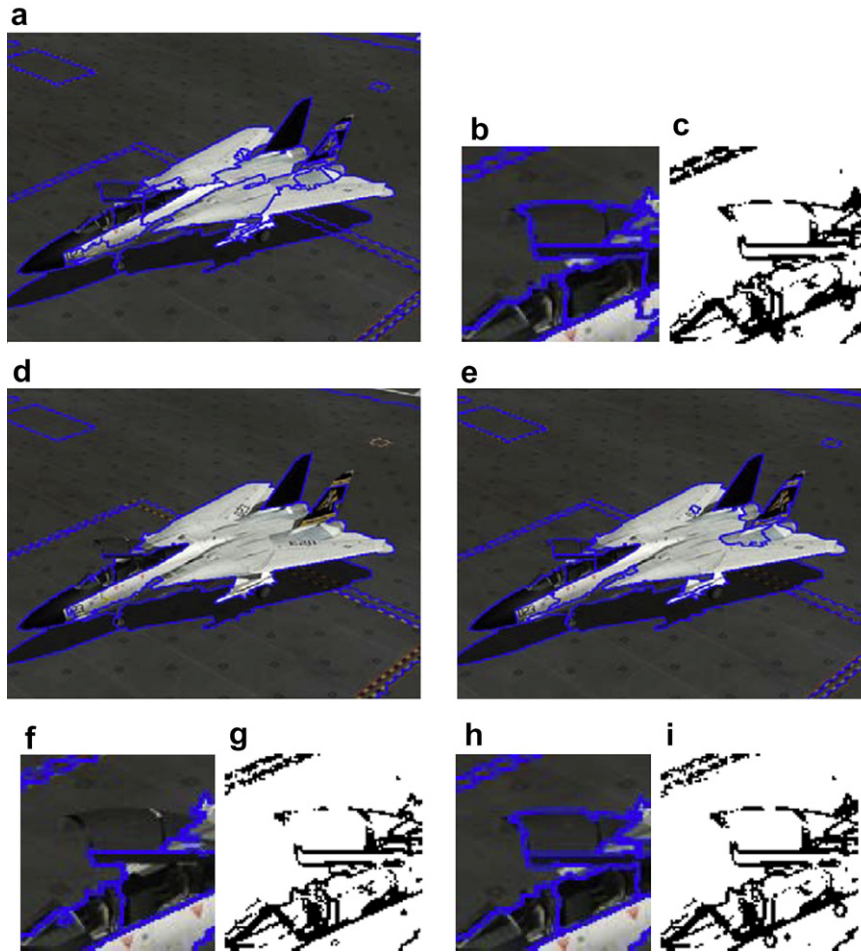
Fig. 16(e) is illustrated in Fig. 16(h) and (i) depicts the edge map of Fig. 16(h). After comparing the three edge maps of magnified sub-images (see Fig. 16(c), (g), and (i)), it is observed that the edge map Fig. 16(i) is quite similar to Fig. 16(c), but Fig. 16(g) is some differ-

ent from Fig. 16(c). In addition, the density and the distribution of edge pixels of Fig. 16(i) are rather close to those in Fig. 16(c), but the density of edge pixels of Fig. 16(g) is some sparse and is much less than that in Fig. 16(c). Tables 1 and 2 reveal that for the color Peppers image, the edge-loss improvement ratio and edge-addition improvement ratio of our proposed color contrast enhancement algorithm over the previous enhancement algorithm are 70.04% ( $= \frac{0.3117-0.0934}{0.3117} \times 100\%$ ) and 12.38% ( $= \frac{0.1018-0.0892}{0.1018} \times 100\%$ ), respectively, and it justifies the edge-preserving effect of our proposed color contrast enhancement algorithm. After comparing Fig. 16(h) with Fig. 16(b) and comparing Fig. 16(f) with Fig. 16(b), it is observed that our segmentation result when running the proposed segmentation algorithm on our obtained enhanced color image (see Fig. 16(h)) is better than the segmentation result (see Fig. 16(f)) when running the proposed segmentation algorithm on the previous obtained enhanced color image. Especially, the segmented boundary of the right pepper in Fig. 16(h) is more clear than that in Fig. 16(f) because our color contrast enhancement improves the segmentation quality.

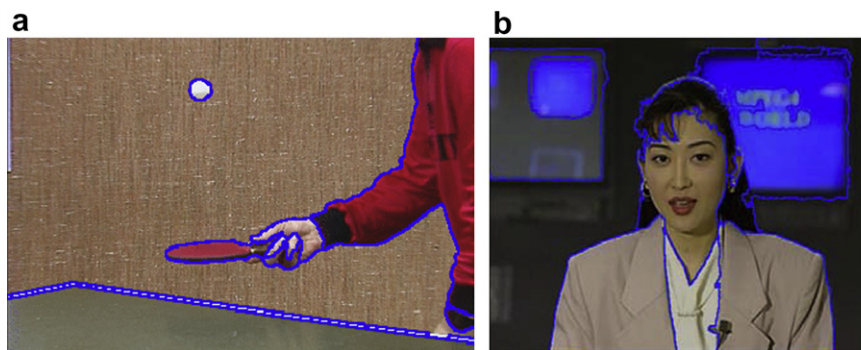
Further, the color F14 image is used to be the second testing image for segmentation comparison when running the proposed segmentation algorithm on the original F14 image and two kinds of enhanced F14 images. From the cockpit part of Fig. 17, it is observed that the number of preserved edge pixels and the distribution of edge pixels in Fig. 17(i) are quite similar to Fig. 17(c). However, the number of preserved edge pixels in Fig. 17(i) is greater than the one in Fig. 17(g). Tables 1 and 2 provide the qualitative evidences. After examining Fig. 17(a), (d), and (e), it is observed that the segmentation result in Fig. 17(e) is quite similar to Fig. 17(a) and is better than that in Fig. 17(d). This confirms that the edge-preserving effect of our proposed color contrast enhancement improves the segmentation quality. For color Table Tennis image and color Akiyo image, we have the same segmentation quality improvement due to our proposed color contrast enhancement. Finally, Fig. 18(a) and (b) demonstrate the good segmentation results when running our proposed color segmentation algorithm on our obtained enhanced color Table Tennis image and color Akiyo image, respectively. For saving space, we ignore the elaborated comparison.

## 6. Conclusions

In this paper, a novel and efficient edge-preserving algorithm has been presented for color contrast enhancement in the CIE Lu'v' color space although our proposed algorithm can be applied to the other color spaces, such as the CIE xyY color space. In order to improve the computational effort of the proposed algorithm, a speedup strategy has also been given. To the best of our knowledge, this is the first edge-preserving algorithm for color contrast enhancement in color space.



**Fig. 17.** The segmentation comparison for original F14 image and two kinds of enhanced F14 images. (a) The segmentation result when running the proposed segmentation algorithm on original image. (b) One magnified subimage extracted from (a). (c) The edge map of (b). (d) The segmentation result when running the proposed segmentation algorithm on the previous obtained enhanced color image. (e) The segmentation result when running the proposed segmentation algorithm on our obtained enhanced color image. (f) One magnified subimage extracted from (d). (g) The edge map of (f). (h) The magnified subimage extracted from (e). (i) The edge map of (g).



**Fig. 18.** The segmentation results when running our proposed color segmentation algorithm on our obtained enhanced color images. (a) The segmentation result for color Table Tennis image. (b) The segmentation result for color Akiyo image.

Further, a new color image segmentation has been presented to justify the edge-preservation effect. Some experimental results have been carried out to demonstrate that our proposed color contrast enhancement algorithm has a good compromise between the edge-preservation effect and the color contrast enhancement effect when compared to the previous algorithm. Besides, experimental results also confirms that the edge-preserving effect of our proposed color contrast enhancement improves the segmentation quality.

It is an interesting research topic to apply the results of this paper to the field of color image retrieval issue [5,9,10,19] when the relevant retrieval techniques include the consideration of edge information.

**References**

[1] R. Adams, L. Bischof, Seeded region growing, *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (1994) 641–647.



- [2] J. Astola, P. Haavisto, Y. Neuvo, Vector median filters, in: Proc. of the IEEE, April 1990, pp. 678–689.
- [3] V. Barnett, The ordering of multivariate data, *J. Roy. Stat. Soc. A* 139 (1976) 318–343.
- [4] E.J. Candes, D.L. Donoho, Curvelets—a surprisingly effective nonadaptive representation for objects with edges, in: T.N. Nashville (Ed.), *Curve and Surface Fitting: Saint-Malo*, Vanderbilt University Press, 1999.
- [5] D.Y. Chen, H.Y.M. Liao, S.Y. Lee, Robust video sequence retrieval using a novel object-based T2D-histogram descriptor, *J. Vis. Commun. Image Rep.* 16 (2005) 212–232.
- [6] J. Chen, T.N. Pappas, A. Mojsilovic, B.E. Rogowitz, Adaptive perceptual color-texture image segmentation, *IEEE Trans. Image Process.* 14 (2005) 1524–1536.
- [7] 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 Process.* 10 (2001) 1454–1466.
- [8] R.C. Gonzalez, R.E. Woods, *Digital Image Process.*, second ed., Prentice Hall, New York, 2002.
- [9] I.S. Hsieh, K.C. Fan, Multiple classifier for color image retrieval, *IEEE Trans. Image Process.* 10 (2001) 938–950.
- [10] J.W. Hsieh, W.E.L. Grimson, Spatial template extraction for image retrieval by region matching, *IEEE Trans. Image Process.* 12 (2003) 1404–1415.
- [11] R.W.G. Hunt, *Measuring Colour*, second ed., Ellis Horwood, New York, 1995.
- [12] O. Lezoray, H. Cardot, Cooperation of color pixel classification schemes and color watershed: a study for microscopic images, *IEEE Trans. Image Process.* 11 (2002) 783–789.
- [13] L. Lucchese, S.K. Mitra, J. Mukherjee, A new algorithm based on saturation and desaturation in the *xy* chromaticity diagram for enhancement and re-rendering of color images, in: Proc. Int. Conf. Image Processing, Thessaloniki, Greece, September 2001, 2001, pp. 1077–1080.
- [14] L. Lucchese, S.K. Mitra, Filtering color images in the *xyY* color space, in: Proc. Int. Conf. Image Processing, Vancouver, BC, Canada, September 2000, 2000, pp. 500–503.
- [15] M.K. Mandal, T. Aboulnasr, S. Panchanathan, Image indexing using moments and wavelets, *IEEE Trans. Consumer Electron.* 42 (1996) 557–565.
- [16] A. Mehnert, P. Jackway, An improved seeded region growing algorithm, *Pattern Recogn. Lett.* 18 (1997) 1065–1071.
- [17] E. Navon, O. Miller, A. Averbuch, Color image segmentation based on adaptive local thresholds, *Image Vis. Comput.* 23 (2005) 69–85.
- [18] G. Paschos, I. Radev, N. Prabhakar, Image content-based retrieval using chromaticity moments, *IEEE Trans. Knowledge Data Eng.* 15 (2003) 1069–1072.
- [19] S.C. Pei, C.M. Cheng, Extracting color features and dynamic matching for image data-base retrieval, *IEEE Trans. Circuits Syst. Video Technol.* 9 (1999) 501–512.
- [20] S.C. Pei, Y.C. Zeng, C.H. Chang, Virtual restoration of ancient Chinese paintings using color contrast enhancement and lacuna texture synthesis, *IEEE Trans. Image Process.* 13 (2004) 416–429.
- [21] W.K. Pratt, *Digital Image Process.*, second ed., Wiley-Interscience, 2001.
- [22] S.J. Sangwine, R.E.N. Horne (Eds.), *The Colour Image Processing Handbook*, Chapman and Hall, London, UK, 1998.
- [23] J. Scharcanski, A.N. Venetsanopoulos, Edge detection of color images using directional operators, *IEEE Trans. Circuits Syst. Video Technol.* 7 (1997) 397–401.
- [24] F.Y. Shih, S. Cheng, Automatic seeded region growing for color image segmentation, *Image Vis. Comput.* 23 (2005) 877–886.
- [25] J.L. Starck, F. Murtagh, E.J. Candes, D.L. Donoho, Gray and color image contrast enhancement by the curvelet transform, *IEEE Trans. Image Process.* 12 (2000) 706–717.
- [26] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, *IEEE Trans. Image Process.* 11 (2002) 131–141.
- [27] M. Stricker, M. Orengo, Similarity of color images, in: Proc. SPIE: Storage and Retrieval for Image and Video Databases III, 1995, pp. 381–392.
- [28] C. Theoharatos, G. Economou, S. Fotopoulos, Color edge detection using the minimal spanning tree, *Pattern Recogn.* 38 (2005) 603–606.
- [29] P.E. Trahanias, A.N. Venetsanopoulos, Color edge detection using order statistics, *IEEE Trans. Image Process.* 2 (1993) 259–264.
- [30] A. Tremeau, P. Colantoni, Regions adjacency graph applied to color image segmentation, *IEEE Trans. Image Process.* 9 (2000) 735–744.
- [31] E. Zaharescu, M. Zamfir, C. Vertan, Color morphology-like operators based on color geometric shape characteristic, in: Proc. of International Symposium on Signal Circuit and Systems, Iasi, Romania, July 2003, pp. 145–148.
- [32] S.D. Zeno, A note on the gradient of a multi-image, *Comput. Vis. Graph. Image Process.* 33 (1986) 116–125.