

國立臺灣大學電機資訊學院資訊工程學系

博士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

針對彩色影像的對比增強的邊緣保留演算法與針對色彩慮波  
陣列之去馬賽克及隨意比例尺寸調整演算法

Edge Preserving Algorithm for Color Image Enhancement and  
Demosaicing and Arbitrary-Ratio Resizing Algorithm for Color  
Filter Array

楊偉楨

Yang, Wei-Jen

指導教授： 顏文明 副教授

傅楸善 教授

Advisor: Yan, Wen-Ming, Associate Professor

Fuh, Chiou-Shann, Professor

中華民國 98 年 9 月

September, 2009

國立臺灣大學博士學位論文  
口試委員會審定書

針對彩色影像的對比增強的邊緣保留演算法與針對色彩濾波陣列之去馬賽克及隨意比例尺寸調整演算法

Edge Preserving Algorithm for Color Image Enhancement  
and Demosaicing and Arbitrary-Ratio Resizing Algorithm  
for Color Filter Array

本論文係楊偉楨君（學號F93922035）在國立臺灣大學資訊工程學系完成之博士學位論文，於民國 98 年 9 月 16 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

傅楸善

顏文明

貝蘇亭

(指導教授)

蘇明章

吳榮根

郭西川

陳祝嵩

劉長遠

李明穗

呂育道

系主任

## 中文摘要

由於彩色影像可以提供更豐富的資訊，因此在近年來彩色影像處理已經廣泛的成爲了非常重要的研究議題。在眾多彩色影像相關的研究中，以針對色彩對比增強的邊緣訊息保留、彩色去馬賽克與馬賽克影像的尺寸調整爲相當熱門的研究議題。因此，本論文將針對色彩對比增強的提出一個邊緣訊息保留演算法與探討其在影像分割上的應用，以及針對Bayer色彩濾波陣列，提出一個透過梯度測邊器與適應性的異質估計的去馬賽克演算法。而後再利用與離散餘弦轉換結合，設計一個結合去馬賽克與隨意尺寸調整演算法。

在針對色彩對比增強的邊緣訊息保留的研究議題上，在CIE Lu'v' 色彩空間下，我們提出了一個針對色彩對比增強的邊緣訊息保留演算法。該演算法不僅可以如同以往的色彩增強方法一般地增強影像的色彩對比，亦有邊緣訊息保留的效果。此外，所提出之演算法也會部份抑制因色彩對比增強而產生的邊點。該演算法是第一個針對色彩對比增強所提出之邊緣訊息保留技術。爲了驗證所提出的演算法在邊緣訊息保留上的優勢，我們又提出一個新的彩色影像分割演算法作爲驗證之應用。實驗結果顯示所提出之演算法在對比增強、邊訊息保留及彩色影像分割上，都有著相當顯著的效果。

在彩色影像去馬賽克的議題上，在不需事先解馬賽克的情形下，我們首先針對馬賽克影像提出一個能有效地擷取其梯度及邊資訊的方法。之後，基植在頻譜空間的相關性上，決定馬賽克影像中每個像素之適應性異質估計遮罩的大小。最後再結合由每個像素中所擷取出的梯度及邊資訊與適應性異質估計值，提出一個邊緣感測之去馬賽克演算法。實驗結果顯示，所提出之去馬賽克演算法比起過去數種去馬賽克演算法，有著較佳的去馬賽克的影像品質與效果。

在馬賽克影像的尺寸調整的議題上，我們提出了一個結合去馬賽克與隨意尺寸調整演算法。透過邊緣感測與色差的方式，首先將綠色平面回復。爲了降低估計

的誤差，我們透過內插色差平面的方式取代直接將紅色及藍色平面回復。之後，再透過離散餘弦轉換的技術，將三個平面分別做隨意倍率的尺寸調整。最後透過三個調整過的平面作用之後，就可以得到尺寸調整過後的紅、藍平面，而後便可以得到隨意倍率的尺寸調整後的去馬賽克影像。具我們所知，這是第一個被提出的去馬賽克與隨意尺寸調整演算法。實驗結果顯示，所提出之馬賽克與隨意尺寸調整演算法比起透過直接單獨去馬賽克後又再做隨意尺寸調整的方式，有著較佳的品質與效果。

關鍵詞：適應性的異質估計，CIE色彩空間，彩色影像的對比增強，彩色影像測邊，色彩濾波陣列，彩色直方圖動差，彩色影像處理，彩色訊號雜訊比，彩色影像切割，色彩飽和及反飽和，去馬賽克演算法，數位相機，離散餘弦轉換，邊訊息保留，亮度估計，馬賽克影像，尺寸調整演算法，索貝爾算子。



# Abstract

Recently, color image processing is extensively becoming a very important research area since color images provide more fruitful information. Among the color image processing research issues, the edge-preserving for color contrast enhancement issue, the color demosaicing issue, and the mosaic image resizing issue are three popular research issues. Thus, this thesis presents an efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation, a demosaicing algorithm for color filter array using gradient edge detection masks and adaptive heterogeneity-projection, and a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images.

In the edge-preserving for color contrast enhancement issue, 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. Experimental results demonstrate the advantages of color contrast enhancement, edge-preservation effect, and segmentation result in our proposed algorithm.

In the color demosaicing issue, without demosaicing processing, a new approach is first proposed to extract more accurate gradient/edge information on mosaic images directly. Next, based on spectral-spatial correlation, a novel adaptive heterogeneity-

projection with proper mask size for each pixel is presented. Combining the extracted gradient/edge information and the adaptive heterogeneity-projection values, a new edge-sensing demosaicing algorithm is presented. Experimental results demonstrated that our proposed high-quality demosaicing algorithm has the best image quality performance when compared with several recently published algorithms.

In the mosaic image resizing issue, a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. First, the fully populated green color plane is constructed by using the edge-sensing approach and color difference idea. Instead of interpolating the R and B color planes directly, the green-red color difference plane and green-blue color difference plane are therefore interpolated in order to reduce the estimation error. Next, based on the discrete cosine transform technique, the above three constructed planes are resized to the arbitrary sized ones. Finally, the resized red and blue color planes are constructed by using the three resized planes, and then the arbitrary sized full color image is obtained. To the best of our knowledge, this is the first time that such a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. Based on twenty-four popular testing mosaic images, the proposed resizing algorithm has better image quality performance when compared with three native algorithms which are the combinations of three well-know demosaicing methods and one existing resizing method.

**Keywords:** Adaptive heterogeneity projection, CIE color model, Color contrast enhancement, Color edge detection, Color filter array (CFA), Color histogram mo-

ment, Color image processing, Color peak signal-to-noise ratio (CPSNR), Color image segmentation, Color saturation and desaturation, Demosaicing algorithm, Digital cameras, Discrete cosine transform (DCT), Edge-preservation effect, Luminance estimation, Mosaic images, Resizing algorithm, Sobel operator.



# Contents

<b>Abstract in Chinese</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background	1
1.2 Organization of the thesis	4
<b>2 Efficient Edge-Preserving Algorithm for Color Contrast Enhancement with Application to Color Image Segmentation</b>	<b>5</b>
2.1 Preliminaries	5
2.2 Two background techniques	7
2.2.1 CIE $Lu'v'$ color space	7
2.2.2 Color edge detector	8
2.3 Proposed edge-preserving algorithm for color contrast enhancement	10
2.3.1 Main concept	10
2.3.2 Speedup strategy	17
2.4 Color image segmentation application	20
2.4.1 Seed-based region growing phase	20
2.4.2 Merging phase	23



2.5	Experimental results . . . . .	27
2.6	Summary . . . . .	39
<b>3</b>	<b>Demosaicing Algorithm for Color Filter Array Using Gradient Edge Detection Masks and Adaptive Heterogeneity-Projection</b>	<b>40</b>
3.1	Preliminaries . . . . .	40
3.2	New approach to extract more accurate gradient information on mo- saic images . . . . .	43
3.2.1	Luminance estimation technique for mosaic images . . . . .	44
3.2.2	Proposed approach to extract more accurate gradient infor- mation on mosaic images . . . . .	46
3.3	Proposed new edge-sensing demosaicing algorithm . . . . .	50
3.3.1	Novel adaptive heterogeneity-projection for mosaic images . . . . .	50
3.3.2	Edge-sensing interpolation estimation for $G$ channel . . . . .	55
3.3.3	Edge-sensing interpolation estimation for $R$ and $B$ channels . . . . .	58
3.4	Experimental results . . . . .	60
3.5	Summary . . . . .	66
<b>4</b>	<b>Joint Demosaicing and Arbitrary-Ratio Resizing Algorithm for Bayer Color Filter Array Based on DCT Approach</b>	<b>69</b>
4.1	Preliminaries . . . . .	69
4.2	Proposed joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images . . . . .	71
4.2.1	Stage 1: Interpolating the mosaic $G$ plane . . . . .	72
4.2.2	Stage 2: Interpolating the mosaic $G$ - $R$ and $G$ - $B$ color differ- ence planes . . . . .	74

4.2.3	Stage 3: Resizing the fully populated G plane, G-R color difference plane, and G-B color difference plane . . . . .	77
4.3	Experimental results . . . . .	81
4.4	Summary . . . . .	85
<b>5</b>	<b>Conclusions</b>	<b>90</b>
<b>A</b>	<b>Derivation of the SL-based quad-mask</b>	<b>93</b>
<b>B</b>	<b>Determination of threshold <math>T_h</math> and parameters <math>\beta</math> and <math>\alpha</math></b>	<b>98</b>
	<b>Bibliography</b>	<b>100</b>
	<b>Curriculum Vitae</b>	<b>111</b>
	<b>Publication List</b>	<b>112</b>



# List of Figures

1.1	The Bayer CFA structure. . . . .	3
2.1	The CIE $u'v'$ chromatic diagram. . . . .	9
2.2	The first two steps in our proposed algorithm. (a) The saturation step. (b) The desaturation step. . . . .	11
2.3	The depiction of our proposed edge-preserving step. . . . .	14
2.4	One $3 \times 3$ subimage covered by the window mask. . . . .	14
2.5	Binary alternative search direction in edge-preservation step. . . . .	16
2.6	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. . . . .	17
2.7	The determined seeds as shown in blue colors of the enhanced Peppers image. . . . .	22
2.8	The over-segmented result after performing the seed-based region growing phase on the enhanced Peppers image in Fig. 6(b) . . . . .	24
2.9	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. . . . .	28

2.10	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).	29
2.11	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.	31
2.12	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.	32
2.13	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.	32
2.14	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.	33
2.15	The other two testing images. (a) The color Table Tennis image. (b) The color Akiyo image.	34

2.16	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). . . . .	37
2.17	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). . . . .	38
2.18	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. . . . .	39

3.1	The $3 \times 3$ single symmetric convolution mask. . . . .	44
3.2	Four possible cases of $3 \times 3$ mosaic subimages. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. . . . .	44
3.3	The $3 \times 3$ luminance estimation mask. . . . .	46
3.4	The $3 \times 3$ Sobel operator. (a) The horizontal mask. (b) The vertical mask. (c) The $\frac{\pi}{4}$ -diagonal mask. (d) The $\frac{-\pi}{4}$ -diagonal mask. . . . .	47
3.5	The four SL-based masks. (a) The horizontal SL-based mask. (b) The vertical SL-based mask. (c) The $\frac{\pi}{4}$ -diagonal SL-based mask. (d) The $\frac{-\pi}{4}$ -diagonal SL-based mask. . . . .	48
3.6	The two SSC maps of the mosaic Lighthouse image. (a) The hori- zontal SSC map. (b) The vertical SSC map. (In order to show the images more clear, the gray value 80 is used to represent the value 0.)	53
3.7	The depiction of our proposed approach to determine the proper hor- izontal mask size $N_H(i, j)$ . . . . .	53
3.8	The data dependence of our proposed interpolation estimation for $G$ channel. (a) Horizontal variation (vertical edge). (b) Vertical varia- tion (horizontal edge). (c) The other variations. . . . .	57
3.9	The patterns of the $R$ channel. . . . .	59
3.10	The twenty-four testing images from Kodak PhotoCD [85]. . . . .	61

3.11	The magnified subimages cut from the testing image No. 19. (a) Original full color image and the demosaiced images obtained from (b) Pei and Tam's algorithm. (c) Lu and Tan's algorithm. (d) Lukac and Plataniotis' algorithm. (e) Dubois' algorithm. (f) Lukac <i>et al.</i> 's algorithm. (g) Chung and Chan's algorithm. (h) Tsai and Song's algorithm. (i) our proposed algorithm. . . . .	67
3.12	The magnified subimages cut from the testing image No. 8. (a) Original full color image and the demosaiced images obtained from (b) Pei and Tam's algorithm. (c) Lu and Tan's algorithm. (d) Lukac and Plataniotis' algorithm. (e) Dubois' algorithm. (f) Lukac <i>et al.</i> 's algorithm. (g) Chung and Chan's algorithm. (h) Tsai and Song's algorithm. (i) our proposed algorithm. . . . .	68
4.1	The flowchart of the proposed arbitrary-ratio resizing algorithm. . . . .	71
4.2	The pattern of the mosaic G-R color difference plane. . . . .	74
4.3	Two patterns of the G-R color difference plane. (a) The pattern of the G-R color difference plane after performing Step 1. (b) The pattern shifting (a) one pixel down. . . . .	76
4.4	An example of the active unit in $\hat{\Phi}_{dm}^g$ . . . . .	78
4.5	The depiction of the $\frac{q}{p}$ -fold resizing procedure. . . . .	79
4.6	The magnified subimage cut from the original testing image No. 8. . . . .	84
4.7	For image No. 8, when the resizing ratio $\frac{q}{p}$ is 2, four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	85

4.8	For image No. 8, when the resizing ratio $\frac{q}{p}$ is $\frac{8}{5}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	86
4.9	For image No. 8, when the resizing ratio $\frac{q}{p}$ is $\frac{4}{3}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	87
4.10	For image No. 8, when the resizing ratio $\frac{q}{p}$ is $\frac{8}{7}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	87
4.11	The magnified subimage cut from the original testing image No. 23. . . . .	88
4.12	For image No. 23, when the resizing ratio $\frac{q}{p}$ is 2, four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	88
4.13	For image No. 23, when the resizing ratio $\frac{q}{p}$ is $\frac{8}{5}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	88
4.14	For image No. 23, when the resizing ratio $\frac{q}{p}$ is $\frac{4}{3}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	89
4.15	For image No. 23, when the resizing ratio $\frac{q}{p}$ is $\frac{8}{7}$ , four magnified subimages cut from the resized images obtained by (a) $A_1$ , (b) $A_2$ , (c) $A_3$ , and (d) the proposed algorithm. . . . .	89
B.1	Two demosaiced image quality surfaces in terms of $T_h$ and $\alpha$ for $\beta \geq 1$ .	
	(a) The CPSNR surface. (b) The S-CIELAB $\Delta E_{ab}^*$ surface. . . . .	99



B.2 Two demosaiced image quality curves in terms of  $\beta$  for  $T_h = 7$  and  $\alpha = 0.5$ . (a) The CPSNR curve. (b) The S-CIELAB  $\Delta E_{ab}^*$  curve. . . 99



# List of Tables

2.1	Edge-loss ratios for the concerned two color contrast enhancement algorithms. . . . .	34
2.2	Edge-add ratios for the concerned two color contrast enhancement algorithms. . . . .	34
3.1	CPSNR quality comparison for twenty-four testing images. . . . .	63
3.2	S-CIELAB $\Delta E_{ab}^*$ quality comparison for twenty-four testing images. . .	64
3.3	The average execution-time of eight demosaicing algorithms for twenty-four testing mosaic images. . . . .	65
4.1	Average CPSNR comparison for the four concerned algorithms. . . . .	83
4.2	Average S-CIELAB $\Delta E_{ab}^*$ comparison for the four concerned algorithms. .	83

# Chapter 1

## Introduction

### 1.1 Background

Recently, color image processing is extensively becoming a very important research area since color images provide more fruitful information. Color images are composed of three independent planes, e.g. the RGB plane. Although color images can be transformed into gray images and the gray image algorithms are followed, they loss color information so that the processing power is limited. Among the color image processing research issues, the edge-preserving for color contrast enhancement issue, the color demosaicing issue, and mosaic image resizing issue are three popular research issues.

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 [25, 64, 66]. Previously, many efficient algorithms for color contrast enhancement, such as the reducing color ordering-based algorithm [80], the curvelet transform-based algorithm [69], the saturation and desaturation-

based algorithm [43, 63], etc., have been successfully developed. However, 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. Thus, in edge-preserving for color contrast enhancement issue, the main motivations of this research are two-fold: (1) presenting a new algorithm to come to a compromise between the edge-preservation effect and the color contrast enhancement effect; (2) presenting a novel color image segmentation algorithm to justify the edge-preservation benefit in some application.

Recently, digital cameras have become more and more popular in consumer electronics market recently. In order to economize the hardware cost, instead of using three sensors, most digital cameras capture a color image with a signal sensor imaging pipeline [51] based on the well-known Bayer CFA [7], where each pixel in the captured image has only one measured color and this kind of images is called mosaic images. Fig. 1.1 depicts the Bayer CFA structure. Because  $G$  (green) color channel is the most important factor to determine the luminance of the color image, half of the pixels in Bayer CFA structure are assigned to  $G$  channel.  $R$  (red) and  $B$  (blue) color channels, which share the other half pixels in the Bayer CFA structure, are considered as the chrominance signals. In order to recover the full color image from the input mosaic image, the demosaicing process is used to estimate the other two color channels for each pixel [27, 34]. Bilinear interpolation (BI) [65] is the simplest demosaicing algorithm in which the unknown two color channels of each pixel are obtained by averaging its proper adjacent pixels. Then, in order to improve

	$j-3$	$j-2$	$j-1$	$j$	$j+1$	$j+2$	$j+3$
$i-3$	R	G	R	G	R	G	R
$i-2$	G	B	G	B	G	B	G
$i-1$	R	G	R	G	R	G	R
$i$	G	B	G	B	G	B	G
$i+1$	R	G	R	G	R	G	R
$i+2$	G	B	G	B	G	B	G
$i+3$	R	G	R	G	R	G	R

Fig. 1.1: The Bayer CFA structure.

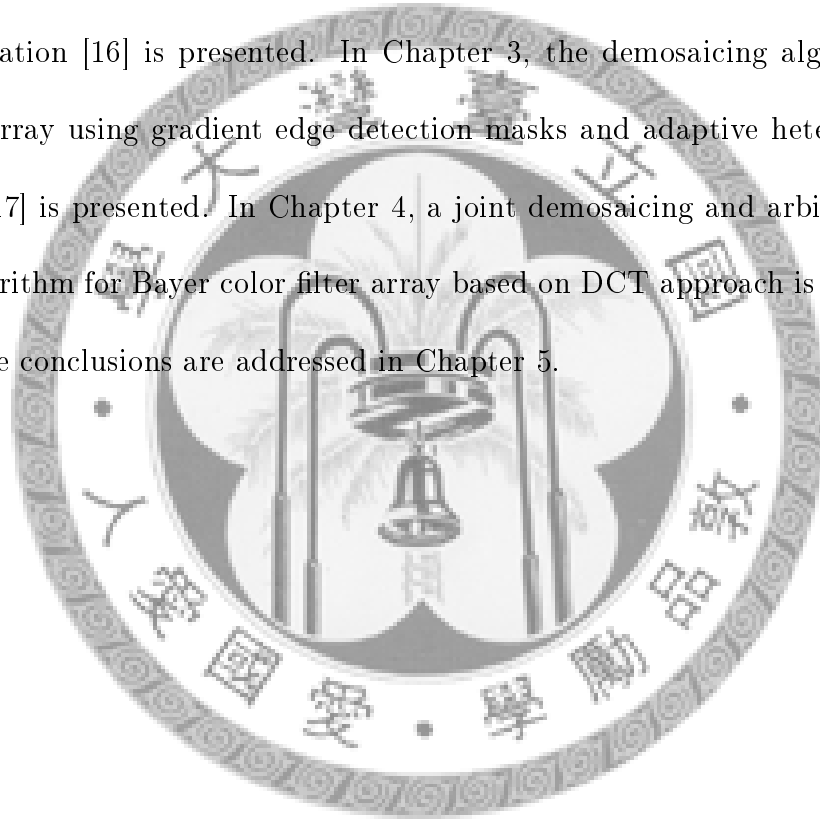
the demosaiced image quality performance, many modified demosaicing algorithm [2, 9, 13, 14, 19, 20, 23, 26, 28, 29, 30, 35, 36, 37, 38, 40, 41, 47, 48, 49, 50, 53, 55, 62, 72, 78, 79, 82, 83] have been developed. After examining most of all these previously published demosaicing algorithms and registered patents, we find that the quality of demosaiced images is heavily dependent on the extracted gradient/edge information from input mosaic images, but usually the extracted gradient/edge information on mosaic images is not so accurate. Since each pixel in the mosaic image only has one color channel, the previous color edge detectors [12, 21, 22, 73, 74, 75, 84] can not work well on mosaic images directly. Thus, in the color demosaicing issue, the motivations of this research are three-fold: (1) developing a new approach to extract more accurate gradient/edge information on mosaic images directly, (2) developing a new approach to determine the adaptive mask size for each pixel in the heterogeneity-projection, and (3) developing a new high-quality edge-sensing demosaicing algorithm based on the more accurate gradient/edge information and the adaptive heterogeneity-projection values.

Besides the demosaicing issue, how to resize mosaic images is another important research issue. The terms “resize” and “zoom” are used exchangeably. Several resizing algorithms for mosaic images have been developed [6, 14, 18, 44, 49, 50, 83].

Unfortunately, all of them only focus on the quad-zooming process. Thus, in the mosaic image resizing issue, the motivation of this research is to develop a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images.

## 1.2 Organization of the thesis

The rest of this thesis is organized as follows. In Chapter 2, the efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation [16] is presented. In Chapter 3, the demosaicing algorithm for color filter array using gradient edge detection masks and adaptive heterogeneity-projection [17] is presented. In Chapter 4, a joint demosaicing and arbitrary-ratio resizing algorithm for Bayer color filter array based on DCT approach is presented. Finally, some conclusions are addressed in Chapter 5.



# Chapter 2

## Efficient Edge-Preserving Algorithm for Color Contrast Enhancement with Application to Color Image Segmentation

### 2.1 Preliminaries

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 [25, 64, 66]. Previously, many efficient algorithms for color contrast enhancement have been successfully developed. Based on reducing color ordering approach [4], Zaharescu *et al.* [80] presented a color contrast enhancement algorithm. Based on the curvelet transform approach [8, 70], Starck *et al.* [69] 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 [43, 63]. In Lucchese *et al.*'s algorithm [43], they considered the chromaticity diagram [33, 42]. In Pei *et al.*'s algorithm [63], the considered color domain is the modified chromaticity diagram, i.e. the CIE  $Lu'v'$  color space [33]. In [63], 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 two-fold: (1) presenting a new algorithm to come to a compromise between the edge-preservation effect and the color contrast enhancement effect; (2) presenting a novel color image segmentation algorithm to justify the edge-preservation benefit in some application.

In this chapter, 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 [43, 63]. 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, we have better segmentation results on our obtained enhanced image when compared to it on the previous obtained enhanced image without edge-preservation.

## 2.2 Two background techniques

Before presenting our proposed color contrast enhancement algorithm, this section introduces two background techniques, namely the notion of CIE  $Lu'v'$  color space [33] and the previously published color edge detector by Trahanias and Venetianopoulos [75]. The two background techniques will be used in next section.

### 2.2.1 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

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.49000 & 0.31000 & 0.20000 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00000 & 0.01000 & 0.99000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.1)$$

In Eq. (2.1), 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:

$$u' = \frac{4X}{X + 15Y + 3Z}, \quad v' = \frac{9Y}{X + 15Y + 3Z} \quad (2.2)$$

By Eq. (2.1) and Eq. (2.2), the  $RGB$  color space can be transformed into the CIE  $Lu'v'$  color space.

Considering only the two components  $u'$  and  $v'$ , Fig. 2.1 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 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 [33]. 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  $W$ .

### 2.2.2 Color edge detector

In this subsection, the color edge detector by Trahanias and Venetsanopoulos [75] is described and it will be used in our proposed edge-preserving algorithm for color contrast enhancement although some other color edge detectors [67, 73, 81] can also be considered.

Suppose the window mask used in the color edge detector is of size  $w \times w$

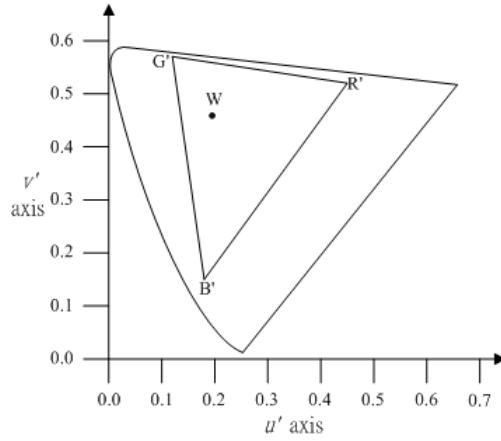


Fig. 2.1: The CIE  $u'v'$  chromatic diagram.

and the  $w^2$  color pixels covered by the window mask are denoted by the set  $P = \{P_1, P_2, \dots, P_{w^2}\}$ . 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 [33]. 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  $P_i = (u'_i, v'_i, Y_i)$ .

Based on the vector order statistic and the R-ordering concept [5], the color edge detector [75] consists of the following three steps.

**Step 1:** Sum up the color distances between each color pixel  $P_i$  and the other color pixels covered by the window mask. For color pixel  $P_i$ ,  $1 \leq i \leq w^2$ , the resulting distance is given by  $d_i = \sum_{k=1}^{w^2} \|P_i - P_k\|$ ,  $i = 1, 2, \dots, w^2$ , where  $\|\cdot\|$  represents an appropriate vector norm.

**Step 2:** Sort these  $w^2$  distances  $d_1, d_2, \dots$ , and  $d_{w^2}$ . Suppose these sorted  $w^2$  ascending distances are  $d_{i(1)}, d_{i(2)}, \dots$ , and  $d_{i(w^2)}$  for  $1 \leq i(1), i(2), \dots, i(w^2) \leq w^2$ . Among these  $w^2$  indices,  $P_{i(1)}$  is the color pixel with the minimal distance  $d_{i(1)}$ ;  $P_{i(w^2)}$  sometimes can be viewed as the outlier pixel in the  $w^2$

color pixels.

**Step 3:** Based on the robustness consideration, compute the minimum vector dispersion (*MVD*) which is given by

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

In [75],  $k$  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  $w \times w$  subimage is determined to be an edge pixel; otherwise, it is determined to be a non-edge pixel.

## 2.3 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.

### 2.3.1 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

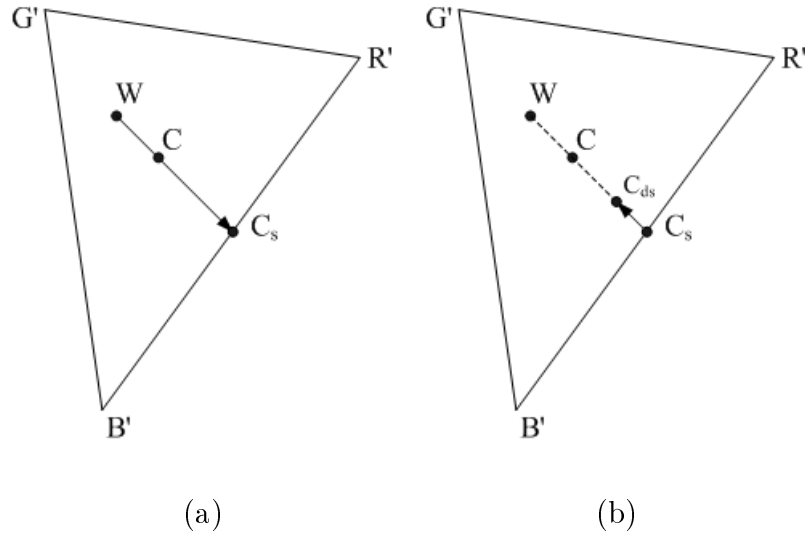


Fig. 2.2: The first two steps in our proposed algorithm. (a) The saturation step. (b) The desaturation step.

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 last section. 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.

### Saturation and desaturation steps

Here, the saturation step and the desaturation step are described. The two steps are adopted from the previous algorithms [43, 63].

Given an input color pixel  $C = (u'_C, v'_C, Y)$  as shown in Fig. 2.2(a), in saturation step, the color pixel  $C$  moves forward along the line  $\overrightarrow{WC}$ . When the color pixel  $C$  intersects the line segment  $\overline{B'R'}$ , 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. 2.2(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 [33]. The three components of the modified color point  $C_{ds} = (u'_{C_{ds}}, v'_{C_{ds}}, Y_{C_{ds}})$  are determined by

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

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

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.

### 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. 2.3 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 subimage covered by the window mask is illustrated in Fig. 2.4 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. 2.4 to determine whether the

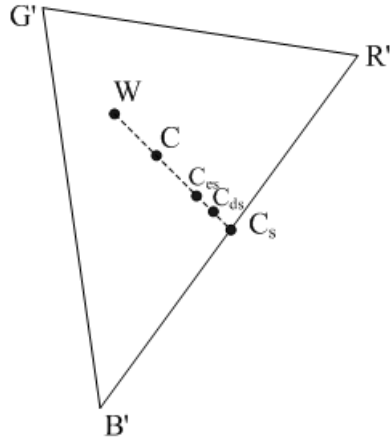


Fig. 2.3: The depiction of our proposed edge-preserving step.

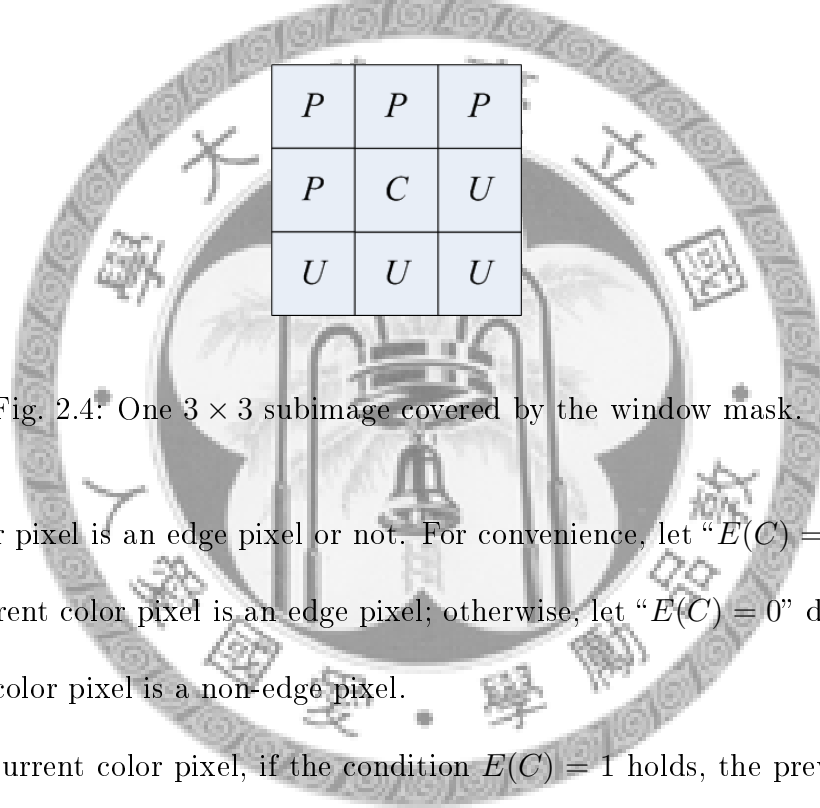


Fig. 2.4: One  $3 \times 3$  subimage covered by the window mask.

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. 2.5 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  $\|\overline{C_{t(1)} C_{ds}}\| = d$ , we move it to the next color pixel  $C_{t(2)}$  which is the near point of  $C_{ds}$  and examine whether the color pixel  $C_{t(2)}$  is an edge pixel or not, where  $\|\overline{C_{t(1)} C_{t(2)}}\| = \frac{d}{2}$ . If  $C_{t(2)}$  is an edge pixel, we move it to the pixel  $C_{t(3)}$ ; otherwise, we move it to the pixel  $C'_{t(3)}$ . 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

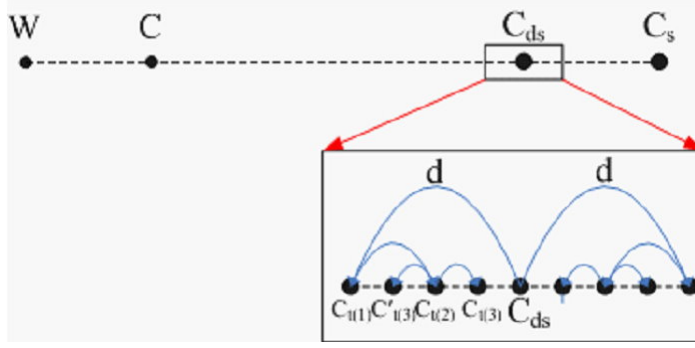


Fig. 2.5: Binary alternative search direction in edge-preservation step.

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:

$$\begin{aligned}
 X &= x(X + Y + Z) \\
 Y &= Y \\
 Z &= z(X + Y + Z)
 \end{aligned} \tag{2.3}$$

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. (2.1), the obtained values of  $X$ ,  $Y$ , and  $Z$  in Eq. (2.3) can be transferred into the values of  $r$ ,  $g$ , and  $b$ .

Based on the original Peppers image (see Fig. 2.6(a)), Fig. 2.6(b) illustrates the enhanced color image obtained by using the previous color contrast enhancement algorithm. Fig. 2.6(c) illustrates the enhanced color image obtained by using our proposed color contrast enhancement algorithm. It is observed that both enhanced

color images, Fig. 2.6(b) and Fig. 2.6(c), have the same color contrast enhancement effect and they look more colorful than that in Fig. 2.6(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 2.5. 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.



(a)

(b)

(c)

Fig. 2.6: 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.

### 2.3.2 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  $\overrightarrow{WC}$  and the gamut triangle is on the segment  $\overline{B'R'}$

(see Fig. 2.2(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:

$$\begin{aligned} u'_W + \tau(u'_C - u'_W) &= u'_{R'} + \lambda(u'_{B'} - u'_{R'}) \\ v'_W + \tau(v'_C - v'_W) &= v'_{R'} + \lambda(v'_{B'} - v'_{R'}). \end{aligned} \quad (2.4)$$

Eq. (2.4) can be rewritten by

$$\begin{aligned} \tau(u'_C - u'_W) - \lambda(u'_{B'} - u'_{R'}) &= u'_{R'} - u'_W \\ \tau(v'_C - v'_W) - \lambda(v'_{B'} - v'_{R'}) &= v'_{R'} - v'_W. \end{aligned} \quad (2.5)$$

After solving Eq. (2.5), 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  $\overline{B'G'}$ , 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  $\overline{R'B'}$ ,  $\overline{B'G'}$ , or  $\overline{G'R'}$ . 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% ( $= \frac{\text{old time} - \text{new time}}{\text{old time}} \times 100\%$ ) 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 2.5), 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 seconds. 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 seconds. Employing the two speedup strategies in our proposed edge-preserving algorithm is 0.347 seconds. In summary, the time required in any one of the above three concerned algorithms is much less than 1 second. 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 seconds. Therefore, the time required in color contrast enhancement could be ignored when compared to the ones in color image operations.

## 2.4 Color image segmentation application

In this section, our proposed novel color image segmentation algorithm, which uses the enhanced color image obtained in Section 2.3 to be the input image, is presented. In Section 2.5, 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 [11, 39, 76] 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.

### 2.4.1 Seed-based region growing phase

Our proposed seed-based region growing phase is modified from the previous ones in [1, 21, 57, 68]. 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.

#### 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. (2.3), the calculated value of  $MVD$  is assigned to  $v_{T_{ij}}$ , i.e.

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

Let

$$\sigma_{P_{ij}} = \frac{\sigma_{T_{ij}}}{\sigma_{max}}$$

$$v_{P_{ij}} = \frac{v_{T_{ij}}}{v_{max}}$$

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. 2.7 are the determined seeds in the enhanced color Peppers image as shown in Fig. 2.6(b).

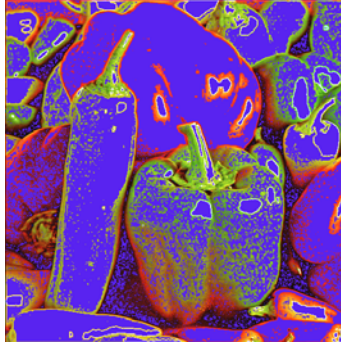


Fig. 2.7: The determined seeds as shown in blue colors of the enhanced Peppers image.

### 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

$$C(P_m, R_i) = \frac{\sqrt{(Y_{P_m} - \overline{Y_{R_i}})^2 + (u'_{P_m} - \overline{u'_{R_i}})^2 + (v'_{P_m} - \overline{v'_{R_i}})^2}}{\sqrt{Y_{P_m}^2 + u'^2_{P_m} + v'^2_{P_m}}}$$

where  $\overline{Y_{R_i}}$ ,  $\overline{u'_{R_i}}$ , and  $\overline{v'_{R_i}}$  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. 2.8 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.

### 2.4.2 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 difference 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



Fig. 2.8: The over-segmented result after performing the seed-based region growing phase on the enhanced Peppers image in Fig. 6(b) .

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  $R_k (= R_i \cup R_j)$  [25] 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

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

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

$$D_{GC}^C(R_i, R_j) = \|(GC_{R_k,x}, GC_{R_k,y}) - (C_{R_k,x}, C_{R_k,y})\|. \quad (2.6)$$

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 [56, 60, 71], our proposed color domain difference between  $R_i$  and  $R_j$  is given by

$$CD(R_i, R_j) = \|CC_{R_i} - CC_{R_j}\| \quad (2.7)$$

where  $CC_R = (CC_{R,u'}, CC_{R,v'}) = (\frac{M_{00}^R}{M_{00}^R}, \frac{M_{01}^R}{M_{00}^R})$ ,  $M_{pq}^R = \sum \sum_{(u',v') \in R} u'^p v'^q D_H^R(u', v')$ , 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. (2.7),  $CC_R = (CC_{R,u'}, CC_{R,v'})$  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 Eq. (2.6) and Eq. (2.7), the boundary edge information between  $R_i$  and  $R_j$  [58] 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

$$\gamma_E(R_i, R_j) = \frac{\sum_{(x,y) \in B_{(R_i, R_j)}} I_E(x,y)}{|B_{(R_i, R_j)}|} \quad (2.8)$$

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

By Eq. (2.6), Eq. (2.7), and Eq. (2.8), 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

$$D_T(R_i, R_j) = w_1 \times \frac{D_{GC}^C(R_i, R_j)}{D_{GC(max)}^C} + w_2 \times \frac{CD(R_i, R_j)}{CD(max)} + w_3 \times \gamma_E(R_i, R_j) \quad (2.9)$$

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. (2.9).

**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. (2.9), 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.

## 2.5 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 [43, 63]. 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. 2.9(a) illustrates the edge map of the original Peppers image shown in Fig. 2.6(a). The edge map of the previous obtained enhanced color Peppers image is illustrated in Fig. 2.9(b). The edge map of our obtained enhanced color Peppers image is illustrated in Fig. 2.9(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. 2.9(a)). However, the edge map in Fig. 2.9(b) is some different from the original edge map. Quantitative demonstrations (see Fig. 2.11, Fig. 2.12, Table 2.1, and Table 2.2) will be given later to explain why Fig. 2.9(c) is better than Fig. 2.9(b). Besides the color contrast enhancement effect existed in the previous color contrast

enhancement algorithm and our proposed one (see Fig. 2.6(b) and Fig. 2.6(c)), the edge-preservation effect of our proposed color contrast enhancement algorithm is better than that of the previous enhancement algorithm.

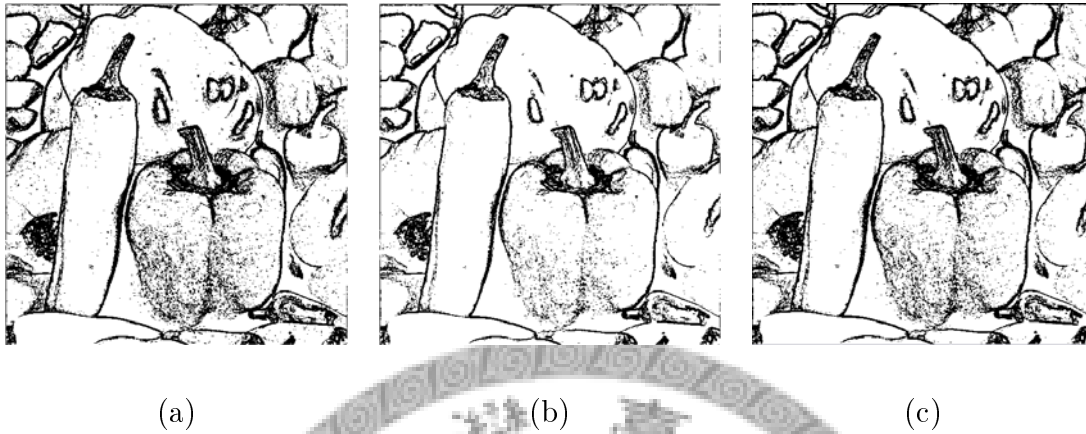


Fig. 2.9: The edge maps of color Peppers images. (a) The edge map of original image. (b) The edge map of the previous obtained enhanced image. (c) The edge map of our obtained enhanced image.

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. 2.10(a)) as the second testing image. Figs. 2.10(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 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.

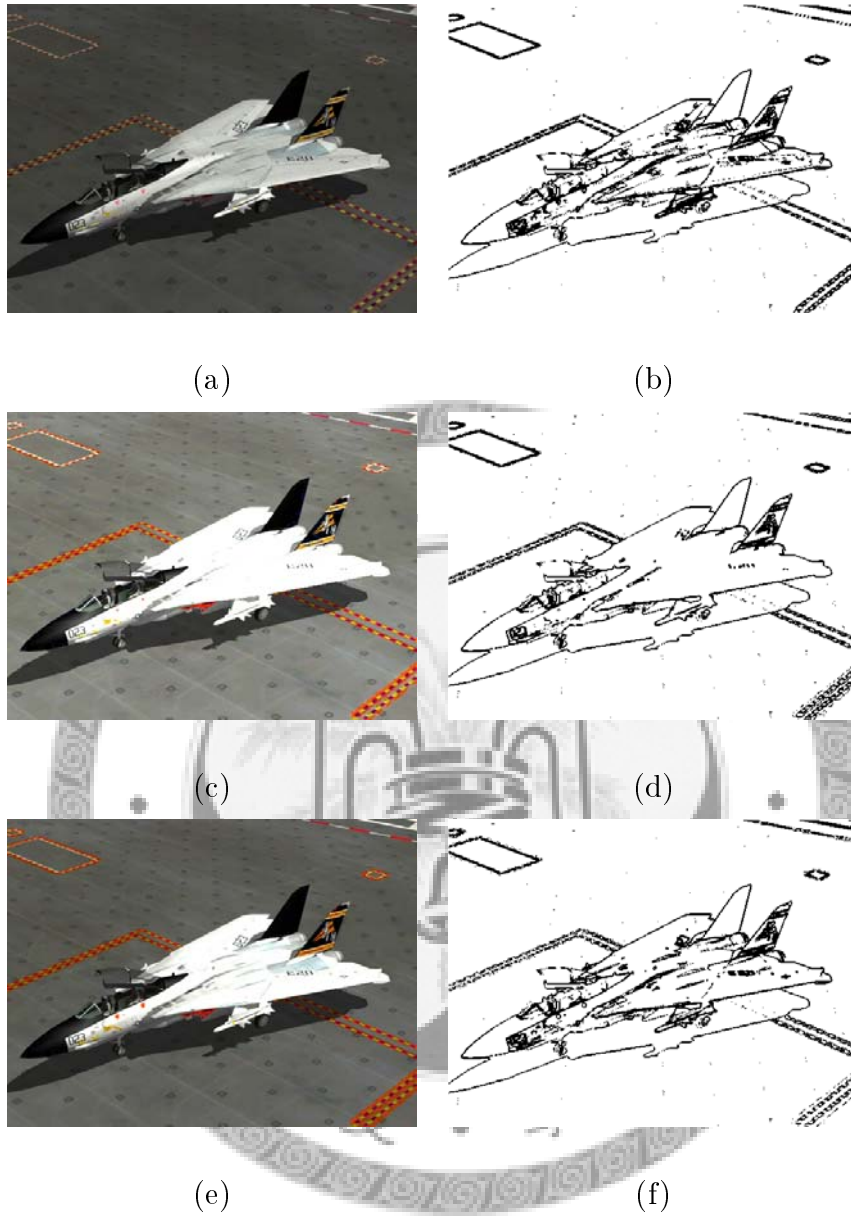


Fig. 2.10: The enhancement demonstration for color F14 images. (a) Original color F14 image. (b) The edge map of image (a). (c) The previous obtained enhanced image. (d) The edge map of image (c). (e) Our obtained enhanced image. (f) The edge map of image (e).

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. 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. 2.11(a) and Fig. 2.11(b) illustrate the set of edge-losing pixels  $N_{el}$  for the edge map of the previous obtained enhanced color Peppers image (see Fig. 2.9(b)) and the edge map of our obtained enhanced color Peppers image (see Fig. 2.9(c)), respectively. Fig. 2.12(a) and Fig. 2.12(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 Fig. 2.11 and Fig. 2.12, 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. 2.10(d)) and the edge map of our obtained enhanced color F14 image (see Fig. 2.10(f)) are illustrated in Fig. 2.13(a) and Fig. 2.13(b), respectively; the set of edge-adding pixels  $N_{ea}$  for



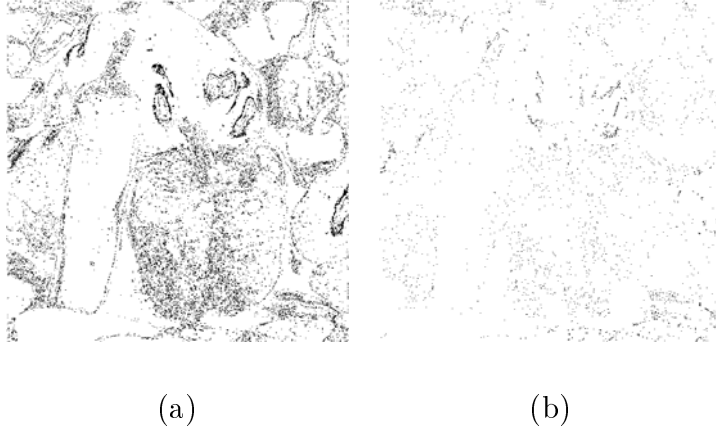


Fig. 2.11: The set of edge-losing pixels  $N_{el}$  for color Peppers image. (a)  $N_{el}$  for the previous obtained enhanced color Peppers image. (b)  $N_{el}$  for our obtained enhanced color Peppers image.

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. 2.14(a) and Fig. 2.14(b), respectively. From the above demonstrations, our proposed color contrast enhancement 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.

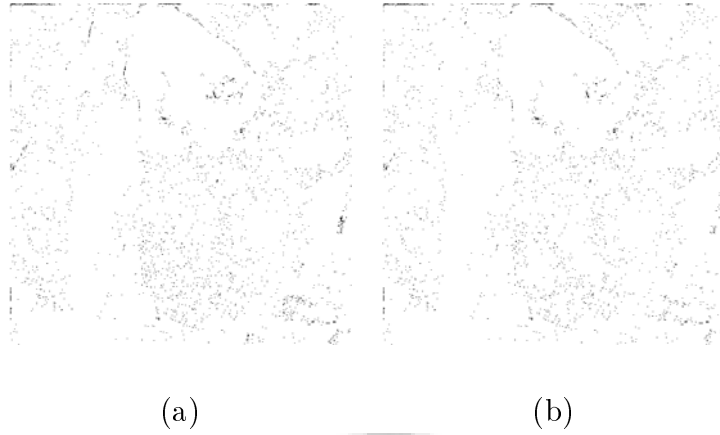


Fig. 2.12: The set of edge-adding pixels  $N_{ea}$  for color Peppers image. (a)  $N_{ea}$  for the previous obtained enhanced color Peppers image. (b)  $N_{ea}$  for our obtained enhanced color Peppers image.

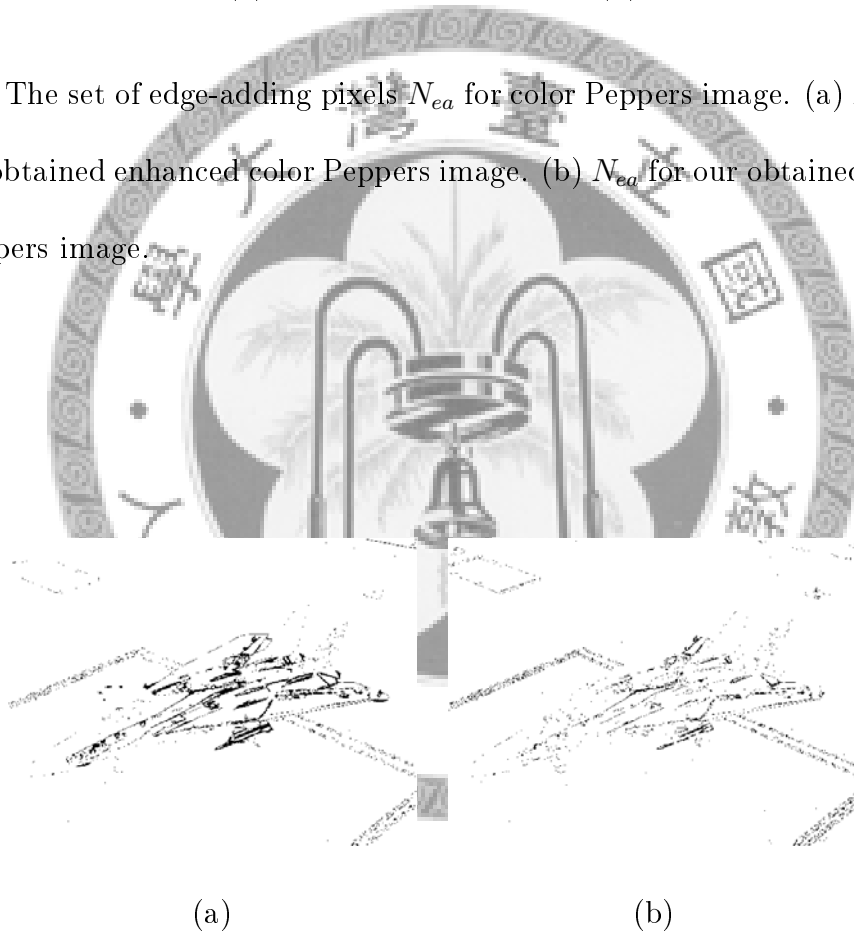


Fig. 2.13: The set of edge-losing pixels  $N_{el}$  for color F14 image. (a)  $N_{el}$  for the previous obtained enhanced color F14 image. (b)  $N_{el}$  for our obtained enhanced color F14 image.

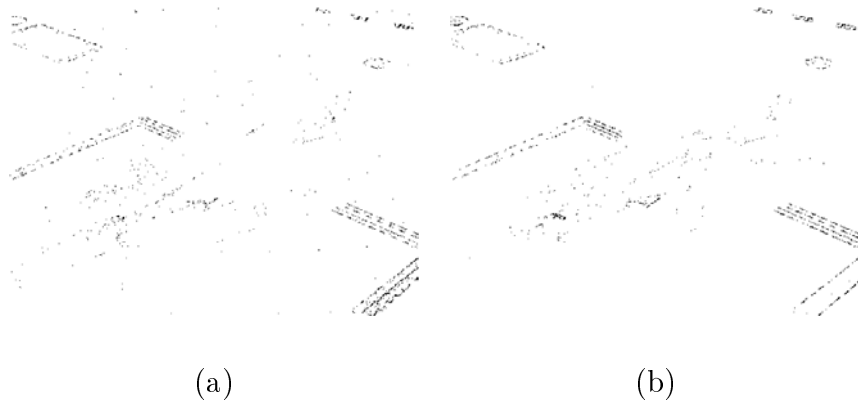
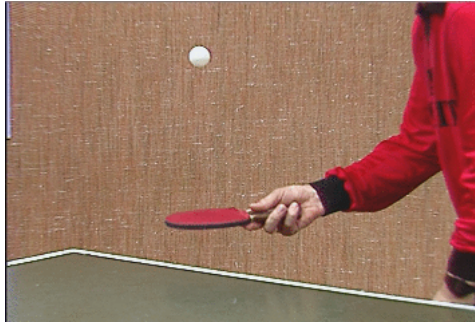


Fig. 2.14: The set of edge-adding pixels  $N_{ea}$  for color F14 image. (a)  $N_{ea}$  for the previous obtained enhanced image. (b)  $N_{ea}$  for our obtained enhanced image.

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. 2.15(a) and Fig. 2.15(b), respectively, are used. Based on the four testing images, Table 2.1 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 2.2 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. 2.16(a) illustrates the segmentation result when running our proposed



(a)



(b)

Fig. 2.15: The other two testing images. (a) The color Table Tennis image. (b) The color Akiyo image.

Table 2.1: 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%	

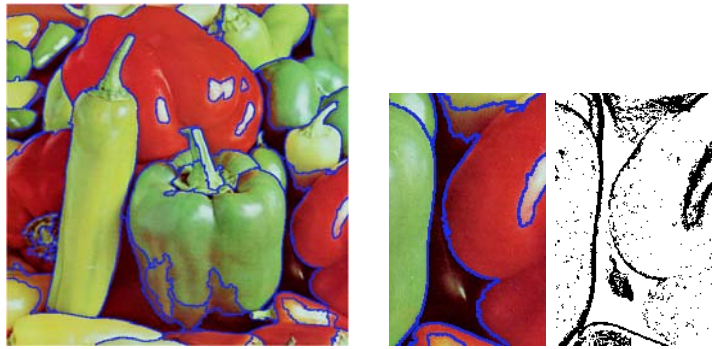
Table 2.2: Edge-add ratios for the concerned two color contrast enhancement algorithms.

	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%	

segmentation algorithm on the original image. Fig. 2.16(b) illustrates the magnified subimage cut from the left portion of one red pepper in Fig. 2.16(a) and the edge map of Fig. 2.16(b) is illustrated in Fig. 2.16(c). Fig. 2.16(d) and Fig. 2.16(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. 2.16(f) illustrates the magnified subimage cut from the left portion of one red pepper in Fig. 2.16(d) and the edge map of Fig. 2.16(f) is illustrated in Fig. 2.16(g). The magnified subimage cut from the same portion of the same red pepper in Fig. 2.16(e) is illustrated in Fig. 2.16(h) and Fig. 2.16(i) depicts the edge map of Fig. 2.16(h). After comparing the three edge maps of magnified subimages (see Fig. 2.16(c), Fig. 2.16(g), and Fig. 2.16(i)), it is observed that the edge map Fig. 2.16(i) is quite similar to Fig. 2.16(c), but Fig. 2.16(g) is some different from Fig. 2.16(c). In addition, the density and the distribution of edge pixels of Fig. 2.16(i) are rather close to those in Fig. 2.16(c), but the density of edge pixels of Fig. 2.16(g) is some sparse and is much less than that in Fig. 2.16(c). Table 2.1 and Table 2.2 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. 2.16(h) with Fig. 2.16(b) and comparing Fig. 2.16(f) with Fig. 2.16(b), it is observed that our segmentation result when running the proposed segmentation algorithm on our obtained enhanced color image (see

Fig. 2.16(h)) is better than the segmentation result (see Fig. 2.16(f)) when running the proposed segmentation algorithm on the previous obtained enhanced color image. Especially, the segmented boundary of the right pepper in Fig. 2.16(h) is more clear than that in Fig. 2.16(f) because our color contrast enhancement improves the segmentation quality.

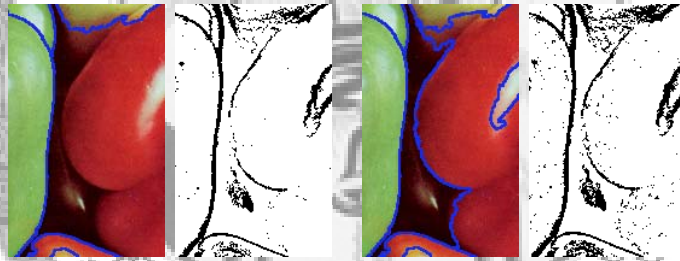
Further, the color F14 image is used to be the second testing image for segmentation comparison when running the proposed segmentation algorithm on the original F14 image and two kinds of enhanced F14 images. From the cockpit part of Fig. 2.17, it is observed that the number of preserved edge pixels and the distribution of edge pixels in Fig. 2.17(i) are quite similar to Fig. 2.17(c). However, the number of preserved edge pixels in Fig. 2.17(i) is greater than the one in Fig. 2.17(g). Table 2.1 and 2.2 provide the qualitative evidences. After examining Fig. 2.17(a), Fig. 2.17(d), and Fig. 2.17(e), it is observed that the segmentation result in Fig. 2.17(e) is quite similar to Fig. 2.17(a) and is better than that in Fig. 2.17(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. 2.18(a) and Fig. 2.18(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.



(a) (b) (c)



(d) (e)



(f) (g) (h) (i)

Fig. 2.16: 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).

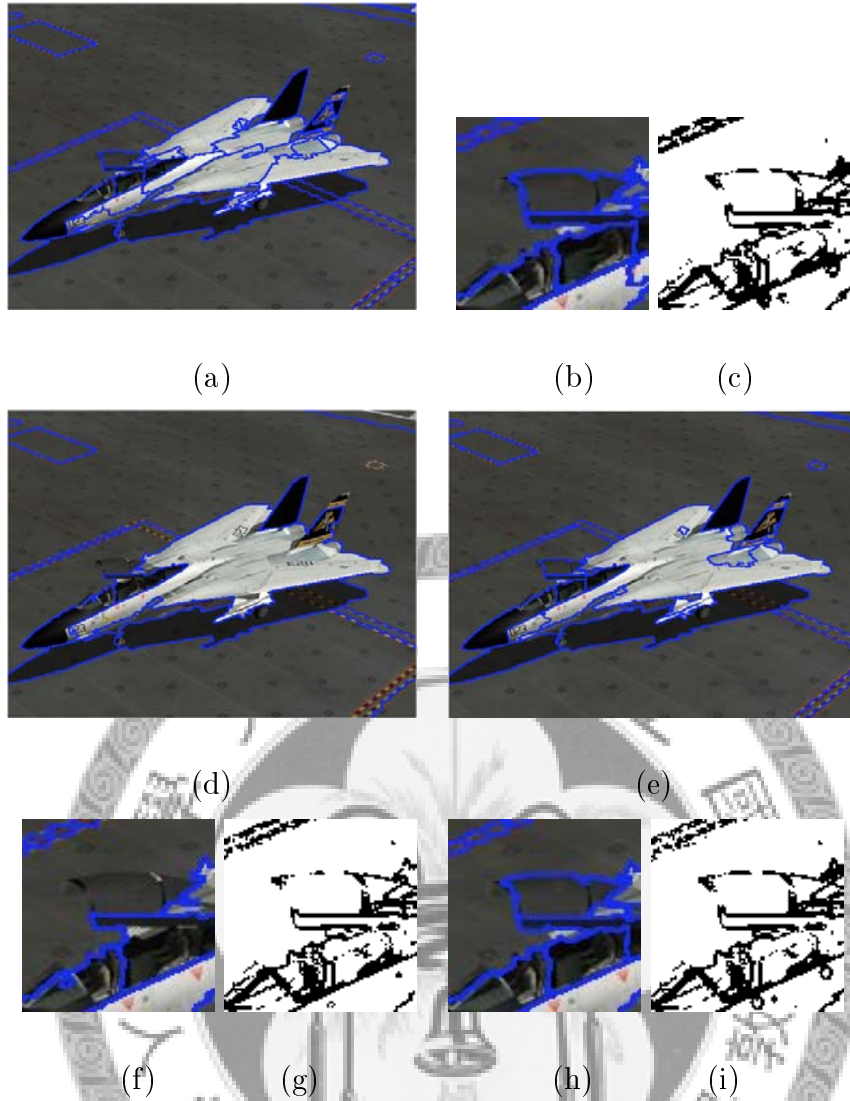


Fig. 2.17: The segmentation comparison for original F14 image and two kinds of enhanced F14 images. (a) The segmentation result when running the proposed segmentation algorithm on original image. (b) One magnified subimage extracted from (a). (c) The edge map of (b). (d) The segmentation result when 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).



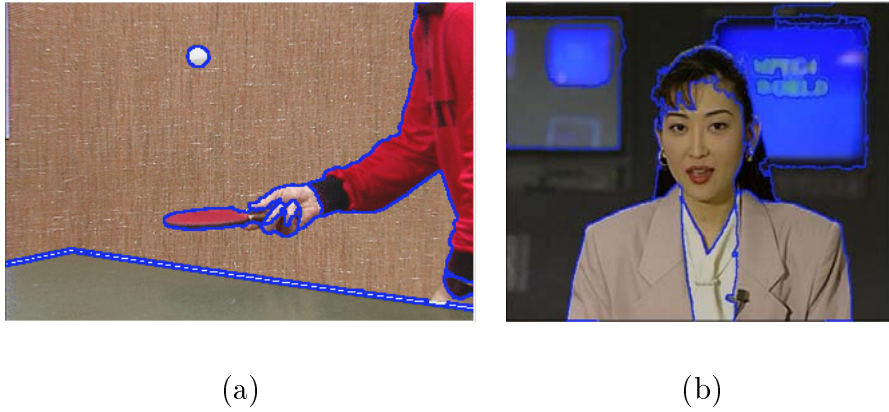


Fig. 2.18: The segmentation results when running our proposed color segmentation algorithm on our obtained enhanced color images. (a) The segmentation result for color Table Tennis image. (b) The segmentation result for color Akiyo image.

## 2.6 Summary

In this chapter, 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.

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.

# Chapter 3

## Demosaicing Algorithm for Color Filter Array Using Gradient Edge Detection Masks and Adaptive Heterogeneity-Projection

### 3.1 Preliminaries

In order to recover the full color image from the input mosaic image, the demosaicing process is used to estimate the other two color channels for each pixel [27, 34]. Bilinear interpolation (BI) [65] is the simplest demosaicing algorithm in which the unknown two color channels of each pixel are obtained by averaging its proper adjacent pixels. In [19], Cok presented a smooth hue transition-based demosaicing algorithm. In [23], Freeman presented a median filter-based demosaicing algorithm. Based on the single color gradient, Laroche and Prescott [37] presented an edge

preservation-based demosaicing algorithm. Based on the gradient evaluation and a specific threshold, Hibbard [29] presented a threshold-based demosaicing algorithm. Based on the adaptive color plane, Adams and Hamilton [2] presented an efficient demosaicing algorithm to modify the previous algorithms [37, 29]. In [35], Kimmel presented a color difference-based demosaicing algorithm by using the template matching technique. In [26], Gunturk *et al.* presented an efficient demosaicing algorithm by using alternating projections. Based on steerable wavelet decomposition, Hel-Or and Keren [28] presented an efficient demosaicing algorithm by using the directional smoothing technique. Based on color correlation concept, Pei and Tam [62] presented an efficient demosaicing algorithm. In [41], Lu and Tan presented an efficient demosaicing algorithm based on exploiting spatial and spectral correlations among adjacent pixels and they also presented a quality measure. Based on primary-consistent soft-decision framework, Wu and Zhang [79] presented a multiple estimation to estimate unknown color values according to different edge directions. In [48], Lukac and Plataniotis presented an efficient demosaicing algorithm by using normalized color-ratios. Based on optimal recovery interpolation of grayscale images, Muresan and Parks [55] presented an improved edge-directed demosaicing algorithm. In [30], Hirakawa and Parks presented an adaptive homogeneity-directed demosaicing algorithm. Based on projection-onto-convex-set approach, Li [40] presented the first iterative demosaicing algorithm. In [20], Dubois presented a novel demosaicing algorithm based on frequency-domain representation. In [36], Lai and Liaw presented an modified mean-removed vector quantization algorithm to improve the image quality performance of the previous algorithm by Pei and Tam [62]. Based on the linear minimum mean square-error estimation technique, Zhang and

Wu [82] presented a new color demosaicing algorithm. Based on concepts of edge-sensing and correlation-correction, Lukac *et al.* [53] presented a new efficient CFA interpolation framework. Su [72] presented an improved iterative demosaicing algorithm using weighted-edge and color-difference interpolations. Based on the spatial correlation and the edge-directional information of the neighboring pixels, Lee *et al.* [38] presented a weighted edge-sensing demosaicing algorithm. In [13], Chung and Chan presented an adaptive demosaicing algorithm by using the variances of color differences along horizontal and vertical edge directions. Currently, based on the  $N$ th-order directional finite derivative of spectral-spatial correlation, empirically  $N = 11$ , Tsai and Song [78] presented a lined-based demosaicing algorithm and their algorithm has the best image quality performance in average when compared with all other existing demosaicing algorithms. In addition, the edge-sensing concept used and the obtained edge-directional information in these developed demosaicing algorithms are very helpful to handle the digital zooming issue on CFA model [14, 47, 49, 50, 83].

After examining most of all these previously published demosaicing algorithms and registered patents, we find that the quality of demosaiced images is heavily dependent on the extracted gradient/edge information from input mosaic images, but usually the extracted gradient/edge information on mosaic images is not so accurate. Since each pixel in the mosaic image only has one color channel, the previous color edge detectors [12, 21, 22, 73, 74, 75, 84] can not work well on mosaic images directly. The motivations of this research are three-fold: (1) developing a new approach to extract more accurate gradient/edge information on mosaic images directly, (2) developing a new approach to determine the adaptive mask size for each pixel in the

heterogeneity-projection, and (3) developing a new high-quality edge-sensing demosaicing algorithm based on the more accurate gradient/edge information and the adaptive heterogeneity-projection values. The above three motivations lead to the three main contributions of this chapter.

In this chapter, without demosaicing processing, a new approach to extract more accurate gradient/edge information on mosaic images directly is first proposed. Next, based on spectral-spatial correlation [78], a novel adaptive heterogeneity-projection with proper mask size for each pixel is presented. Combining the extracted gradient/edge information and the adaptive heterogeneity-projection values, we present a new high-quality edge-sensing demosaicing algorithm. Based on twenty-four popular testing images, experimental results demonstrated that our proposed new demosaicing algorithm has the best quality performance when compared with current published seven demosaicing algorithms by Pei and Tam [62], Lu and Tan [41], Lukac and Plataniotis [48], Dubois [20], Lukac *et al.* [53], Chung and Chan [13], and Tsai and Song [78].

## 3.2 New approach to extract more accurate gradient information on mosaic images

This section presents a new approach to extract more accurate gradient information on mosaic images directly. In what follows, the luminance estimation technique [3] for mosaic images is first introduced. Then, combining the luminance estimation technique and Sobel operator [24], our proposed new approach to extract more gradient information on mosaic images is presented. The extracted gradient infor-

c	b	c
b	a	b
c	b	c

Fig. 3.1: The  $3 \times 3$  single symmetric convolution mask.

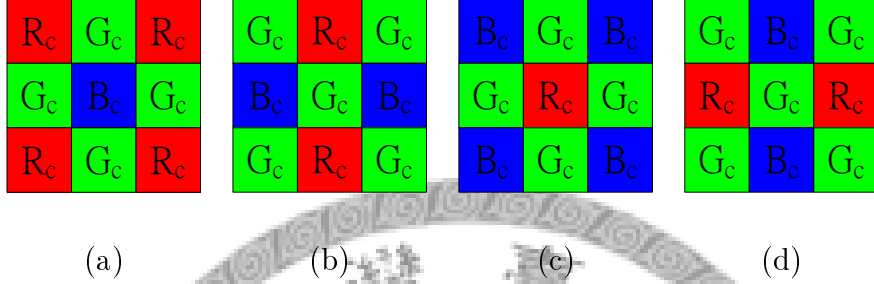


Fig. 3.2: Four possible cases of  $3 \times 3$  mosaic subimages. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4.

mation will be used in our proposed edge-sensing demosaicing algorithm described in Section 3.3 and it could lead to a high-quality advantage.

### 3.2.1 Luminance estimation technique for mosaic images

In this subsection, the luminance estimation technique for mosaic images is introduced. In the discussions hereafter, the luminance is defined as  $L = \frac{1}{4}(R + 2G + B)$  and the luminance of the pixel located at position  $(i, j)$  is denoted by  $L(i, j)$ ; the  $R$ ,  $G$ , and  $B$  color pixels located at position  $(i, j)$  of the mosaic image are denoted by  $I_{mo}^r(i, j)$ ,  $I_{mo}^g(i, j)$ , and  $I_{mo}^b(i, j)$ , respectively; for a demosaiced full color image, suppose the  $R$ ,  $G$ , and  $B$  color values of the pixel at position  $(i, j)$  are denoted by  $I_{dm}^r(i, j)$ ,  $I_{dm}^g(i, j)$ , and  $I_{dm}^b(i, j)$ , respectively, where  $i$  denotes the vertical axis and  $j$  denotes the horizontal axis (see Fig. 1.1).

In the luminance estimation technique, a  $3 \times 3$  single symmetric convolution mask

as shown in Fig. 3.1 is used to estimate the luminance of the pixel at  $(i, j)$  in the mosaic image. Within a small smooth region of the mosaic image, the color values of  $R$ ,  $G$ , and  $B$  components approach three different constants, i.e.  $I_{mo}^r(i, j) \cong R_c$ ,  $I_{mo}^g(i, j) \cong G_c$ , and  $I_{mo}^b(i, j) \cong B_c$ . The four possible cases of the  $3 \times 3$  mosaic subimage are illustrated in Fig. 3.2(a)–(d), respectively. Because of the symmetry of the  $3 \times 3$  mask in Fig. 3.1, only Case 1 and Case 2 need to be considered. First, each  $R$  channel of Case 1 and Case 2 is considered. After running the mask of Fig. 3.1 on the two  $3 \times 3$  mosaic subimages of Fig. 3.2(a) and Fig. 3.2(b), we have  $4cR_c = 2bR_c$ . By the same argument, for the  $B$  channel, we have  $aB_c = 2bB_c$ . For normalizing the sum of the nine coefficients in the mask, we let  $a + 4b + 4c = 1$ . From  $4c = 2b$ ,  $a = 2b$ , and  $a + 4b + 4c = 1$ , we have the following three equations

$$\begin{cases} 4c = 2b \\ a = 2b \\ a + 4b + 4c = 1 \end{cases} \quad (3.1)$$

After solving Eq. (3.1), it yields  $a = \frac{1}{4}$ ,  $b = \frac{1}{8}$ , and  $c = \frac{1}{16}$ . Thus, the luminance estimation mask can be determined and it is shown in Fig. 3.3. After running the the luminance estimation mask of Fig. 3.3 on the  $3 \times 3$  mosaic subimage centered at position  $(i, j)$ , the luminance  $L(i, j)$  can be obtained by

$$L(i, j) = \frac{1}{16} \left\{ \begin{array}{l} \left[ \begin{array}{l} I_{mo}^C(i-1, j-1) + I_{mo}^C(i-1, j+1) \\ + I_{mo}^C(i+1, j-1) + I_{mo}^C(i+1, j+1) \end{array} \right] \\ + 2 \left[ \begin{array}{l} I_{mo}^C(i-1, j) + I_{mo}^C(i+1, j) \\ + I_{mo}^C(i, j-1) + I_{mo}^C(i, j+1) \end{array} \right] \\ + 4I_{mo}^C(i, j) \end{array} \right\} \quad (3.2)$$

1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16

Fig. 3.3: The  $3 \times 3$  luminance estimation mask.

where

$$I_{mo}^C(m, n) = \begin{cases} I_{mo}^r(m, n) & \text{if } m \in \text{odd} \text{ and } n \in \text{even} \\ I_{mo}^g(m, n) & \text{if } (m + n) \in \text{even} \\ I_{mo}^b(m, n) & \text{if } m \in \text{even} \text{ and } n \in \text{odd} \end{cases}$$

In next section, the proposed approach to extract more accurate gradient information on mosaic images by combining the Sobel operator and the luminance estimation technique will be presented.

### 3.2.2 Proposed approach to extract more accurate gradient information on mosaic images

In this subsection, combining the Sobel operator [24] and the luminance estimation technique mentioned above, our proposed new approach to extract more accurate gradient information on mosaic images is presented.

Before presenting the proposed new approach, for completeness, how to use the Sobel operator to extract the gradient information on the luminance map is first introduced. Fig. 3.4 illustrates the four masks used in the  $3 \times 3$  Sobel operator for the luminance map.

Given a luminance map, it is known that the luminance of the pixel located at position  $(i, j)$  is denoted by  $L(i, j)$ . After running Sobel horizontal, vertical,



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

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

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

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

(a)
(b)
(c)
(d)

Fig. 3.4: The  $3 \times 3$  Sobel operator. (a) The horizontal mask. (b) The vertical mask. (c) The  $\frac{\pi}{4}$ -diagonal mask. (d) The  $\frac{-\pi}{4}$ -diagonal mask.

$\frac{\pi}{4}$ -diagonal, and  $\frac{-\pi}{4}$ -diagonal masks as shown in Fig. 3.4 on the  $3 \times 3$  luminance submap centered at position  $(i, j)$ , the horizontal response  $\Delta I_{dm}^H(i, j)$ , the vertical response  $\Delta I_{dm}^V(i, j)$ , the  $\frac{\pi}{4}$ -diagonal response  $\Delta I_{dm}^{\frac{\pi}{4}}(i, j)$ , and the  $\frac{-\pi}{4}$ -diagonal response  $\Delta I_{dm}^{\frac{-\pi}{4}}(i, j)$  can be calculated by

$$\begin{aligned}
 \Delta I_{dm}^H(i, j) &= \left\{ \begin{array}{l} [L(i-1, j+1) + 2L(i, j+1) + L(i+1, j+1)] \\ -[L(i-1, j-1) + 2L(i, j-1) + L(i+1, j-1)] \end{array} \right\} \\
 \Delta I_{dm}^V(i, j) &= \left\{ \begin{array}{l} [L(i+1, j-1) + 2L(i+1, j) + L(i+1, j+1)] \\ -[L(i-1, j-1) + 2L(i-1, j) + L(i-1, j+1)] \end{array} \right\} \\
 \Delta I_{dm}^{\frac{\pi}{4}}(i, j) &= \left\{ \begin{array}{l} [L(i-1, j) + 2L(i-1, j+1) + L(i, j+1)] \\ -[L(i+1, j) + 2L(i+1, j-1) + L(i, j-1)] \end{array} \right\} \\
 \Delta I_{dm}^{\frac{-\pi}{4}}(i, j) &= \left\{ \begin{array}{l} [L(i-1, j) + 2L(i-1, j-1) + L(i, j-1)] \\ -[L(i+1, j) + 2L(i+1, j+1) + L(i, j+1)] \end{array} \right\}
 \end{aligned} \tag{3.3}$$

In order to make Sobel operator workable on mosaic images to extract more accurate gradient information, the luminance estimation technique could be embedded into the Sobel operator. Combining Eq. (3.2) and Eq. (3.3), our proposed new approach to extract more accurate gradient information on mosaic images directly can

be followed (detailed derivations are shown in Appendix A). According to Appendix A, the derived Sobel- and luminance estimation-based (SL-based) quad-mask can be obtained. For saving computational effort, the coefficients in the derived quad-mask can be normalized to integers, and the four normalized SL-based quad-masks are shown in Fig. 3.5.

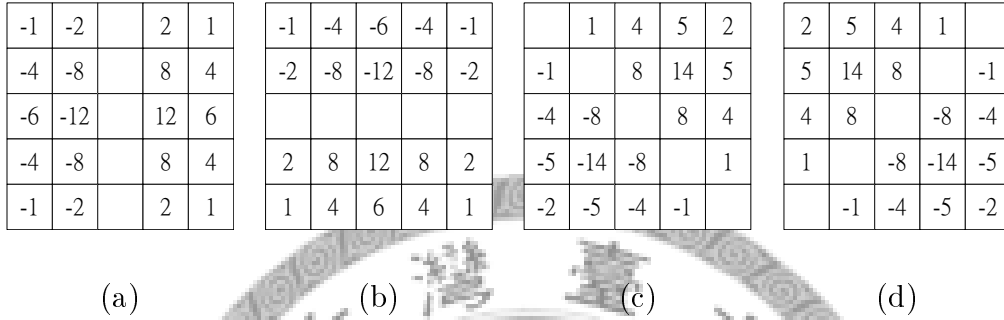


Fig. 3.5: The four SL-based masks. (a) The horizontal SL-based mask. (b) The vertical SL-based mask. (c) The  $\frac{\pi}{4}$ -diagonal SL-based mask. (d) The  $\frac{-\pi}{4}$ -diagonal SL-based mask.

After running the above four SL-based masks on the  $5 \times 5$  mosaic subimage centered at position  $(i, j)$ , the horizontal response  $\Delta I_{dm}^H(i, j)$ , the vertical response  $\Delta I_{dm}^V(i, j)$ , the  $\frac{\pi}{4}$ -diagonal response  $\Delta I_{dm}^{\frac{\pi}{4}}(i, j)$ , and the  $\frac{-\pi}{4}$ -diagonal response  $\Delta I_{dm}^{\frac{-\pi}{4}}(i, j)$  can be obtained. In addition, experimental results show that the proposed approach to extract gradient information on mosaic images directly has better performance and the average execution-time improvement ratio is 17.8% ( $= \frac{0.360s - 0.296s}{0.360s} \times 100\%$ ) when compared with the indirect approach: first apply the bilinear demosaicing process to the input mosaic image; next convert the demosaiced full color image to the luminance map, and then the Sobel edge detector is run on the obtained luminance map [54].

Note that besides our proposed approach and the indirect approach mentioned

above, we now briefly introduce the third approach in [15]: first, plug the bilinear interpolation technique [65] into the Sobel operator to derive two quad-mask for the  $G$  channel, four quad-mask for  $R$  channel, and the same four mask-pairs for  $B$  channel. Next, run the above proper mask-pairs on the  $5 \times 5$  mosaic subimage to compute the gradient response of the corresponding color channel. Further, let the four gradient responses of  $R$ ,  $G$ , and  $B$  channels are denoted by  $\Delta I_{dm}^{d,r}$ ,  $\Delta I_{dm}^{d,g}$ , and  $\Delta I_{dm}^{d,b}$  for  $d \in \{V, H, \frac{\pi}{4}, \frac{-\pi}{4}\}$ , respectively. Finally, combine  $R$ ,  $G$ , and  $B$  gradient responses to obtain the integrated gradient responses  $\Delta I_{dm}^d$  by using the equation:  $\Delta I_{dm}^d = 0.299\Delta I_{dm}^{d,r} + 0.587\Delta I_{dm}^{d,g} + 0.114\Delta I_{dm}^{d,b}$  for  $d \in \{V, H, \frac{\pi}{4}, \frac{-\pi}{4}\}$ . For convenience, the demosaicing algorithm based on the third approach is called the Sobel- and interpolation-based (SI-based) demosaicing algorithm. Experimental results show that the average image quality performance, CPSNR and  $\Delta E_{ab}^*$ , of the SI-based demosaicing algorithm could be improved by 0.027% ( $= \frac{40.200dB - 40.189dB}{40.200dB} \times 100\%$ ) and 0.135% ( $= \frac{1.2606 - 1.2589}{1.2606} \times 100\%$ ), respectively, when compared with the proposed demosaicing algorithm in this chapter. However, the average time performance of the SI-based demosaicing algorithm would be degraded 8.3% ( $= \frac{1.573s - 1.442s}{1.573s} \times 100\%$ ). Since the average image quality improvement of the SI-based demosaicing algorithm over our proposed algorithm is negligible but the degraded time performance is moderate, thus, our proposed SL-based operator described in this section is the better choice for handling the edge-sensing process which will be used in our proposed demosaicing algorithm described in next section.

### 3.3 Proposed new edge-sensing demosaicing algorithm

Based on the accurate gradient information obtained in last section, this section presents our proposed new high-quality edge-sensing demosaicing algorithm. Our proposed novel adaptive heterogeneity-projection with proper mask size for each pixel is first presented in Subsection 3.3.1 to extract more accurate horizontal and vertical heterogeneity-projection value. Based on the two obtained heterogeneity-projection values and those obtained gradient information mentioned in Section 3.2, the proposed new edge-sensing interpolation estimation is presented. The proposed interpolation estimation consists of two steps, the first step for  $G$  channel and the second step for  $R$  and  $B$  channels.

#### 3.3.1 Novel adaptive heterogeneity-projection for mosaic images

In this subsection, a novel adaptive heterogeneity-projection with proper mask size for each pixel is presented. Given an original mosaic image  $I_{mo}$ , its horizontal heterogeneity-projection map  $HP_{H-map}$  and vertical heterogeneity-projection map  $HP_{V-map}$  can be obtained by running the following two 1-D Laplacian operations [78]:

$$\begin{aligned}
 HP_{H-map} &= |I_{mo} * (\mathbf{F}^{N \times (N-3)} \mathbf{T}^{(N-3) \times 1})^t| \\
 HP_{V-map} &= |I_{mo} * (\mathbf{F}^{N \times (N-3)} \mathbf{T}^{(N-3) \times 1})|
 \end{aligned}
 \tag{3.4}$$

where  $N$  ( $=11$  empirically [78]) denotes the vector length (or the 1-D mask size);  $\mathbf{F}^{N \times (N-3)} = [1 \ -1 \ -1 \ 1]^t * \mathbf{I}^{(N-3) \times (N-3)}$  denotes a  $N \times (N-3)$  coefficient matrix;

$\mathbf{T}^{(N-3) \times 1} = \prod_{x=1}^{N-4} [1 \ -1]^t * \mathbf{I}^{(N-3-x) \times (N-3-x)}$  denotes a  $(N-3) \times 1$  coefficient vector;  $\mathbf{I}^{M \times M}$  denotes an  $M \times M$  identity matrix; the symbol “ $*$ ” denotes the 2-D convolution operator;  $|\cdot|$  denotes the absolute value operator and the operator “ $t$ ” denotes the transpose operator. In what follows, our approach can determine the suitable value of  $N$  adaptively.

Let us examine some cases of the current pixel on the mosaic image. If the surrounding region of the current pixel is homogenous, the two responses by Eq. (3.4) are almost the same whether a large mask size or small mask size is adopted. If there is one tiny horizontal edge passing through the current pixel, a small mask size for  $HP_{V-map}$  is enough rather than a large mask size. It still holds for the tiny vertical edge corresponding to  $HP_{H-map}$ . In [78], the adopted mask size is fixed and is set to  $N = 11$ . According to the above discussion, we now present our proposed adaptive heterogeneity-projection with proper mask size for each pixel such that the used mask size is as small as possible and the computed responses are more accurate than those computed by the mask with size 11. Experimental results reveal that our proposed adaptive heterogeneity-projection approach under proper horizontal and vertical mask sizes has the computation-saving and more accurate advantages.

We utilize the horizontal spectral-spatial correlation (SSC) [78] between the current pixel at location  $(i, j)$  and its neighboring pixel at location  $(i, j+1)$  to determine the proper horizontal mask size  $N_H(i, j)$ , and the proper vertical mask size  $N_V(i, j)$  is determined by the vertical SSC between the current pixel and its neighboring pixel at location  $(i+1, j)$ . For a horizontal SSC map, the horizontal SSC value at

location  $(i, j)$  can be obtained by using the following rules:

$$S_H(i, j) = \begin{cases} |I_{mo}^g(i, j) - I_{mo}^r(i, j + 1)| & \text{if the current pixel is G and } i \in \text{odd.} \\ |I_{mo}^g(i, j) - I_{mo}^b(i, j + 1)| & \text{if the current pixel is G and } i \in \text{even.} \\ |I_{mo}^r(i, j) - I_{mo}^g(i, j + 1)| & \text{if the current pixel is R.} \\ |I_{mo}^b(i, j) - I_{mo}^g(i, j + 1)| & \text{if the current pixel is B.} \end{cases}$$

Similarly, the vertical SSC value at location  $(i, j)$  can be obtained by using the following rules:

$$S_V(i, j) = \begin{cases} |I_{mo}^g(i, j) - I_{mo}^r(i + 1, j)| & \text{if the current pixel is green and } j \in \text{even.} \\ |I_{mo}^g(i, j) - I_{mo}^b(i + 1, j)| & \text{if the current pixel is green and } j \in \text{odd.} \\ |I_{mo}^r(i, j) - I_{mo}^g(i + 1, j)| & \text{if the current pixel is red.} \\ |I_{mo}^b(i, j) - I_{mo}^g(i + 1, j)| & \text{if the current pixel is blue.} \end{cases}$$

The horizontal SSC map and the vertical SSC map of the mosaic Lighthouse image are illustrated in Fig. 3.6(a) and Fig. 3.6(b), respectively, and it is observed that the SSC values are locally constant in homogeneous regions.

Because the determination of the proper horizontal mask size  $N_H(i, j)$  is the same as that for the vertical mask size  $N_V(i, j)$ , we thus only focus on the determination of horizontal mask size. Fig. 3.7 depicts our proposed approach to determine the proper horizontal mask size  $N_H(i, j)$ . Centered at location  $(i, j)$ , Fig. 3.7 illustrates a  $1 \times 11$  row data extracted from the horizontal SSC map. The procedure to determine the proper horizontal mask size  $N_H(i, j)$  consists of the following three steps.

**Step 1:** Initially, we compute the two SSC values  $S_H(i, j - 2)$  and  $S_H(i, j + 2)$ , and

temporarily set  $N_H(i, j) = 5$ ,  $x_{left} = j - 2$ , and  $x_{right} = j + 2$ .

**Step 2:** If the condition  $N_H(i, j) = 11$  holds, output  $N_H(i, j)$  and stop. Here, the maximal horizontal mask size is bounded by eleven. Otherwise, go to Step



(a)

(b)

Fig. 3.6: The two SSC maps of the mosaic Lighthouse image. (a) The horizontal SSC map. (b) The vertical SSC map. (In order to show the images more clear, the gray value 80 is used to represent the value 0.)

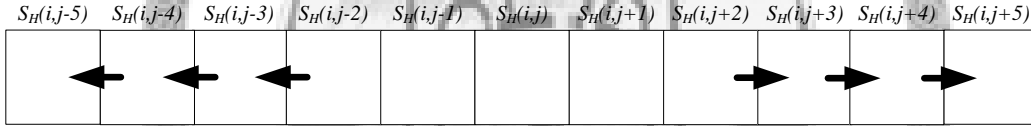


Fig. 3.7: The depiction of our proposed approach to determine the proper horizontal mask size  $N_H(i, j)$ .

3.

**Step 3:** Examine whether the neighboring horizontal SSC values are locally chosen by using the following testing condition:

$$\text{Max}(DS_H^{\text{left}}, DS_H^{\text{right}}) < T_h$$

where  $DS_H^{\text{left}} = |S_H(i, x_{\text{left}}) - S_H(i, x_{\text{left}} - 1)| + |S_H(i, x_{\text{left}}) - S_H(i, x_{\text{left}} + 1)|$ ,  $DS_H^{\text{right}} = |S_H(i, x_{\text{right}}) - S_H(i, x_{\text{right}} - 1)| + |S_H(i, x_{\text{right}}) - S_H(i, x_{\text{right}} + 1)|$ , and determination of the threshold  $T_h$  is discussed in Appendix B.

If the above testing condition holds, output the value of  $N_H(i, j)$  as the proper horizontal mask size and stop. Otherwise, perform the operation  $N_H(i, j) = N_H(i, j) + 2$ ,  $x_{left} = x_{left} - 1$ , and  $x_{right} = x_{right} + 1$ , and then go to Step 2.

The above three-step procedure can be easily applied to determine the proper horizontal mask size. In order to normalize the masks for different sizes, the normal factor  $\frac{1}{Q_{(N)}}$  is used to normalize the coefficients of the mask. In other words, instead of using  $\mathbf{F}^{N \times (N-3)} \mathbf{T}^{(N-3) \times 1}$ , we use  $\frac{1}{Q_{(N)}} (\mathbf{F}^{N \times (N-3)} \mathbf{T}^{(N-3) \times 1})$  to obtain the normalized heterogeneity-projection map. The value of  $Q_{(N)}$  is defined as the sum of the positive coefficients in the mask. For example, if  $N = 5$ , the mask  $\mathbf{F}^{5 \times (5-3)} \mathbf{T}^{(5-3) \times 1} = [1 \ -2 \ 0 \ 2 \ -1]^t$ , can be normalized to  $\frac{\mathbf{F}^{5 \times 2} \mathbf{T}^{2 \times 1}}{Q_{(5)}} = \frac{[1 \ -2 \ 0 \ 2 \ -1]^t}{3}$ .

Finally, in order to reduce the estimation error, we use the local mean to tune the heterogeneity-projection maps. For  $HP_{H-map}$ , the horizontal heterogeneity-projection value at location  $(i, j)$  is denoted by  $HP_H(i, j)$  and for  $HP_{V-map}$ ,  $HP_V(i, j)$  denotes the vertical heterogeneity-projection value at location  $(i, j)$ . The tuned horizontal heterogeneity-projection value  $HP'_H(i, j)$  and the tuned vertical heterogeneity-projection value  $HP'_V(i, j)$  can be computed by the following operations:

$$HP'_H(i, j) = \frac{1}{6} \sum_{k=-2}^2 \delta_k HP_H(i, j + k)$$

$$HP'_V(i, j) = \frac{1}{6} \sum_{k=-2}^2 \delta_k HP_V(i + k, j)$$

where  $\delta_k = 2$  if  $k = 0$ ;  $\delta_k = 1$ , otherwise.

After performing the above adaptive heterogeneity-projection for mosaic images, the values  $HP'_H(i, j)$  and  $HP'_V(i, j)$  can be obtained for each mosaic pixel. In next two subsections, the six responses  $HP'_H(i, j)$ ,  $HP'_V(i, j)$ ,  $\Delta I_{dm}^H(i, j)$ ,  $\Delta I_{dm}^V(i, j)$ ,



$\Delta I_{dm}^{\frac{\pi}{4}}(i, j)$ , and  $\Delta I_{dm}^{\frac{-\pi}{4}}(i, j)$  will be used in our proposed new edge-sensing interpolation estimation for demosaicing mosaic images.

### 3.3.2 Edge-sensing interpolation estimation for $G$ channel

In this subsection, we present the proposed high-quality edge-sensing interpolation estimation for  $G$  channel. For exposition, let us take Fig. 1.1 to explain how to estimate the  $G$  channel value located at the center position of Fig. 1.1. Before performing the interpolation estimation for  $G$  channel, assume the gradient information of the current mosaic pixel at position  $(i, j)$  and the eight neighboring pixels at positions  $\Omega_n = \{(x, y) | (x, y) = (i \pm 1, j), (i \pm 2, j), (i, j \pm 1), (i, j \pm 2)\}$  have been extracted by using our proposed method described in last section, and the extracted nine horizontal gradient magnitudes and nine vertical gradient magnitudes are denoted by  $\Delta I_{dm}^H(*, *)$  and  $\Delta I_{dm}^V(*, *)$ , respectively.

According to the tuned horizontal heterogeneity-projection value  $HP'_H(i, j)$  and the tuned vertical heterogeneity-projection value  $HP'_V(i, j)$  of the current mosaic pixel at position  $(i, j)$ , the interpolation estimation scheme in our proposed edge-sensing demosaicing algorithm considers three cases, namely (1) horizontal variation as shown in Fig. 3.8(a), (2) vertical variation as shown in Fig. 3.8(b), and (3) the other variations as shown in Fig. 3.8(c). The arrows in Fig. 3.8 denote the data dependence.

In addition, in order to estimate  $I_{dm}^g(i, j)$  more accurately from its four neighbors, four proper weights in terms of gradient information are assigned to corresponding four spectral-correlation terms in the interpolation estimation. Considering the neighboring pixel located at location  $(i - 1, j)$ , if there is a horizontal edge passing

through it, i.e. the vertical gradient magnitude of the pixel at location  $(i - 1, j)$  is large, the color difference assumption [41, 62] reveals that the green component of this pixel makes less contribution to the estimation of green component for the current pixel at location  $(i, j)$ ; otherwise, the green component of this pixel makes more contribution to the estimation of green component for the current pixel. On the other hand, if the gradient magnitudes of the pixels at location  $(i - 2, j)$  and  $(i, j)$  are large, the pixel at location  $(i - 1, j)$  is located in a nonhomogeneous region and we claim that the pixel at location  $(i - 1, j)$  makes less green contribution to the estimation of green component for the current mosaic pixel at location  $(i, j)$ ; otherwise, the pixel at location  $(i - 1, j)$  is located in a homogeneous region and we claim that the pixel at location  $(i - 1, j)$  makes more green contribution to the estimation of green component for the current mosaic pixel. Combining the above analysis of gradient and direction effects, the weight of pixel at location  $(i - 1, j)$  can be given by  $w_g(i - 1, j) = \frac{1}{1 + \beta[\Delta I_{dm}^V(i - 2, j) + 2\Delta I_{dm}^V(i - 1, j) + \Delta I_{dm}^V(i, j)]}$ . Following the similar discussion, the weights of the four neighbors of the current pixel are expressed by  $w_g(i - 1, j) = \frac{1}{1 + \beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^V(i - k, j)]}$ ,  $w_g(i + 1, j) = \frac{1}{1 + \beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^V(i + k, j)]}$ ,  $w_g(i, j - 1) = \frac{1}{1 + \beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^H(i, j - k)]}$ , and  $w_g(i, j + 1) = \frac{1}{1 + \beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^H(i, j + k)]}$  where  $\delta'_k = 2$  if  $k = 1$ ;  $\delta'_k = 1$ , otherwise. The determination of the parameter  $\beta$  is discussed in Appendix B.

According to the above description, the value of  $I_{dm}^g(i, j)$  of the current pixel at

location  $(i, j)$  can be estimated by the following rules:

$$I_{dm}^g(i, j) = I_{mo}^b(i, j) + \frac{\sum_{(x,y) \in \Omega_g} w_g(x, y) D_g(x, y)}{\sum_{(x,y) \in \Omega_g} w_g(x, y)}$$

$$\Omega_g = \begin{cases} \{(i \pm 1, j)\} & \text{if } HP'_V(i, j) < \alpha HP'_H(i, j) \text{ (horizontal variation).} \\ \{(i, j \pm 1)\} & \text{if } HP'_H(i, j) < \alpha HP'_V(i, j) \text{ (vertical variation).} \\ \{(i \pm 1, j), (i, j \pm 1)\} & \text{otherwise, (other variations).} \end{cases}$$

where for  $(x_1, y_1) \in \{(i \pm 1, j)\}$ ,  $D_g(x_1, y_1) = I_{mo}^g(x_1, y_1) - \frac{I_{mo}^b(x_1+1, y_1) + I_{mo}^b(x_1-1, y_1)}{2}$ ;

for  $(x_2, y_2) \in \{(i, j \pm 1)\}$ ,  $D_g(x_2, y_2) = I_{mo}^g(x_2, y_2) - \frac{I_{mo}^b(x_2, y_2+1) + I_{mo}^b(x_2, y_2-1)}{2}$ ; the

determination of the parameter  $\alpha$  is discussed in Appendix B.

After performing the above edge-sensing interpolation estimation for  $G$  channel, the  $G$  channel of the demosaiced image is fully populated. In next subsection, the fully populated  $G$  channel of the image will be used to assist the interpolation of  $R$  and  $B$  channels.

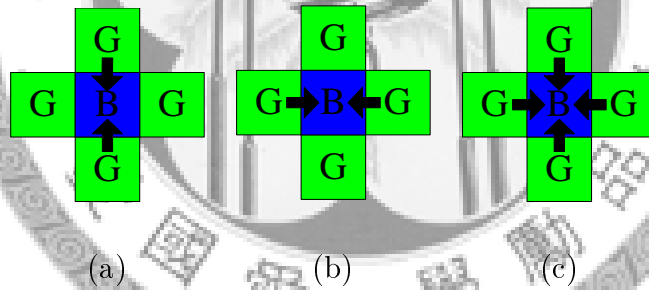


Fig. 3.8: The data dependence of our proposed interpolation estimation for  $G$  channel. (a) Horizontal variation (vertical edge). (b) Vertical variation (horizontal edge). (c) The other variations.

### 3.3.3 Edge-sensing interpolation estimation for $R$ and $B$ channels

Because the number of  $R$  pixels or  $B$  pixels is less than  $G$  in the mosaic image, the interpolation estimation for  $R$  and  $B$  channels should be partitioned into two steps: (1) estimating the red values at blue pixels, and vice versa; (2) then, recovering missing red and blue values at green pixels. Because the interpolation estimation for  $R$  channel is the same as it for  $B$  channel, we thus only present it for  $R$  channel. In our proposed interpolation estimation for  $R$  and  $B$  channels, the fully populated  $G$  channel is used to assist the interpolation of  $R$  and  $B$  channels. For convenience, we still use Fig. 1.1 to explain how to estimate the  $R$  channel value for the current pixel at position  $(i, j)$ .

Similar to the interpolation estimation for  $G$  channel, assume the gradient information of the current mosaic pixel at position  $(i, j)$  and the eight pixels at positions  $\Omega'_n = \{(x, y) | (x, y) = (i \pm 1, j \pm 1), (i \pm 2, j \pm 2)\}$ , respectively, have been extracted by using our proposed method. By the same argument, in order to estimate  $I_{dm}^r(i, j)$  more accurately from its four red neighbors, four proper weights in terms of gradient/direction information are assigned to the corresponding four spectral-correlation terms in the interpolation estimation. For estimating  $I_{dm}^r(i, j)$ , we consider four diagonal variations of the mosaic pixel at position  $(i, j)$  to determine the four weights which will be used in the proposed edge-sensing estimation of  $I_{dm}^r(i, j)$ . Considering the neighboring pixel at location  $(i - 1, j - 1)$ , if there is a  $\frac{\pi}{4}$ -diagonal edge passing through it, i.e. the  $\frac{-\pi}{4}$ -diagonal gradient magnitude of the pixel at location  $(i - 1, j - 1)$  is large, and then it claims that the red component of this pixel makes less contribution to the estimation of red component for the current pixel

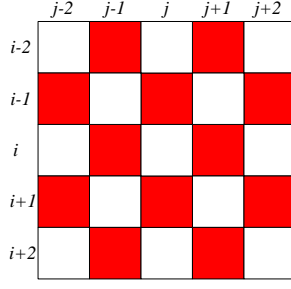


Fig. 3.9: The patterns of the  $R$  channel.

at location  $(i, j)$ ; otherwise, it claims that the red component of this pixel makes more contribution to the estimation of red component for the current pixel. On the contrary, if the gradient magnitudes of the pixel at location  $(i - 1, j - 1)$  and  $(i - 2, j - 2)$  are large, i.e. the pixel at location  $(i - 1, j - 1)$  is inside nonhomogeneous region, the color difference assumption indicates that the red component of this pixel makes less contribution to the estimation of red component for the current pixel at location  $(i, j)$ ; otherwise, the red component of this pixel makes more contribution to the estimation of red component for the current pixel. Following the above analysis, the weights of the four diagonal red neighbors of the current pixel

can be expressed by  $w_r(i - 1, j - 1) = \frac{1}{1 + \beta \left[ \sum_{k=0}^2 \delta'_k \Delta I_{dm}^{\frac{-\pi}{4}}(i-k, j-k) \right]}$ ,  $w_r(i - 1, j + 1) = \frac{1}{1 + \beta \left[ \sum_{k=0}^2 \delta'_k \Delta I_{dm}^{\frac{\pi}{4}}(i-k, j+k) \right]}$ ,  $w_r(i + 1, j - 1) = \frac{1}{1 + \beta \left[ \sum_{k=0}^2 \delta'_k \Delta I_{dm}^{\frac{-\pi}{4}}(i+k, j-k) \right]}$ , and  $w_r(i + 1, j + 1) = \frac{1}{1 + \beta \left[ \sum_{k=0}^2 \delta'_k \Delta I_{dm}^{\frac{\pi}{4}}(i+k, j+k) \right]}$  where  $\delta'_k = 2$  if  $k = 1$ ;  $\delta'_k = 1$ , otherwise. Based on the four weights obtained above and the color difference concept, the demosaiced full

red color for the blue pixel in the mosaic image,  $I_{dm}^r(i, j)$ , can be estimated by

$$I_{dm}^r(i, j) = I_{dm}^g(i, j) + \frac{\sum_{(x,y) \in \Omega_r} w_r(x, y) D_r(x, y)}{\sum_{(x,y) \in \Omega_r} w_r(x, y)}$$

where  $\Omega_r = \{(x, y) | (i \pm 1, j \pm 1)\}$ ; for  $(x, y) \in \{(i \pm 1, j \pm 1)\}$ ,  $D_r(x, y) = I_{mo}^r(x, y) - I_{dm}^g(x, y)$ .

After describing how to estimate demosaiced full red colors for those blue pixels

in the mosaic image, we now introduce how to estimate full red colors for those green pixels. Fig. 3.9 illustrates the pattern of  $R$  channel. Referring to Fig. 3.9, the full red color for the green pixel,  $I_{dm}^r(i, j)$ , can be estimated by the following rules:

$$I_{dm}^r(i, j) = I_{dm}^g(i, j) + \frac{\sum_{(x,y) \in \Omega'_r} w_r(x, y) D_r(x, y)}{\sum_{(x,y) \in \Omega'_r} w_r(x, y)}$$

$$\Omega'_r = \begin{cases} \{(i \pm 1, j)\} & \text{if } HP'_V(i, j) < \alpha HP'_H(i, j) \text{ (horizontal variation).} \\ \{(i, j \pm 1)\} & \text{if } HP'_H(i, j) < \alpha HP'_V(i, j) \text{ (vertical variation).} \\ \{(i \pm 1, j), (i, j \pm 1)\} & \text{otherwise (other variations).} \end{cases}$$

where for  $(x, y) \in \{(i \pm 1, j), (i, j \pm 1)\}$ , we perform  $D_r(x, y) = I_{mo}^r(x, y) - I_{dm}^g(x, y)$ .

If  $k = 1$ , we set  $\delta'_k = 2$ ; otherwise, we set  $\delta'_k = 1$ . We further perform  $w_r(i - 1, j) =$

$$\frac{1}{1+\beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^V(i-k, j)]}, w_r(i+1, j) = \frac{1}{1+\beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^V(i+k, j)]}, w_r(i, j-1) = \frac{1}{1+\beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^H(i, j-k)]},$$

$$\text{and } w_r(i, j+1) = \frac{1}{1+\beta[\sum_{k=0}^2 \delta'_k \Delta I_{dm}^H(i, j+k)]}.$$

After presenting our proposed new edge-sensing demosaicing algorithm, experimental results in next section will illustrate the quality advantage of our proposed new edge-sensing demosaicing algorithm.

### 3.4 Experimental results

In this section, based on twenty-four popular testing mosaic images, some experimental results are demonstrated to show that our proposed new demosaicing algorithm has better image quality performance when compared with the previous seven algorithms by Pei and Tam [62], Lu and Tan [41], Lukac and Plataniotis [48], Dubois [20], Lukac *et al.* [53], Chung and Chan [13], and Tsai and Song [78], respectively. The concerned algorithms are implemented on the IBM compatible computer with Intel Core 2 Duo CPU 1.6GHz and 1GB RAM. The operating system used is MS-Windows XP and the program developing environment is Borland C++ Builder 6.0.



Fig. 3.10: The twenty-four testing images from Kodak PhotoCD [85].

Our program has been uploaded in [86].

Fig. 3.10 illustrates the twenty-four testing images from Kodak PhotoCD [85]. In our experiments, the twenty-four testing images shown in Fig. 3.10 are first down-sampled to obtain the mosaic images. Furthermore, the boundaries of the image are dealt with using the mirroring method. Here, we adopt two objective image quality measures, color peak signal-to-noise ratio (CPSNR) and S-CIELAB  $\Delta E_{ab}^*$  metric [33, 41], and one subjective image quality measure, color artifacts, to justify the better quality performance of our proposed novel demosaicing algorithm. The CPSNR of a color image with size  $X \times Y$  is defined by

$$\text{CPSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}$$

$$\text{MSE} = \frac{1}{3XY} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \sum_{c \in C} [I_{ori}^c(i, j) - I_{dm}^c(i, j)]^2$$

where  $C = \{r, g, b\}$ ;  $I_{ori}^r(i, j)$ ,  $I_{ori}^g(i, j)$ , and  $I_{ori}^b(i, j)$  denote the three color components of the pixel at location  $(i, j)$  in the original full color image;  $I_{dm}^r(i, j)$ ,  $I_{dm}^g(i, j)$ , and  $I_{dm}^b(i, j)$  denote the three color components of the pixel at location  $(i, j)$  in the demosaiced image. The S-CIELAB  $\Delta E_{ab}^*$  of a color image with size  $X \times Y$  is defined

by

$$\Delta E_{ab}^* = \frac{1}{XY} \sum_{i=0}^{X-1} \sum_{j=0}^{Y-1} \left\{ \sqrt{\sum_{c \in \mathcal{L}} [LAB_{ori}^c(i, j) - LAB_{dm}^c(i, j)]^2} \right\}$$

where  $\mathcal{L} = \{L, a, b\}$ ;  $LAB_{ori}^L(i, j)$ ,  $LAB_{ori}^a(i, j)$ , and  $LAB_{ori}^b(i, j)$  denote the three CIELAB color components of the pixel at location  $(i, j)$  in the original full color image;  $LAB_{dm}^L(i, j)$ ,  $LAB_{dm}^a(i, j)$ , and  $LAB_{dm}^b(i, j)$  denote the three CIELAB color components of the pixel at location  $(i, j)$  in the demosaiced image.

For fairness, among the eight concerned demosaicing algorithms, the three existing algorithms [13, 20, 53] still apply their own refinement schemes; the other four existing algorithms [41, 48, 62, 78] and our proposed algorithm utilize the post-processing approach by Lukac *et al.* [46] to enhance the demosaiced image quality. Based on twenty-four testing mosaic images, Table 3.1 and Table 3.2 demonstrate the demosaiced image quality comparison in terms of CPSNR and S-CIELAB  $\Delta E_{ab}^*$  among our proposed algorithm and the other seven demosaicing algorithms, respectively. In Table 3.1 and Table 3.2, the entries with the best CPSNR or the least  $\Delta E_{ab}^*$  are highlighted by bold black. It is observed that in average, our proposed demosaicing algorithm has the best demosaiced image quality in terms of CPSNR and  $\Delta E_{ab}^*$ .

Next, we adopt the subjective image visual measure, color artifacts, to demonstrate the quality advantage of our proposed demosaicing algorithm. After performing the demosaicing processing, some degree of color artifacts may happen on edges or textures of the demosaiced image. We first take the magnified subimages cut from the testing image No. 19 as shown in Fig. 3.11 to compare the visual effect among the concerned eight algorithms. Figs. 3.11(a)–(i) illustrate the nine magnified subimages cut from the original testing image No. 19, the demosaiced



Table 3.1: CPSNR quality comparison for twenty-four testing images.

Method	[62]	[41]	[48]	[20]	[53]	[13]	[78]	Proposed
Image01	36.119	39.134	37.647	38.017	37.414	37.995	39.128	<b>39.879</b>
Image02	38.155	38.737	39.701	39.319	<b>40.012</b>	39.028	39.553	39.790
Image03	40.769	41.092	41.964	41.693	<b>42.393</b>	41.492	42.084	42.314
Image04	39.402	39.466	40.451	40.515	<b>40.786</b>	39.735	40.042	40.147
Image05	36.795	35.830	36.983	37.875	37.778	36.962	37.779	<b>37.955</b>
Image06	37.160	37.838	38.066	39.959	37.445	39.435	40.292	<b>40.754</b>
Image07	40.516	40.713	41.899	41.973	<b>42.478</b>	40.89	41.51	41.668
Image08	32.578	34.805	35.487	35.272	34.572	35.723	37.067	<b>37.558</b>
Image09	40.743	41.363	42.024	42.174	42.320	42.030	42.683	<b>42.914</b>
Image10	41.338	41.026	41.582	42.166	42.188	41.495	41.979	<b>42.323</b>
Image11	38.278	38.406	39.367	39.805	39.139	39.331	40.160	<b>40.556</b>
Image12	40.755	41.819	42.415	42.977	42.274	42.669	43.300	<b>43.541</b>
Image13	34.358	33.139	34.160	35.127	34.292	34.651	35.595	<b>36.224</b>
Image14	35.425	35.013	36.556	36.035	<b>37.066</b>	35.747	36.199	36.244
Image15	38.293	38.461	39.264	39.265	39.245	38.808	39.008	<b>39.326</b>
Image16	39.907	41.697	41.547	43.705	40.452	43.293	43.926	<b>44.314</b>
Image17	41.068	40.587	40.991	41.706	41.585	41.244	41.752	<b>42.050</b>
Image18	37.020	35.747	36.668	37.266	37.089	36.688	37.262	<b>37.683</b>
Image19	37.284	39.940	40.160	40.416	39.335	40.399	41.310	<b>41.752</b>
Image20	40.075	40.266	41.006	40.386	41.175	40.345	41.263	<b>41.580</b>
Image21	37.886	38.236	38.736	38.857	38.735	38.517	39.711	<b>40.165</b>
Image22	37.514	37.521	38.219	38.223	<b>38.518</b>	37.650	38.139	38.445
Image23	41.055	41.219	42.007	42.193	<b>42.928</b>	41.526	41.776	42.069
Image24	34.986	33.569	34.389	<b>35.338</b>	34.89	34.778	34.779	35.291
Average	38.228	38.491	39.220	39.594	39.338	39.185	39.846	<b>40.189</b>

Table 3.2: S-CIELAB  $\Delta E_{ab}^*$  quality comparison for twenty-four testing images.

Method	[62]	[41]	[48]	[20]	[53]	[13]	[78]	Proposed
Image01	1.7773	1.4103	1.6251	1.5592	1.5948	1.5276	1.3680	<b>1.2739</b>
Image02	1.7874	1.6021	1.5501	1.7603	1.5337	1.6490	1.5584	<b>1.5327</b>
Image03	1.0050	0.9238	0.9311	0.9465	<b>0.8761</b>	0.9562	0.8897	0.8939
Image04	1.2817	1.1987	1.1788	1.2177	<b>1.1507</b>	1.2702	1.2119	1.1950
Image05	2.2565	2.1473	2.2298	2.0353	1.9759	2.2026	1.9745	<b>1.9432</b>
Image06	1.3383	1.1518	1.2577	1.0926	1.2867	1.0883	1.0131	<b>0.9750</b>
Image07	1.2586	1.1609	1.1060	1.0755	<b>1.0107</b>	1.2080	1.1252	1.1213
Image08	2.1716	1.6341	1.7945	1.7774	1.8203	1.7101	1.4984	<b>1.4329</b>
Image09	0.8860	0.8208	0.8419	0.8078	0.7935	0.8205	0.7796	<b>0.7736</b>
Image10	0.8452	0.8195	0.8558	0.8170	0.7911	0.8475	0.8005	<b>0.7902</b>
Image11	1.5881	1.3496	1.4200	1.3677	1.3768	1.4086	1.2814	<b>1.2508</b>
Image12	0.7489	0.6641	0.6823	0.6569	0.6663	0.6649	0.6261	<b>0.6202</b>
Image13	2.3943	2.2976	2.5256	2.4274	2.4413	2.4568	2.2369	<b>2.0849</b>
Image14	2.0050	1.8286	1.7552	1.8755	<b>1.6659</b>	1.8928	1.7252	1.7157
Image15	1.4960	1.3949	1.3969	1.4358	<b>1.3592</b>	1.4729	1.4038	1.3785
Image16	1.1752	0.9422	1.0296	0.8828	1.0934	0.8922	0.8472	<b>0.8306</b>
Image17	1.4391	1.3341	1.3937	1.3296	<b>1.2998</b>	1.3934	1.3269	1.3160
Image18	2.3071	2.3321	2.3421	2.2634	2.2438	2.4106	2.2796	<b>2.2193</b>
Image19	1.4089	1.1726	1.2415	1.2176	1.2403	1.2341	1.1383	<b>1.0986</b>
Image20	1.0446	0.9674	0.9880	1.0351	0.9467	1.0340	0.9456	<b>0.9229</b>
Image21	1.3317	1.2030	1.2934	1.2943	1.2460	1.3000	1.1681	<b>1.1197</b>
Image22	1.4972	1.4718	1.4323	<b>1.3643</b>	1.4096	1.5187	1.4541	1.4219
Image23	1.0353	0.9821	0.9564	0.9263	<b>0.9045</b>	1.0153	0.9854	0.9741
Image24	1.4904	1.5017	1.5346	1.4323	1.4187	1.5077	1.4201	<b>1.3697</b>
Average	1.4821	1.3463	1.3901	1.3583	1.3392	1.3951	1.2941	<b>1.2606</b>

Table 3.3: The average execution-time of eight demosaicing algorithms for twenty-four testing mosaic images.

Method	[62]	[41]	[48]	[20]	[53]	[13]	[78]	Proposed
Time(s)	0.607	1.704	1.303	1.406	1.492	1.052	1.469	1.442

image obtained by Pei and Tam’s demosaicing algorithm, the one obtained by Lu and Tan’s demosaicing algorithm, the one obtained by Lukac and Plataniotis’ algorithm, the one obtained by Dubois’ demosaicing algorithm, the one obtained by Lukac *et al.*’s algorithm, the one obtained by Chung and Chan’s algorithm, the one obtained by Tsai and Song’s demosaicing algorithm, and the one obtained by our proposed demosaicing algorithm, respectively. Comparing the visual effect between the original full color image and the one in Figs. 3.11(b)–(i), it is observed that our proposed demosaicing algorithm creates less color artifacts when compared with the other seven demosaicing algorithms. Then, we take the magnified subimages cut from the testing image No. 8 to depict the visual comparison. Figs. 3.12(a)–(i) illustrate the magnified subimages cut from the original full color testing image No. 8 and the eight demosaiced images. From visual comparison, it is observed that our proposed demosaicing algorithm produces less color artifacts when compared with the other seven demosaicing algorithms.

Furthermore, the average execution-time of the eight concerned demosaicing algorithms for the twenty-four testing mosaic images are shown in Table 3.3. It is observed that the average execution-time of our proposed demosaicing algorithm is moderate when compared with the other seven algorithms. However, our proposed algorithm has the best demosaiced image quality performance among the eight algorithms.

Finally, in order to demonstrate the advantage of our proposed adaptive heterogeneity-projection, the mask-use ratio  $\mathfrak{R}_{(N)}$  for demosaicing a mosaic image with size  $X \times Y$  is defined as  $\mathfrak{R}_{(N)} = \frac{T_{(N)}}{XY} \times 100\%$  where  $T_{(N)}$  denotes the number of times which the mask with size  $N$  is adopted during the demosaicing processing. Based on twenty-four testing mosaic images, in average, the four mask-use ratios for  $N$  in  $\{5, 7, 9, 11\}$  are  $\mathfrak{R}_{(5)} = 28.58\%$ ,  $\mathfrak{R}_{(7)} = 5.30\%$ ,  $\mathfrak{R}_{(9)} = 4.28\%$ , and  $\mathfrak{R}_{(11)} = 61.84\%$ , respectively. According to the above statistical data,  $N = 11$  could be set for about 62% cases, but for the other 38% cases, smaller mask sizes could be more suitable for obtaining the heterogeneity-projection values.

### 3.5 Summary

Without demosaicing processing, this chapter first proposes a new approach to extract more accurate gradient/edge information on mosaic images directly. Next, based on spectral-spatial correlation [78], a novel adaptive heterogeneity-projection with proper mask size for each pixel is presented. Combining the extracted gradient/edge information and the adaptive heterogeneity-projection values, a new edge-sensing demosaicing algorithm is presented. Some experimental results have been carried out to demonstrate the quality advantage in terms of CPSNR and S-CIELAB  $\Delta E_{ab}^*$  of our proposed new demosaicing algorithm when compared with several recently published algorithms.

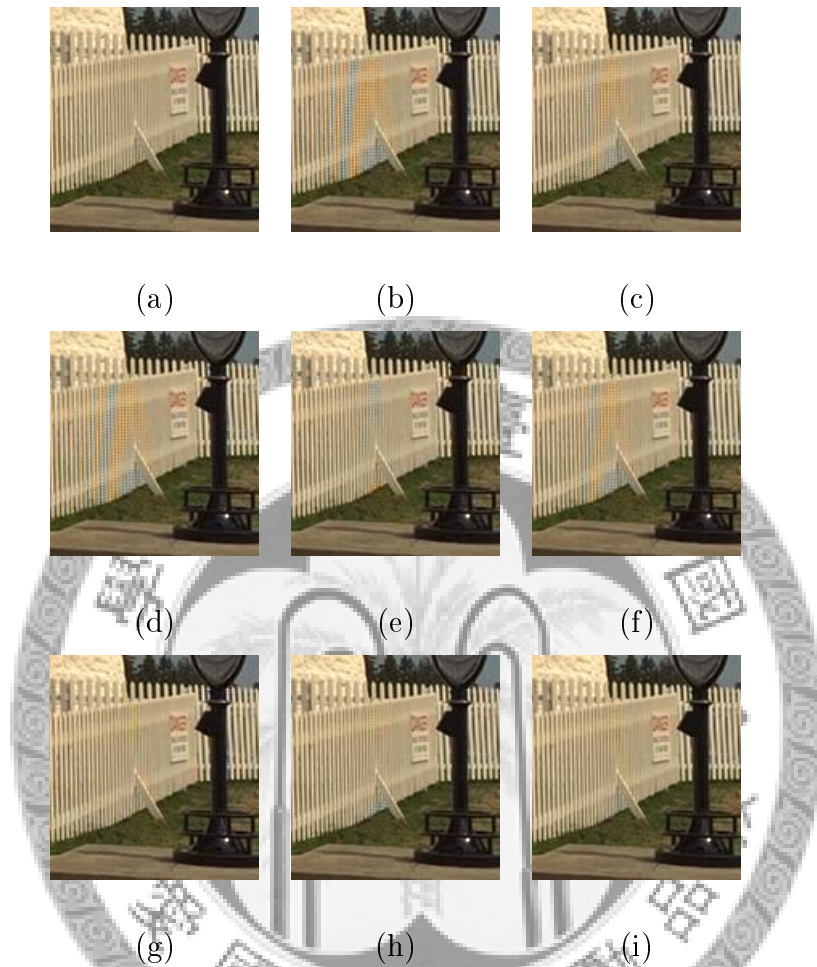


Fig. 3.11: The magnified subimages cut from the testing image No. 19. (a) Original full color image and the demosaiced images obtained from (b) Pei and Tam's algorithm. (c) Lu and Tan's algorithm. (d) Lukac and Plataniotis' algorithm. (e) Dubois' algorithm. (f) Lukac *et al.*'s algorithm. (g) Chung and Chan's algorithm. (h) Tsai and Song's algorithm. (i) our proposed algorithm.

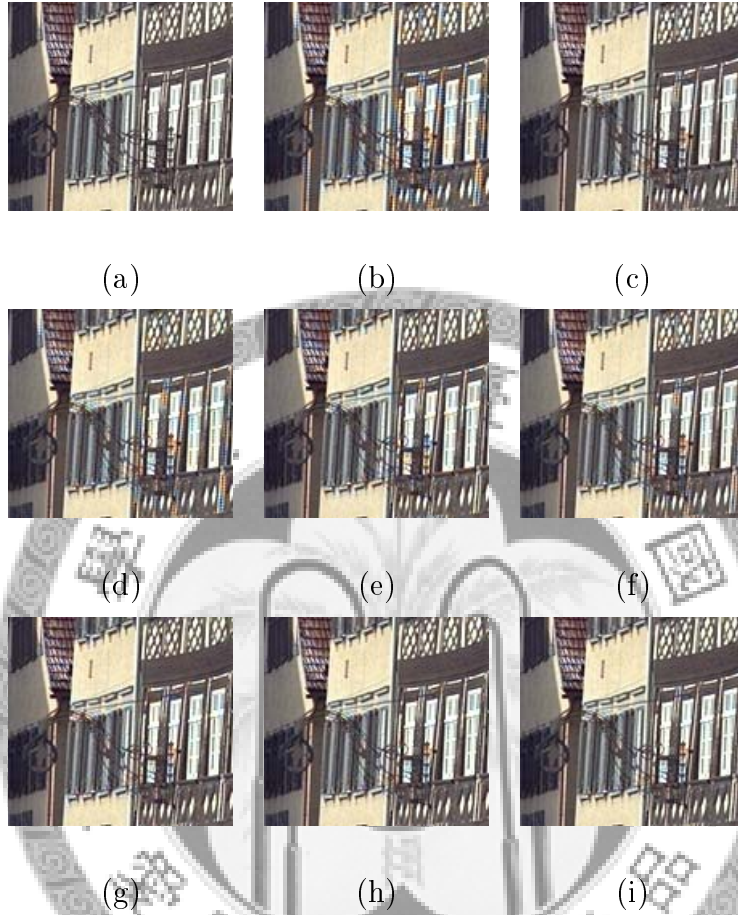


Fig. 3.12: The magnified subimages cut from the testing image No. 8. (a) Original full color image and the demosaiced images obtained from (b) Pei and Tam's algorithm. (c) Lu and Tan's algorithm. (d) Lukac and Plataniotis' algorithm. (e) Dubois' algorithm. (f) Lukac *et al.*'s algorithm. (g) Chung and Chan's algorithm. (h) Tsai and Song's algorithm. (i) our proposed algorithm.

# Chapter 4

## Joint Demosaicing and Arbitrary-Ratio Resizing Algorithm for Bayer Color Filter Array Based on DCT Approach

### 4.1 Preliminaries

The purpose of mosaic image resizing algorithm is to resize a mosaic image. Since using the optical hardware resizing approach costs too much, camera manufacturers have adopted the software resizing approach. Here, the terms “resize” and “zoom” are used exchangeably. Previously, several resizing algorithms on mosaic images have been developed. The previously developed zooming algorithms can be roughly classified into three approaches. Based on the local adaptive zooming concept, Battiato *et al.* [6] presented the first zooming algorithm. Based on the adaptive edge-

sensing mechanism, Lukac *et al.* [44, 50] presented two better zooming algorithms to improve the zoomed image quality. Later, Lukac and Plataniotis [52] proposed a computation-saving interpolation method to meet the real-time surveillance requirement. Recently, the approach using joint demosaicing process and zooming process was proposed by Chung and Chen [14] and Zhang and Zhang [83] independently. Unfortunately, all of them only focus on the quad-zooming process. The motivation of this research is to develop a new joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images.

In this chapter, a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. First, more accurate edge information is extracted from the mosaic image by using the adaptive heterogeneity projection masks and SL-based masks [17]. Next, the fully populated G color plane is constructed by using the edge-sensing approach and color difference idea. Instead of interpolating the R and B color planes directly, the green-red color difference plane and green-blue color difference plane are therefore interpolated in order to reduce the estimation error. Then, based on the composite length DCT technique [59], the three constructed planes can be resized to obtain arbitrary-ratio sized ones. Finally, the resized red and blue color planes are gotten by using the two resized color difference planes and the resized green color plane, and then the arbitrary-ratio resized full color image is followed. The flowchart of the proposed resizing algorithm is illustrated in Fig. 4.1. This is the first time that such a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. Based on twenty-four popular testing mosaic images, the proposed algorithm has better image quality performance in terms of two objective color image quality measures, the CPSNR and the S-



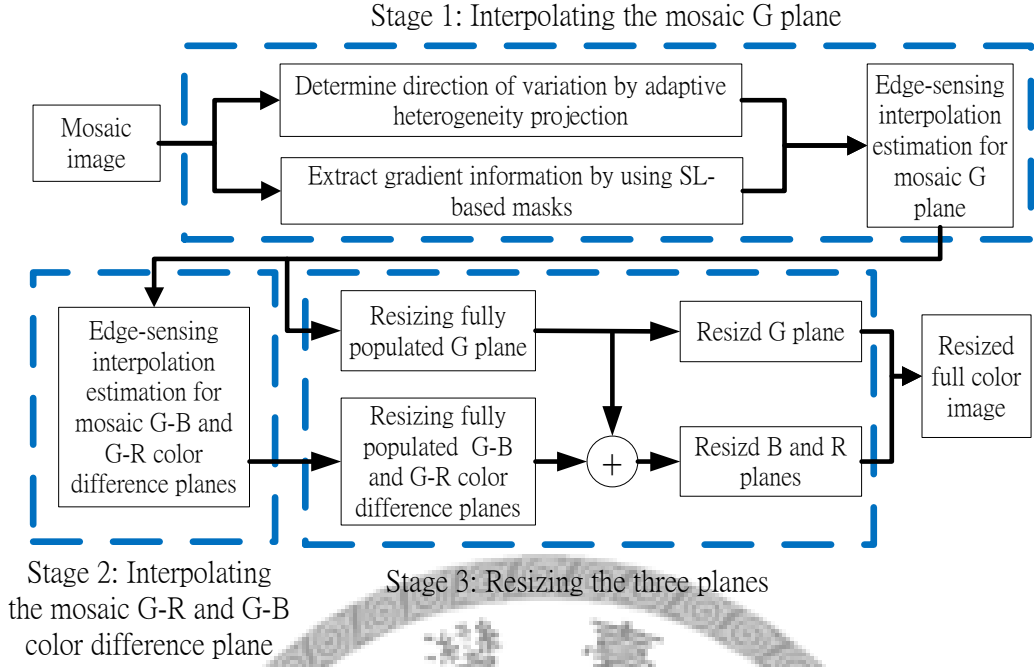


Fig. 4.1: The flowchart of the proposed arbitrary-ratio resizing algorithm.

CIELAB  $\Delta E_{ab}^*$ , and one subjective color image quality measure, the color artifacts, when compared with three native algorithms which are the combinations of three well-know demosaicing methods [13, 48, 62] one existing resizing method [59].

## 4.2 Proposed joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images

Our proposed algorithm consists of the following three stages: (1) interpolating the mosaic G plane to construct the fully populated G plane by using the edge-sensing interpolation estimation; (2) interpolating the mosaic G-R and G-B color difference planes to construct the fully populated G-R and G-B color difference planes, respectively; (3) resizing the three constructed planes mentioned above to obtain the arbitrary-ratio sized ones, and then based on the three resided planes,

recovering the resized R and B planes to obtain arbitrary-ratio resized full color image.

### 4.2.1 Stage 1: Interpolating the mosaic G plane

In this subsection, the interpolation for the mosaic G plane  $I_{mo}^g$  to construct the fully populated G plane  $I_{dm}^g$  by using the edge-sensing approach and color difference idea is presented. For exposition, let us take the central pixel at position  $(i, j)$  in Fig. 1.1 as the representative to explain how to estimate the G color value  $I_{dm}^g(i, j)$  from its four neighboring pixels with movement  $\Omega_g = \{(x, y) | (x, y) = (i \pm 1, j), (i, j \pm 1)\}$ . First, according to the tuned horizontal heterogeneity projection value  $HP'_H(i, j)$  and the tuned vertical heterogeneity projection value  $HP'_V(i, j)$  of the current pixel, three cases, namely the horizontal variation as shown in Fig. 3.8(a), the vertical variation as shown in Fig. 3.8(b), and the other variations as shown in Fig. 3.8(c), are considered in the interpolation estimation phase for  $I_{mo}^g(i, j)$ . The arrows in Fig. 3.8 denote the relevant data dependence.

Further, in order to estimate  $I_{dm}^g(i, j)$  more accurately, four proper weights in terms of the gradient magnitude are assigned to the corresponding four pixels in the interpolation estimation phase. Given a pixel at position  $(x, y)$ , based on the horizontal and vertical gradient magnitudes, its horizontal and vertical weights can be determined by  $w_g(H, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^H(x, y+k)}$  and  $w_g(V, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^V(x+k, y)}$ , respectively, where  $\delta_k = 3$  if  $k = 0$ ;  $\delta_k = 1$ , otherwise. Considering the neighboring pixel located at position  $(i - 1, j)$ , if the vertical gradient magnitude is large, i.e. there is a horizontal edge passing through it, based on the color difference assumption [41, 62], it reveals that the G component of this pixel makes less contribution to

estimate that of the current pixel; otherwise, it reveals that the G component of this pixel makes more contribution to estimate that of the current pixel. According to the above analysis, the vertical weight  $w_g(V, i-1, j) = \frac{1}{1+\Delta I_{dm}^V(i-2, j)+3\Delta I_{dm}^V(i-1, j)+\Delta I_{dm}^V(i, j)}$  is selected for the pixel at position  $(i-1, j)$ . By the same argument, the weights of the other three neighbors are selected by  $w_g(V, i+1, j)$ ,  $w_g(H, i, j-1)$ , and  $w_g(H, i, j+1)$ , respectively. Consequently, the value of  $I_{dm}^g(i, j)$  can be estimated by

$$I_{dm}^g(i, j) = I_{mo}^b(i, j) + \frac{\sum_{(d,x,y) \in \xi_g} w_g(d, x, y) D_{gb}(x, y)}{\sum_{(d,x,y) \in \xi_g} w_g(d, x, y)}$$

$$\xi_g = \begin{cases} \xi_1 & \text{if } HP'_V(i, j) < \alpha HP'_H(i, j) \\ \xi_2 & \text{if } HP'_H(i, j) < \alpha HP'_V(i, j) \\ \xi_1 \cup \xi_2 & \text{otherwise} \end{cases}$$

where  $\xi_1 = \{(V, i \pm 1, j)\}$  and  $\xi_2 = \{(H, i, j \pm 1)\}$ ; for  $(d_1, x_1, y_1) \in \xi_1$ ,  $D_{gb}(x_1, y_1) = I_{mo}^g(x_1, y_1) - \frac{1}{2} \sum_{k \in \{\pm 1\}} I_{mo}^b(x_1 + k, y_1)$ ; for  $(d_2, x_2, y_2) \in \xi_2$ ,  $D_{gb}(x_2, y_2) = I_{mo}^g(x_2, y_2) - \frac{1}{2} \sum_{k \in \{\pm 1\}} I_{mo}^b(x_2, y_2 + k)$ ; the parameter  $\alpha$  is set to  $\alpha = 0.55$  empirically.

Finally, a new proposed refinement approach, which combines the concept of the local color ratios [46] and the extracted more accurate edge information, is addressed to refine the fully populated G plane. For the current pixel at position  $(i, j)$ , its G value  $I_{dm}^g(i, j)$  can be refined by the following rule:

$$I_{dm}^g(i, j) = -\beta + (I_{mo}^b(i, j) + \beta) \frac{\sum_{(d,x,y) \in \xi'_g} w_g(d, x, y) R_{gb}(x, y)}{\sum_{(d,x,y) \in \xi'_g} w_g(d, x, y)}$$

$$\xi'_g = \begin{cases} \xi'_1 & \text{if } HP'_V(i, j) < \alpha HP'_H(i, j) \\ \xi'_2 & \text{if } HP'_H(i, j) < \alpha HP'_V(i, j) \\ \xi'_1 \cup \xi'_2 & \text{otherwise} \end{cases}$$

where  $\xi'_1 = \{(V, i+k, j) | k \in \{0, \pm 2\}\}$ ;  $\xi'_2 = \{(H, i, j+k) | k \in \{0, \pm 2\}\}$ ;  $R_{gb}(x, y) = \frac{I_{dm}^g(x, y) + \beta}{I_{mo}^b(x, y) + \beta}$ ; the parameter  $\beta$  is set to  $\beta = 256$  empirically.

	$j-3$	$j-2$	$j-1$	$j$	$j+1$	$j+2$	$j+3$
$i-3$	$D_{gr}$		$D_{gr}$		$D_{gr}$		$D_{gr}$
$i-2$							
$i-1$	$D_{gr}$		$D_{gr}$		$D_{gr}$		$D_{gr}$
$i$							
$i+1$	$D_{gr}$		$D_{gr}$		$D_{gr}$		$D_{gr}$
$i+2$							
$i+3$	$D_{gr}$		$D_{gr}$		$D_{gr}$		$D_{gr}$

Fig. 4.2: The pattern of the mosaic G-R color difference plane.

After performing the above the edge-sensing interpolation estimation, the fully populated G plane is constructed. In next subsection, the fully populated G plane will be utilized to assist the interpolation of the mosaic G-R and G-B color difference planes.

#### 4.2.2 Stage 2: Interpolating the mosaic G-R and G-B color difference planes

Instead of interpolating the R and B color planes directly, we interpolate the G-R and G-B color difference planes because the color difference plane is much smoother than the original color plane and it would alleviate the amplification of the estimation error in the later resizing stage. Since the interpolation for G-R color difference plane is the same as that for G-B color difference plane, we only present it for G-R color difference plane consisting of three steps.

In Step 1, according to the mosaic image  $I_{mo}$  (see Fig. 1.1) and the fully populated G plane  $I_{dm}^g$ , the mosaic G-R plane can be obtained by  $D_{gr}(i_r, j_r) = I_{dm}^g(i_r, j_r) - I_{mo}^r(i_r, j_r)$  where  $(i_r, j_r) \in \{(i \pm 2m + 1, j \pm 2n + 1)\}$ . After performing Step 1, Fig. 4.2 illustrates the pattern of the obtained mosaic G-R color difference

plane for the positions depicted in in gray cells. The G-R color difference plane interpolation estimation for the other positions consists of two steps: Step 2: interpolating the G-R color difference values of the pixels at positions  $(i \pm 2m, j \pm 2n)$  in Fig. 4.2; Step 3: interpolating the G-R color difference values of the pixels at positions  $(i \pm 2m, j \pm 2n + 1)$  and  $(i \pm 2m + 1, j \pm 2n)$ . For simplicity, the central pixel at position  $(i, j)$  in Fig. 4.2 is taken as the representative to explain the G-R color difference plane interpolation performed in Step 2. The G-R color difference value  $D_{gr}(i, j)$  can be estimated from its four neighboring pixels, which have been interpolated in last subsection, with movement  $\Omega_r = \{(x, y) | (x, y) = (i \pm 1, j \pm 1)\}$ . In order to estimate  $D_{gr}(i, j)$  more accurately, the gradient magnitudes of four diagonal variations are considered to determine the proper four weights. Given a pixel at position  $(x, y)$ , its  $\frac{-\pi}{4}$ -diagonal and  $\frac{\pi}{4}$ -diagonal weights can be determine by  $w_{gr}(\frac{-\pi}{4}, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^{\frac{-\pi}{4}}(i+k, j+k)}$  and  $w_{gr}(\frac{\pi}{4}, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^{\frac{\pi}{4}}(i-k, j+k)}$ , respectively, where  $\delta_k = 3$  if  $k = 0$ ;  $\delta_k = 1$ , otherwise. Thus, the four weights of the four diagonal neighbors of the current pixel at position  $(i, j)$  are denoted by  $w_{gr}(\frac{-\pi}{4}, i-1, j-1)$ ,  $w_{gr}(\frac{\pi}{4}, i-1, j+1)$ ,  $w_{gr}(\frac{\pi}{4}, i+1, j-1)$ , and  $w_{gr}(\frac{-\pi}{4}, i+1, j+1)$ , respectively. Based on the four weights, the G-R color difference value at position  $(i, j)$  can be estimated by

$$D_{gr}(i, j) = \frac{\sum_{(d,x,y) \in \xi_{gr}} w_{gr}(d, x, y) D_{gr}(x, y)}{\sum_{(d,x,y) \in \xi_{gr}} w_{gr}(d, x, y)}$$

where  $\xi_{gr} = \{(\frac{-\pi}{4}, i-1, j-1), (\frac{\pi}{4}, i-1, j+1), (\frac{\pi}{4}, i+1, j-1), (\frac{-\pi}{4}, i+1, j+1)\}$ .

After performing Step 2, the current pattern of the G-R color difference plane is illustrated in Fig. 4.3(a). For easy exposition, the central pixel at position  $(i', j')$  in Fig. 4.3(b), which is obtained by shifting Fig. 4.3(a) one pixel down, is taken as the representative to explain the G-R color difference plane interpolation in Step

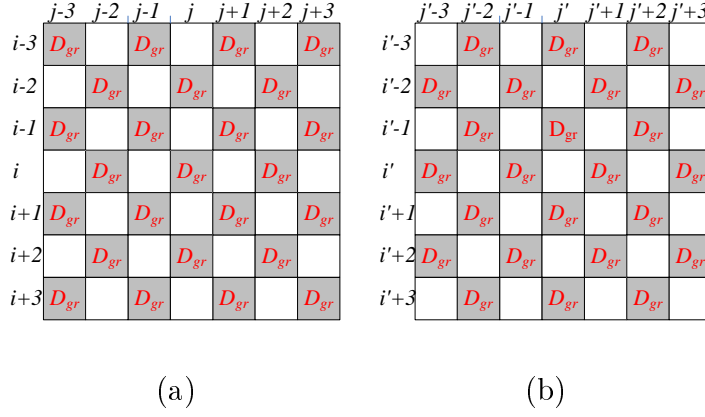


Fig. 4.3: Two patterns of the G-R color difference plane. (a) The pattern of the G-R color difference plane after performing Step 1. (b) The pattern shifting (a) one pixel down.

2. Referring to Fig. 4.3(b), it is not hard to find that the pattern of the G-R color difference plane at present is the same as that of the G plane in the mosaic image as shown in Fig. 1.1. Therefore, the interpolation estimation approach described in last subsection can be directly used to estimate the G-R color difference value at position  $(i', j')$ . Consequently, the G-R color difference value of the current pixel can be estimated by

$$D_{gr}(i', j') = \frac{\sum_{(d,x,y) \in \xi'_{gr}} w_{gr}(d, x, y) D_{gr}(x, y)}{\sum_{(d,x,y) \in \xi'_{gr}} w_{gr}(d, x, y)}$$

$$\xi'_{gr} = \begin{cases} \xi_1 & \text{if } HP'_V(i', j') < \alpha HP'_H(i', j') \\ \xi_2 & \text{if } HP'_H(i', j') < \alpha HP'_V(i', j') \\ \xi_1 \cup \xi_2 & \text{otherwise} \end{cases}$$

where  $\xi_1 = \{(V, i' \pm 1, j')\}$  and  $\xi_2 = \{(H, i', j' \pm 1)\}$ ;  $w_{gr}(H, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^H(x, y+k)}$  and  $w_{gr}(V, x, y) = \frac{1}{1 + \sum_{k=-1}^1 \delta_k \Delta I_{dm}^V(x+k, y)}$  where  $\delta_k = 3$  if  $k = 0$ ;  $\delta_k = 1$ , otherwise; the parameter  $\alpha$  is set to  $\alpha = 0.55$  empirically.

After constructing the fully populated G plane, G-R color difference plane, and G-B color difference plane, in next subsection, the three constructed planes will be

resized to the arbitrary-ratio sized ones by using the DCT approach, and then the arbitrary-ratio resized full color image is followed.

### 4.2.3 Stage 3: Resizing the fully populated G plane, G-R color difference plane, and G-B color difference plane

Based on the composite length DCT [59], this subsection presents the arbitrary-ratio resizing stage for constructing the fully populated G plane  $I_{dm}^g$ , G-R color difference plane  $D_{gr}$ , and G-B color difference plane  $D_{gb}$ . Since the resizing stage for  $I_{dm}^g$  is the same as that for  $D_{gr}$  and  $D_{gb}$ , we only present it for  $I_{dm}^g$ . Let  $DCT(B^{(a)})$  and  $IDCT(B^{(a)})$  be the DCT and inverse DCT on the  $a \times a$  block  $B^{(a)}$ . For the fully populated G plane  $I_{dm}^g$  with size  $M \times N$ , we first divide it into a set of the image blocks, each with size  $8 \times 8$ , and the obtained image block set is denoted by  $\Phi_{I_{dm}^g} = \{B_{m,n}^{(8)} | 0 \leq m \leq \frac{M}{8} - 1, 0 \leq n \leq \frac{N}{8} - 1\}$  where  $B_{m,n}^{(8)}$  denotes the  $(\frac{mN}{8} + n)$ -th block in  $\Phi_{I_{dm}^g}$ . Then, the DCT is performed on each  $8 \times 8$  image block to obtain the set of transformed blocks,  $\hat{\Phi}_{I_{dm}^g} = \{\hat{B}_{m,n}^{(8)} | 0 \leq m \leq \frac{M}{8} - 1, 0 \leq n \leq \frac{N}{8} - 1\}$ . If we want to resize the  $M \times N$  G plane  $I_{dm}^g$  to the one with size  $\frac{q}{p}M \times \frac{q}{p}N$ , the resizing ratio is said to  $\frac{q}{p}$ . According to the resizing ratio  $\frac{q}{p}$ , first  $p^2$  blocks in  $\hat{\Phi}_{I_{dm}^g}$  are collected to be an active unit which are surrounded by dashed lines in Fig. 4.4. In order to achieve resizing ratio  $\frac{q}{p}$ , the  $p^2$  blocks in each active unit should be increased or decreased to  $q^2$  blocks.

For the active unit  $\hat{\Psi}_{Act} = \{\hat{B}_{m,n}^{(8)} | 0 \leq m, n \leq p-1\}$ , the  $\frac{q}{p}$ -fold resizing procedure consists of the following four steps and the resizing procedure is depicted in Fig. 4.5.

**Step 1:** Each  $8 \times 8$  DCT coefficient block  $\hat{B}_{m,n}^{(8)}$  in the active unit is expanded to a

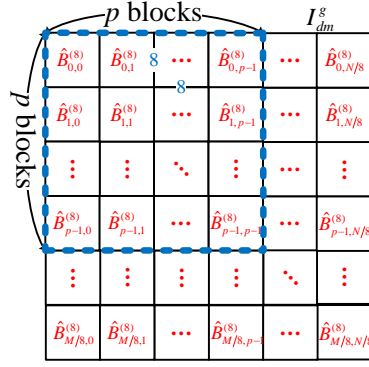


Fig. 4.4: An example of the active unit in  $\hat{\Phi}_{dm}^g$ .

$(8+z) \times (8+z)$  block  $\hat{B}_{m,n}^{(8+z)}$  by the following zero padding rule:

$$\hat{B}_{m,n}^{(8+z)}(x,y) = \begin{cases} \hat{B}_{m,n}^{(8)}(x,y) & \text{if } 0 \leq x, y < 8 \\ 0 & \text{otherwise} \end{cases}$$

$$\forall x, y \in \{0, 1, \dots, z-1\}; \forall m, n \in \{0, 1, \dots, p-1\}$$

where  $z$  denotes the smallest nonnegative integer satisfying the condition:

$p(8+z) = Cq$ ,  $C \geq 8$ . For example, assume  $\frac{q}{p} = \frac{4}{3}$ , then the smallest  $z$  is

4 due to  $3(8+4) = 9 \times 4$ . Consequently, we have the set of zero padded

DCT coefficient blocks  $\hat{\Psi}'_{Act} = \{\hat{B}_{m,n}^{(8+z)} | 0 \leq m, n \leq p-1\}$ .

**Step 2:** For each  $\hat{B}_{m,n}^{(8+z)}$  in  $\hat{\Psi}'_{Act}$ , the  $(8+z) \times (8+z)$  IDCT is performed on it to obtain the upsized image:

$$B_{m,n}^{(8+z)} = IDCT(\hat{B}_{m,n}^{(8+z)}), \quad 0 \leq m, n \leq p-1.$$

After performing the  $(8+z) \times (8+z)$ -sample IDCT's on all the zero

padded DCT coefficient blocks in  $\hat{\Psi}'_{Act}$ , the upsized subimage  $I_{up}$  ( $=$

$\bigcup_{0 \leq m, n \leq p-1} B_{m,n}^{(8+z)}$ ) is constructed.

**Step 3:** Then, the upsized subimage  $I_{up}$  is divided into  $q^2$  blocks, each block with size  $\frac{p}{q}(8+z) \times \frac{p}{q}(8+z)$ , and the set of the resampled image blocks is denoted



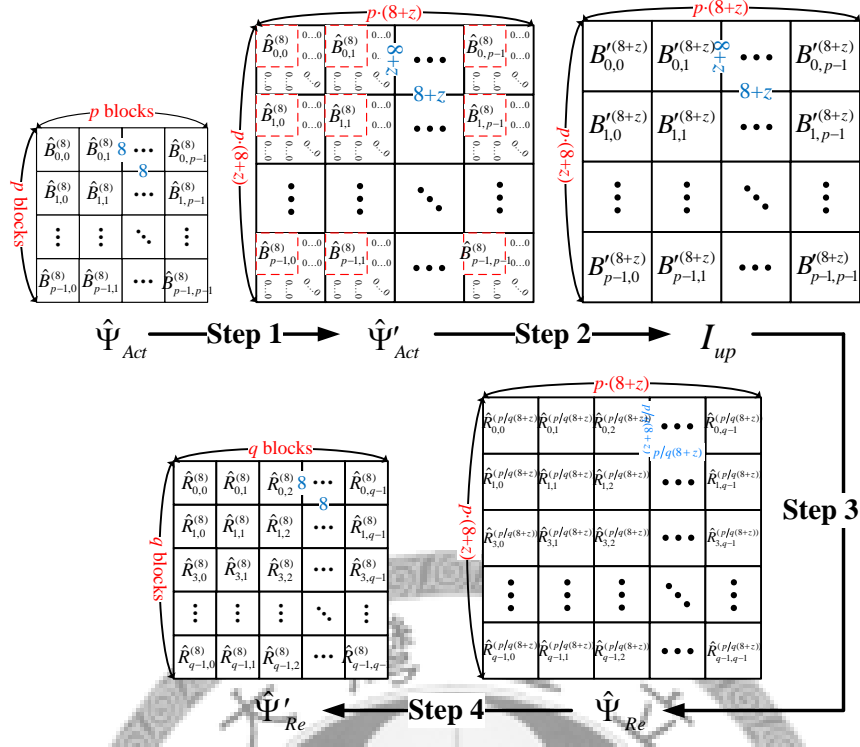


Fig. 4.5: The depiction of the  $\frac{q}{p}$ -fold resizing procedure.

by  $\Psi_{Re} = \{R_{m,n}^{(\frac{q}{p}(8+z))} \mid 0 \leq m, n \leq q-1\}$ . Next, the DCT is performed on each  $R_{m,n}^{(\frac{q}{p}(8+z))}$  in  $\Psi_{Re}$  to construct the resampled DCT coefficient block  $\hat{R}_{m,n}^{(\frac{q}{p}(8+z))}$

$$\hat{R}_{m,n}^{(\frac{q}{p}(8+z))} = DCT(R_{m,n}^{(\frac{q}{p}(8+z))}), \quad 0 \leq m, n \leq q-1.$$

We thus have the set of resampled DCT coefficient blocks  $\hat{\Psi}_{Re} = \{\hat{R}_{m,n}^{(\frac{q}{p}(8+z))} \mid 0 \leq m, n \leq q-1\}$ .

**Step 4:** Finally, the high-frequency DCT coefficients of each block in  $\hat{\Psi}_{Re}$  are truncated by the following rule:

$$\hat{R}'_{m,n}{}^{(8)}(x, y) = \hat{R}_{m,n}^{(\frac{q}{p}(8+z))}(x, y)$$

$$\forall x, y \in \{0, 1, \dots, 7\}; \quad \forall m, n \in \{0, 1, \dots, q-1\}$$

where  $\hat{R}'_{m,n}{}^{(8)}$  denotes the left-upper  $8 \times 8$  subblock of  $\hat{R}_{m,n}^{(\frac{q}{p}(8+z))}$ ;  $\hat{R}_{m,n}^{(\frac{q}{p}(8+z))}(x, y)$

and  $\hat{R}'_{m,n}{}^{(8)}(x, y)$  denote the DCT coefficients of the pixels at position  $(x, y)$  in  $\hat{R}'_{m,n}{}^{(\frac{q}{p}(8+z))}$  and  $\hat{R}'_{m,n}{}^{(8)}$ , respectively. Consequently, the resized active unit  $\hat{\Psi}'_{Re} = \{\hat{R}'_{m,n}{}^{(8)} | 0 \leq m, n \leq q - 1\}$  is constructed.

After performing the above resizing procedure on all the active units in  $I_{dm}^g$ , a set of  $8 \times 8$  DCT coefficient blocks  $\hat{\Phi}_{Z_{dm}^g} = \{\hat{B}_{m,n}{}^{(8)} | \forall 0 \leq m \leq \frac{\frac{q}{p}M}{8} - 1, 0 \leq n \leq \frac{\frac{q}{p}N}{8} - 1\}$  can be obtained. Consequently, the  $\frac{q}{p}M \times \frac{q}{p}N$  sized G plane  $Z_{dm}^g$  can be obtained by performing the IDCT on each  $8 \times 8$  DCT coefficient blocks in  $\hat{\Phi}_{Z_{dm}^g}$ . By the same argument, the  $\frac{q}{p}M \times \frac{q}{p}N$  sized G-R and G-B color difference planes, which are denoted by  $ZD_{gr}$  and  $ZD_{gb}$ , respectively, can be also obtained. Finally, the  $\frac{q}{p}M \times \frac{q}{p}N$  sized R and B planes can be constructed by

$$\begin{aligned} Z_{dm}^r(i_z, j_z) &= Z_{dm}^g(i_z, j_z) - ZD_{gr}(i_z, j_z) \\ Z_{dm}^b(i_z, j_z) &= Z_{dm}^g(i_z, j_z) - ZD_{gb}(i_z, j_z) \\ \forall i_z \in \{0, 1, \dots, \frac{q}{p}M - 1\}, \forall j_z \in \{0, 1, \dots, \frac{q}{p}N - 1\} \end{aligned}$$

where  $Z_{dm}^r(i_z, j_z)$ ,  $Z_{dm}^g(i_z, j_z)$ , and  $Z_{dm}^b(i_z, j_z)$  denote the three color components of the pixel at position  $(i_z, j_z)$  in the  $\frac{q}{p}M \times \frac{q}{p}N$  sized full color image  $Z_{dm}$ , respectively;  $ZD_{gr}(i_z, j_z)$  and  $ZD_{gb}(i_z, j_z)$  denote the G-R and G-B color difference value of the pixel at position  $(i_z, j_z)$  in  $ZD_{gr}$  and  $ZD_{gb}$ , respectively.

After presenting the proposed arbitrary-ratio resizing algorithm for mosaic images, in next section, some experimental results are demonstrated to show the quality advantage of the proposed algorithm.

### 4.3 Experimental results

In this section, based on twenty-four popular testing mosaic images, some experimental results are demonstrated to show the applicability and quality advantages of the proposed algorithm. Fig. 3.10 illustrates the twenty-four testing images from Kodak PhotoCD [85]. In our experiments, the twenty-four testing images, each with size  $512 \times 728$ , are first downsized to the  $\frac{p}{q} \cdot 512 \times \frac{p}{q} \cdot 728$  sized ones by using the resizing method proposed in [59], and then the downsized images are down-sampled to the mosaic images.

In order to evaluate the performance of the proposed joint demosaicing and resizing algorithm, three native resizing algorithms for mosaic images are adopted to compare with the proposed algorithm. In the three native resizing algorithms, one of the three demosaicing methods proposed in [62], [48], and [13], respectively, is first utilized to obtain the demosaiced images and then the resizing method proposed in [59] is applied to construct the resized full color images. In addition, the postprocessing approach proposed in [46] is adopted to the three concerned demosaicing methods to enhance the demosaiced image quality. For convenience, the three native algorithms based on the demosaicing methods proposed in [62], [48], and [13] are called  $A_1$ ,  $A_2$ , and  $A_3$ , respectively. The concerned algorithms are implemented on the IBM compatible computer with Intel Core 2 Duo CPU 1.83GHz and 2GB RAM. The operating system used is MS-Windows XP and the program developing environment is Borland C++ Builder 6.0. Further, the concerned four programs and results are available in [87].

Here, we adopt two objective color image quality measures, the CPSNR and the S-CIELAB  $\Delta E_{ab}^*$  [33, 41], and one subjective color image quality measure, the color

artifacts, to justify the the quality advantage of of the proposed algorithm. The CPSNR for a color image with size  $M \times N$  is defined by

$$\text{CPSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{3MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{c \in C} [I_{ori}^c(m, n) - Z_{dm}^c(m, n)]^2}, \quad C = \{r, g, b\}$$

where  $I_{ori}^r(m, n)$ ,  $I_{ori}^g(m, n)$ , and  $I_{ori}^b(m, n)$  denote the three color components of the color pixel at position  $(m, n)$  in the original full color image;  $Z_{dm}^r(m, n)$ ,  $Z_{dm}^g(m, n)$ , and  $Z_{dm}^b(m, n)$  denote the three color components of the color pixel at position  $(m, n)$  in the resized full color image. The greater the CPSNR is, the better the image quality is. The S-CIELAB  $\Delta E_{ab}^*$  of a color image with size  $M \times N$  is defined by

$$\Delta E_{ab}^* = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left\{ \sqrt{\sum_{c \in \Gamma} [EI_{ori}^c(m, n) - EZ_{dm}^c(m, n)]^2} \right\}, \quad \Gamma = \{L, a, b\}$$

where  $EI_{ori}^L(m, n)$ ,  $EI_{ori}^a(m, n)$ , and  $EI_{ori}^b(m, n)$  denote the three CIELAB color components of the color pixel at position  $(m, n)$  in the original full color image;  $EZ_{dm}^L(m, n)$ ,  $EZ_{dm}^a(m, n)$ , and  $EZ_{dm}^b(m, n)$  denote the three CIELAB color components of the color pixel at position  $(m, n)$  in the resized full color image. The smaller the S-CIELAB  $\Delta E_{ab}^*$  is, the better the image quality is.

Based on twenty-four testing images, among the four concerned resizing algorithms, Table 4.1 and Table 4.2 demonstrate the image quality comparison in terms of the average CPSNR and the average S-CIELAB  $\Delta E_{ab}^*$ , respectively. In Table 4.1 and Table 4.2, the entries with the largest CPSNR and the smallest S-CIELAB  $\Delta E_{ab}^*$  are highlighted by boldface. From Table 4.1 and Table 4.2, it is observed that the proposed resizing algorithm produces the best image quality in terms of CPSNR and S-CIELAB  $\Delta E_{ab}^*$  among the four algorithms.

Next, the subjective image visual measure, color artifacts, is adopted to demonstrate the visual quality advantage of the proposed resizing algorithm. Some color

Table 4.1: Average CPSNR comparison for the four concerned algorithms.

Algorithm	Resizing ratio $\frac{q}{p}$				Average
	2	$\frac{8}{5}$	$\frac{4}{3}$	$\frac{8}{7}$	
$A_1$	28.4051	30.2171	32.1575	34.1260	31.2264
$A_2$	28.4107	30.1089	31.8667	33.6307	31.0043
$A_3$	28.6069	30.5344	32.5719	34.6149	31.5820
Ours	<b>29.1714</b>	<b>31.3134</b>	<b>33.6423</b>	<b>35.8527</b>	<b>32.4950</b>

Table 4.2: Average S-CIELAB  $\Delta E_{ab}^*$  comparison for the four concerned algorithms.

Algorithm	Resizing ratio $\frac{q}{p}$				Average
	2	$\frac{8}{5}$	$\frac{4}{3}$	$\frac{8}{7}$	
$A_1$	3.52927	3.05053	2.62441	2.17526	2.84487
$A_2$	3.50429	3.09378	2.74100	2.35818	2.92431
$A_3$	3.41551	2.92146	2.50336	2.07863	2.72974
Ours	<b>3.17182</b>	<b>2.70866</b>	<b>2.32772</b>	<b>2.00323</b>	<b>2.55286</b>



Fig. 4.6: The magnified subimage cut from the original testing image No. 8.

artifacts may appear on nonsmooth regions of the full color image after demosaicing and resizing the mosaic image. In order to examine the color artifacts among the concerned algorithms, the magnified subimages containing nonsmooth contents in the resized full color images are cut off. First, seventeen magnified subimages cut from the testing image No. 8 are adopted to examine the visual effect among the four concerned algorithms. Fig. 4.6 illustrates the magnified subimage cut from the original testing image No. 8. Based on different resizing ratios, Figs. 4.7–4.10 illustrate magnified subimages cut from the resized images obtained by the four concerned resized algorithms. From the visual comparison, it is observed that based on the same resizing ratio, the proposed algorithm produces less color artifacts when compared with the other three algorithms. We take the magnified subimages cut from the testing image No. 23 for the visual comparison and Figs. 4.11–4.15 illustrate the concerned magnified subimages. Similar to the color artifact examination for image No. 19, it is observed that the proposed resizing algorithm produces the least color artifacts, i.e. the best visual effect. More visual results of the concerned algorithms are available in [87].

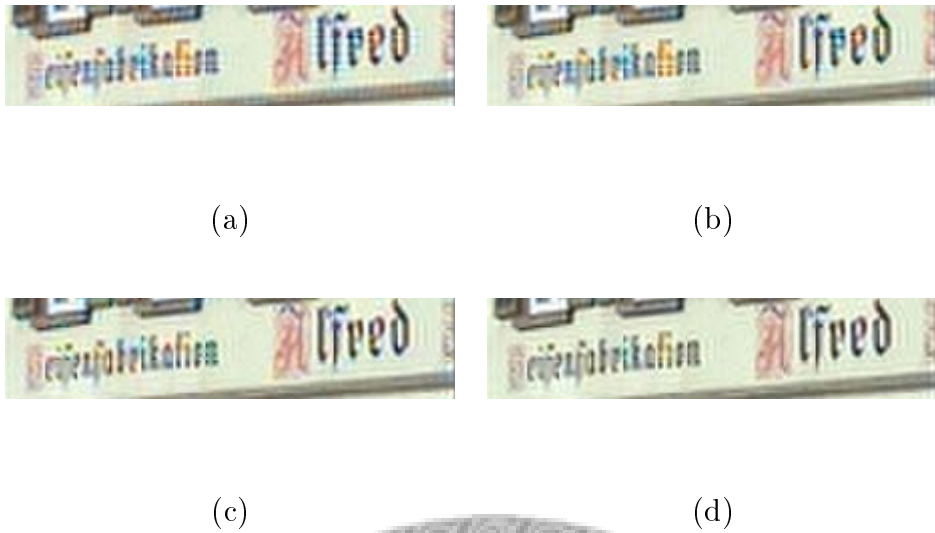


Fig. 4.7: For image No. 8, when the resizing ratio  $\frac{q}{p}$  is 2, four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

#### 4.4 Summary

In this chapter, a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images has been presented. By using the adaptive heterogeneity projection masks and SL-based masks, more accurate edge information is extracted first. Next, the fully populated green color plane is constructed by using the edge-sensing approach and color difference idea. Instead of interpolating the R and B color planes directly, the green-red color difference plane and green-blue color difference plane are therefore interpolated in order to reduce the estimation error. Next, based on the DCT technique, the above three constructed planes are resized to the arbitrary sized ones. Finally, the resized red and blue color planes are constructed by using the three resized planes, and then the arbitrary sized full color image is obtained. To the best of our knowledge, this is the first time that such a joint demosaicing and

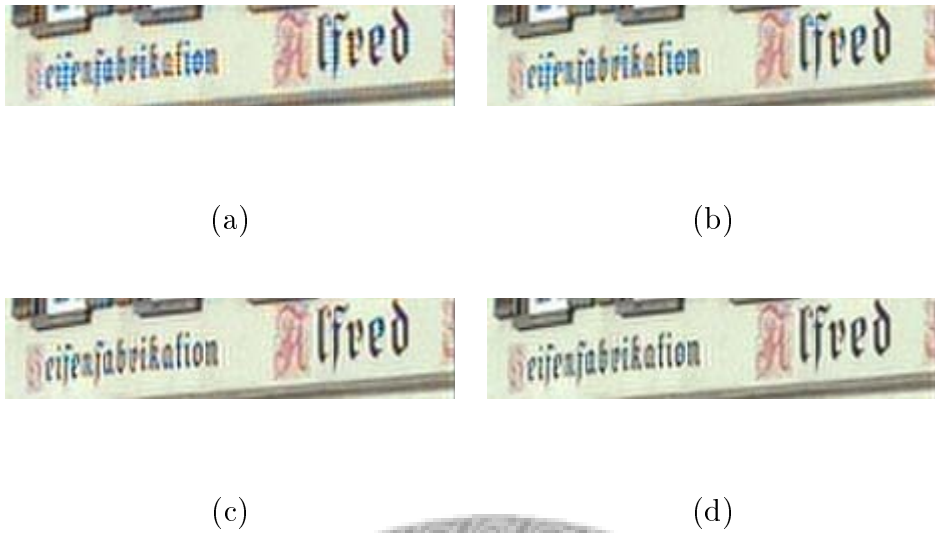


Fig. 4.8: For image No. 8, when the resizing ratio  $\frac{q}{p}$  is  $\frac{8}{5}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

arbitrary-ratio resizing algorithm for mosaic images is presented. Based on twenty-four popular testing mosaic images, the proposed new resizing algorithm has better image quality performance when compared with three native algorithms which are the combinations of three well-know demosaicing methods and one existing resizing method.



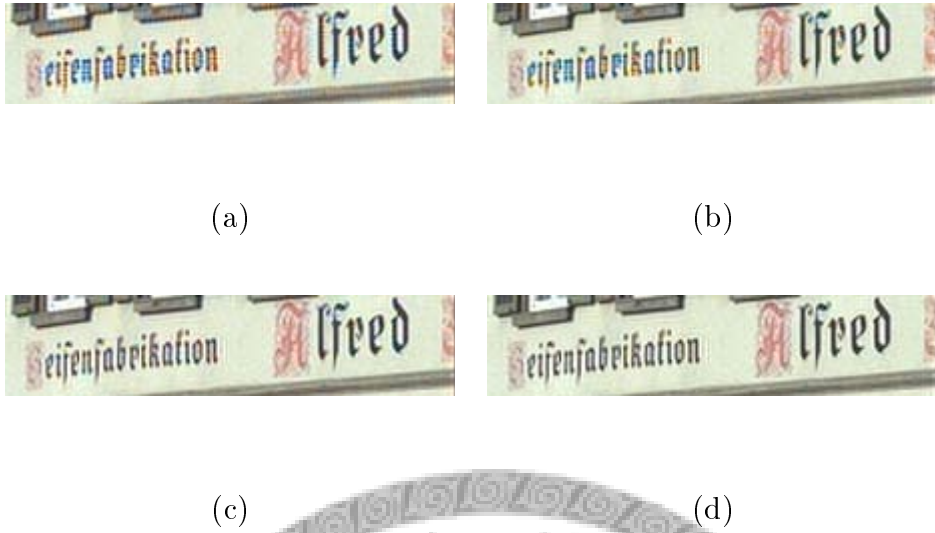


Fig. 4.9: For image No. 8, when the resizing ratio  $\frac{q}{p}$  is  $\frac{4}{3}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

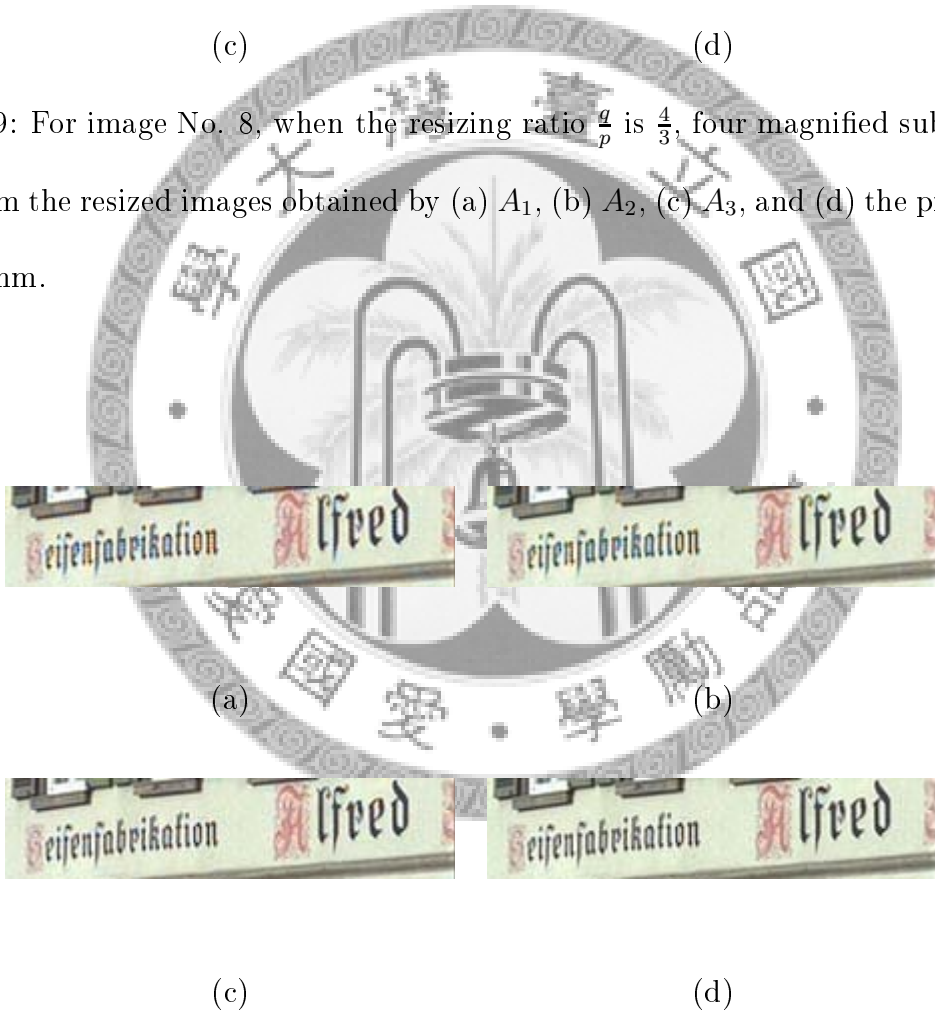
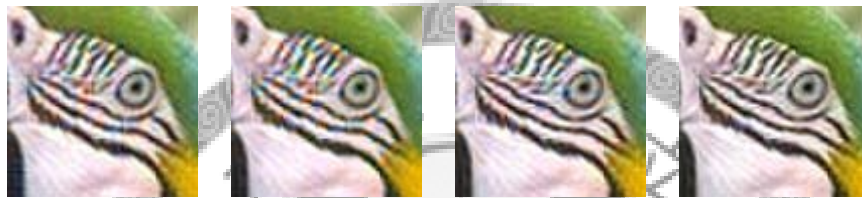


Fig. 4.10: For image No. 8, when the resizing ratio  $\frac{q}{p}$  is  $\frac{8}{7}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

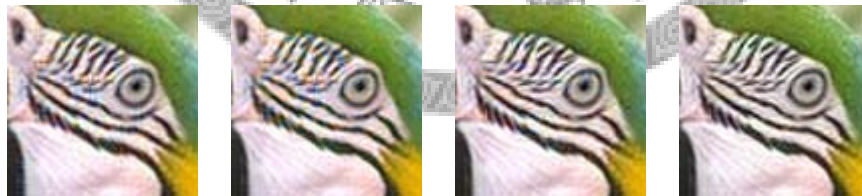


Fig. 4.11: The magnified subimage cut from the original testing image No. 23.



(a) (b) (c) (d)

Fig. 4.12: For image No. 23, when the resizing ratio  $\frac{q}{p}$  is 2, four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.



(a) (b) (c) (d)

Fig. 4.13: For image No. 23, when the resizing ratio  $\frac{q}{p}$  is  $\frac{8}{5}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.



Fig. 4.14: For image No. 23, when the resizing ratio  $\frac{q}{p}$  is  $\frac{4}{3}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

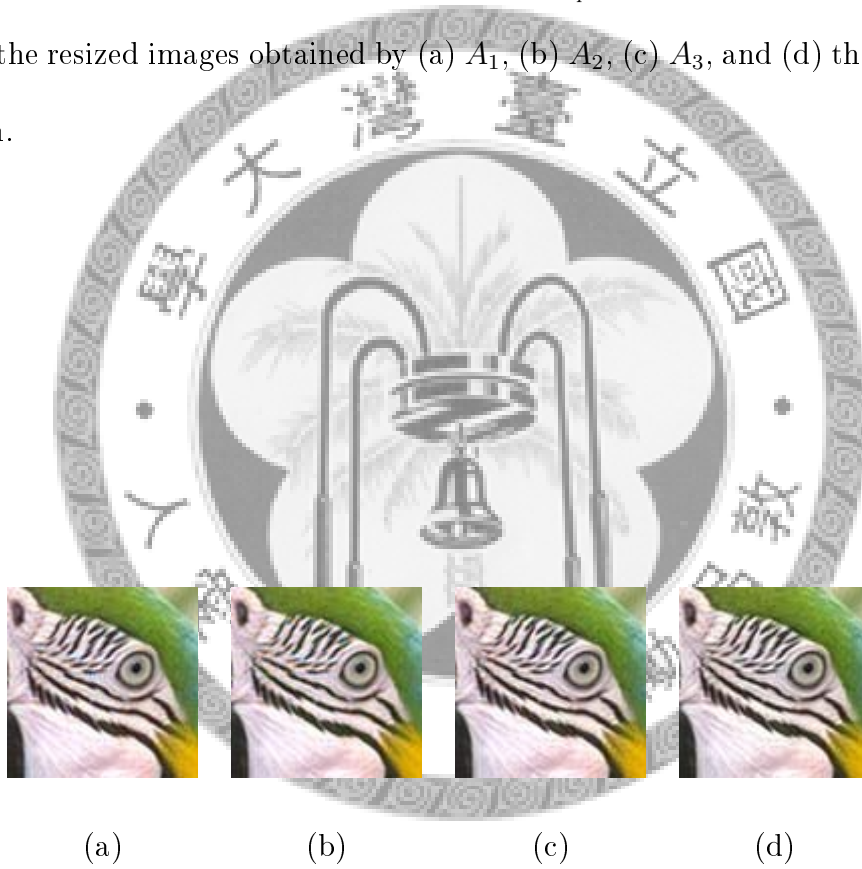


Fig. 4.15: For image No. 23, when the resizing ratio  $\frac{q}{p}$  is  $\frac{8}{7}$ , four magnified subimages cut from the resized images obtained by (a)  $A_1$ , (b)  $A_2$ , (c)  $A_3$ , and (d) the proposed algorithm.

# Chapter 5

## Conclusions

In this thesis, we present an efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation and a demosaicing algorithm for color filter array using gradient edge detection masks and adaptive heterogeneity-projection.

In the edge-preserving for color contrast enhancement issue, 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.

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 thesis to the field of color image retrieval issue [10, 31, 32, 61] when the relevant retrieval techniques include the consideration of edge information.

In the color demosaicing issue, without demosaicing processing, we first proposes a new approach to extract more accurate gradient/edge information on mosaic images directly. Next, based on spectral-spatial correlation [78], a novel adaptive heterogeneity-projection with proper mask size for each pixel is presented. Combining the extracted gradient/edge information and the adaptive heterogeneity-projection values, a new edge-sensing demosaicing algorithm is presented. Some experimental results have been carried out to demonstrate the quality advantage in terms of CPSNR and S-CIELAB  $\Delta E_{ab}^*$  of our proposed new demosaicing algorithm when compared with several recently published algorithms.

Besides combining the bilinear interpolation and the edge detection, it is an interesting research issue to replace the bilinear solution with another more accurate method, e.g. nonlinear solution, to have better gain of both edge-detection and demosaicing performance.

In mosaic image resizing issue, a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images has been presented. Based on the color difference concept and the composite length DCT, the mosaic image can be demosaiced and resized to an arbitrary-ratio sized full color image. To the best of our knowledge, this is the first time that such a joint demosaicing and arbitrary-ratio resizing algorithm for mosaic images is presented. Based on twenty-four popular testing mosaic images,

the proposed new resizing algorithm has better image quality performance when compared with three native algorithms which are the combinations of three well-know demosaicing methods and one existing resizing method.



# Appendix A

## Derivation of the SL-based

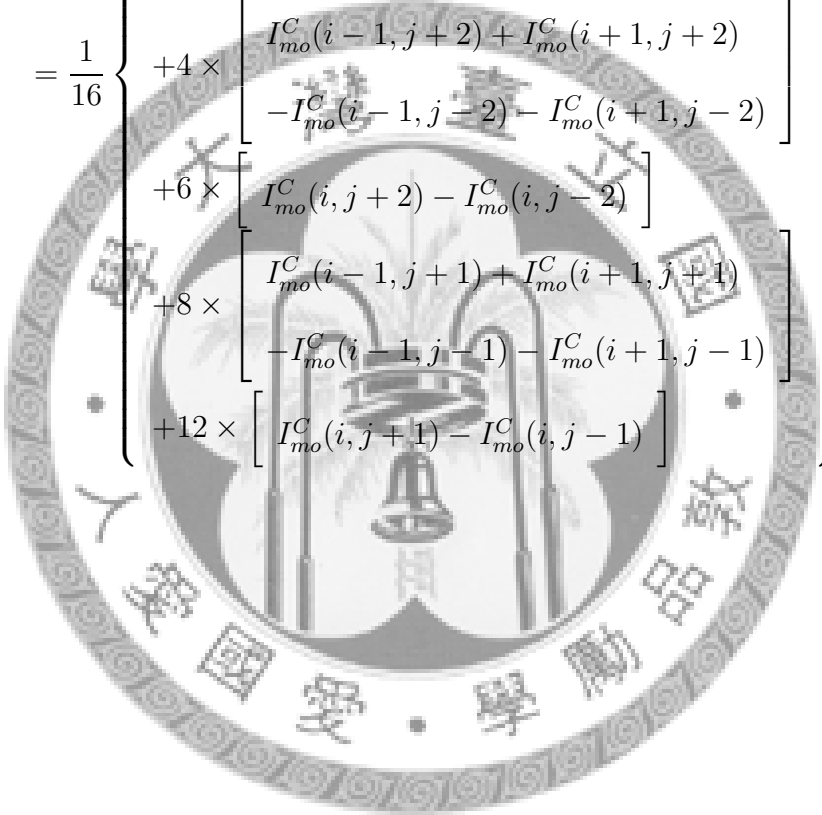
### quad-mask

Combining Eq. (3.2) and Eq. (3.3), the four masks used to extract more accurate color gradient information on mosaic images directly can be obtained by the



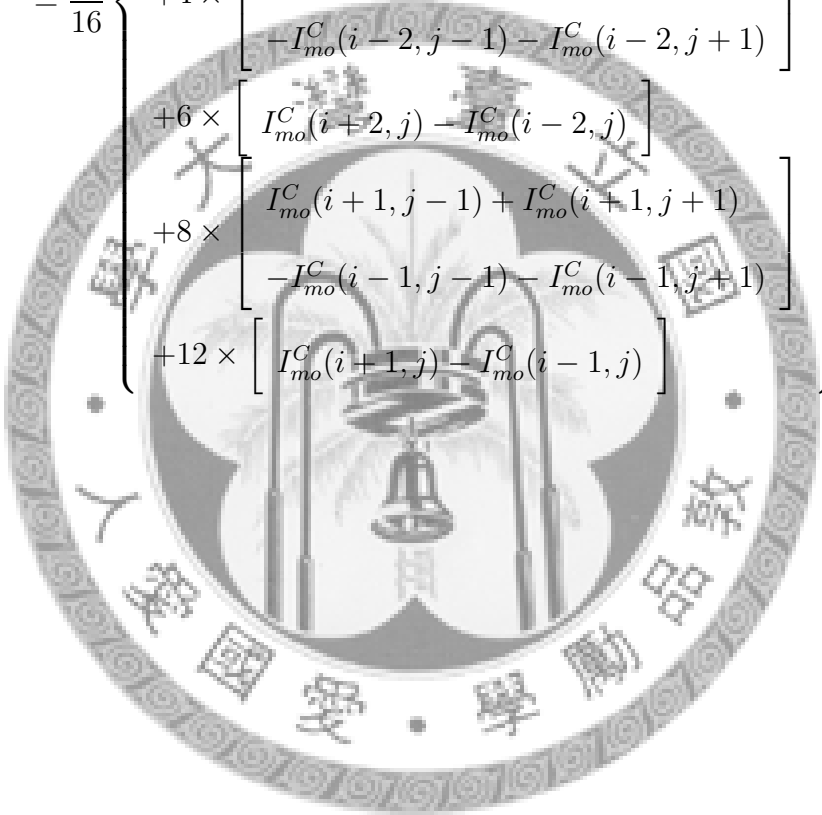
following derivation:

$$\Delta I_{dm}^H(i, j) = \left\{ \begin{array}{l} [L(i-1, j+1) + 2L(i, j+1) + L(i+1, j+1)] \\ -[L(i-1, j-1) + 2L(i, j-1) + L(i+1, j-1)] \end{array} \right\}$$

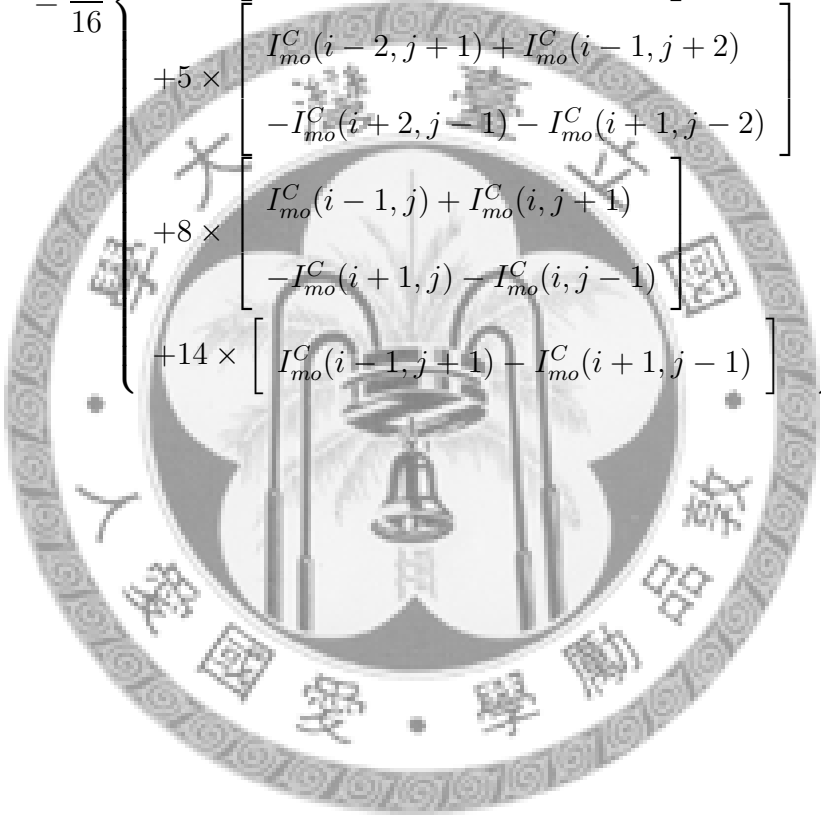
$$= \frac{1}{16} \left\{ \begin{array}{l} \left[ \begin{array}{l} I_{mo}^C(i-2, j+2) + I_{mo}^C(i+2, j+2) \\ -I_{mo}^C(i-2, j-2) - I_{mo}^C(i+2, j-2) \end{array} \right] \\ +2 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j+1) + I_{mo}^C(i+2, j+1) \\ -I_{mo}^C(i-2, j-1) - I_{mo}^C(i+2, j-1) \end{array} \right] \\ +4 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j+2) + I_{mo}^C(i+1, j+2) \\ -I_{mo}^C(i-1, j-2) - I_{mo}^C(i+1, j-2) \end{array} \right] \\ +6 \times \left[ \begin{array}{l} I_{mo}^C(i, j+2) - I_{mo}^C(i, j-2) \end{array} \right] \\ +8 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j+1) + I_{mo}^C(i+1, j+1) \\ -I_{mo}^C(i-1, j-1) - I_{mo}^C(i+1, j-1) \end{array} \right] \\ +12 \times \left[ \begin{array}{l} I_{mo}^C(i, j+1) - I_{mo}^C(i, j-1) \end{array} \right] \end{array} \right\}$$




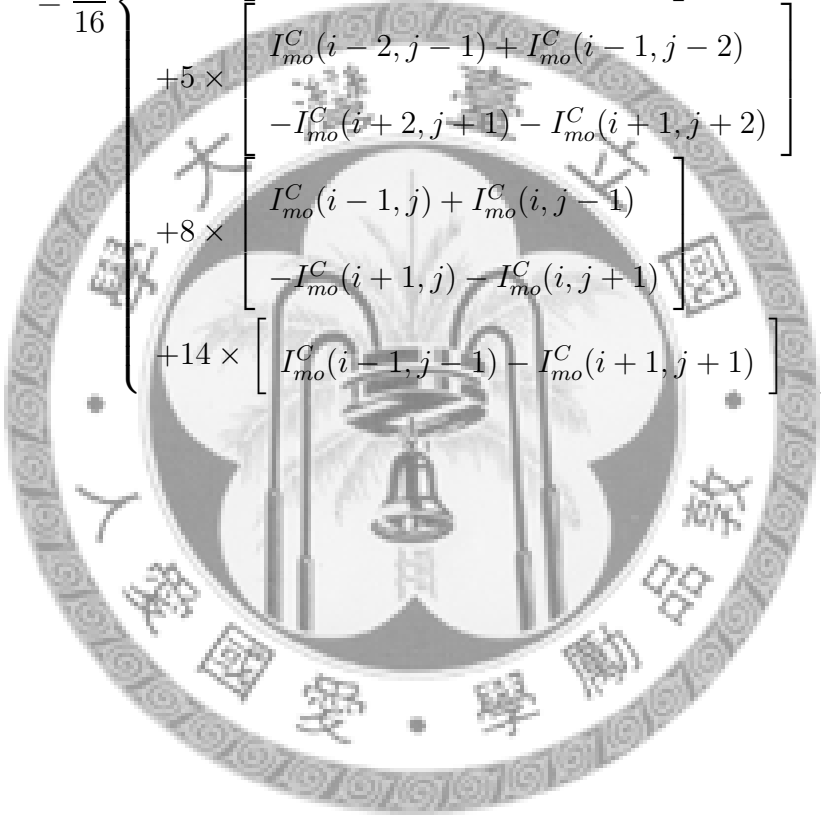
$$\Delta I_{dm}^V(i, j) = \left\{ \begin{array}{l} [L(i+1, j-1) + 2L(i+1, j) + L(i+1, j+1)] \\ -[L(i-1, j-1) + 2L(i-1, j) + L(i-1, j+1)] \end{array} \right\}$$

$$= \frac{1}{16} \left\{ \begin{array}{l} \left[ \begin{array}{l} I_{mo}^C(i+2, j-2) + I_{mo}^C(i+2, j+2) \\ -I_{mo}^C(i-2, j-2) - I_{mo}^C(i-2, j+2) \end{array} \right] \\ +2 \times \left[ \begin{array}{l} I_{mo}^C(i+1, j-2) + I_{mo}^C(i+1, j+2) \\ -I_{mo}^C(i-1, j-2) - I_{mo}^C(i-1, j+2) \end{array} \right] \\ +4 \times \left[ \begin{array}{l} I_{mo}^C(i+2, j-1) + I_{mo}^C(i+2, j+1) \\ -I_{mo}^C(i-2, j-1) - I_{mo}^C(i-2, j+1) \end{array} \right] \\ +6 \times \left[ \begin{array}{l} I_{mo}^C(i+2, j) - I_{mo}^C(i-2, j) \end{array} \right] \\ +8 \times \left[ \begin{array}{l} I_{mo}^C(i+1, j-1) + I_{mo}^C(i+1, j+1) \\ -I_{mo}^C(i-1, j-1) - I_{mo}^C(i-1, j+1) \end{array} \right] \\ +12 \times \left[ \begin{array}{l} I_{mo}^C(i+1, j) - I_{mo}^C(i-1, j) \end{array} \right] \end{array} \right\}$$


$$\Delta I_{dm}^{\frac{\pi}{4}}(i, j) = \left\{ \begin{array}{l} [L(i-1, j) + 2L(i-1, j+1) + L(i, j+1)] \\ -[L(i+1, j) + 2L(i+1, j-1) + L(i, j-1)] \end{array} \right\}$$

$$= \frac{1}{16} \left\{ \begin{array}{l} \left[ \begin{array}{l} I_{mo}^C(i-2, j-1) + I_{mo}^C(i+1, j+2) \\ -I_{mo}^C(i-1, j-2) - I_{mo}^C(i+2, j+1) \end{array} \right] \\ +2 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j+2) - I_{mo}^C(i+2, j-2) \end{array} \right] \\ +4 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j) + I_{mo}^C(i, j+2) \\ -I_{mo}^C(i+2, j) - I_{mo}^C(i, j-2) \end{array} \right] \\ +5 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j+1) + I_{mo}^C(i-1, j+2) \\ -I_{mo}^C(i+2, j-1) - I_{mo}^C(i+1, j-2) \end{array} \right] \\ +8 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j) + I_{mo}^C(i, j+1) \\ -I_{mo}^C(i+1, j) - I_{mo}^C(i, j-1) \end{array} \right] \\ +14 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j+1) - I_{mo}^C(i+1, j-1) \end{array} \right] \end{array} \right\}$$


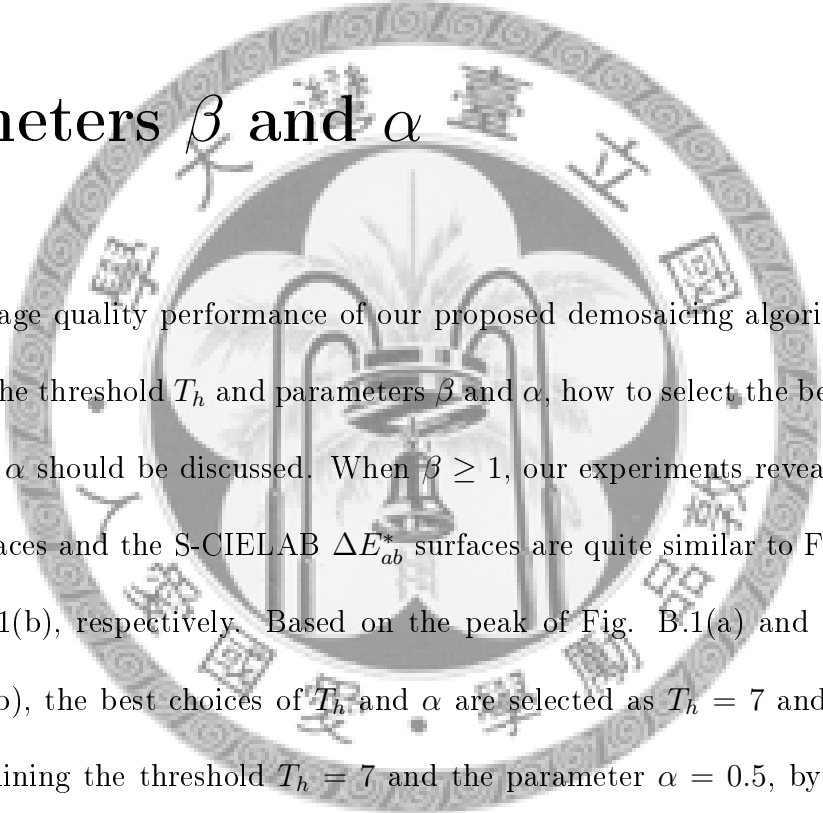
$$\Delta I_{dm}^{\frac{-\pi}{4}}(i, j) = \left\{ \begin{array}{l} [L(i-1, j) + 2L(i-1, j-1) + L(i, j-1)] \\ -[L(i+1, j) + 2L(i+1, j+1) + L(i, j+1)] \end{array} \right\}$$

$$= \frac{1}{16} \left\{ \begin{array}{l} \left[ \begin{array}{l} I_{mo}^C(i-2, j+1) + I_{mo}^C(i+1, j-2) \\ -I_{mo}^C(i-1, j+2) - I_{mo}^C(i+2, j-1) \end{array} \right] \\ +2 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j-2) - I_{mo}^C(i+2, j+2) \end{array} \right] \\ +4 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j) + I_{mo}^C(i, j-2) \\ -I_{mo}^C(i+2, j) - I_{mo}^C(i, j+2) \end{array} \right] \\ +5 \times \left[ \begin{array}{l} I_{mo}^C(i-2, j-1) + I_{mo}^C(i-1, j-2) \\ -I_{mo}^C(i+2, j+1) - I_{mo}^C(i+1, j+2) \end{array} \right] \\ +8 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j) + I_{mo}^C(i, j-1) \\ -I_{mo}^C(i+1, j) - I_{mo}^C(i, j+1) \end{array} \right] \\ +14 \times \left[ \begin{array}{l} I_{mo}^C(i-1, j-1) - I_{mo}^C(i+1, j+1) \end{array} \right] \end{array} \right\}$$


## Appendix B

### Determination of threshold $T_h$ and

### parameters $\beta$ and $\alpha$



Since the image quality performance of our proposed demosaicing algorithm is influenced by the threshold  $T_h$  and parameters  $\beta$  and  $\alpha$ , how to select the best choices of  $T_h$ ,  $\beta$  and  $\alpha$  should be discussed. When  $\beta \geq 1$ , our experiments reveal that the CPSNR surfaces and the S-CIELAB  $\Delta E_{ab}^*$  surfaces are quite similar to Fig. B.1(a) and Fig. B.1(b), respectively. Based on the peak of Fig. B.1(a) and the valley of Fig. B.1(b), the best choices of  $T_h$  and  $\alpha$  are selected as  $T_h = 7$  and  $\alpha = 0.5$ . After determining the threshold  $T_h = 7$  and the parameter  $\alpha = 0.5$ , by the same argument, the best choice of the parameter  $\beta$  is selected as  $\beta = 1$  based on the peak of Fig. B.2(a) and the valley of Fig. B.2(b).

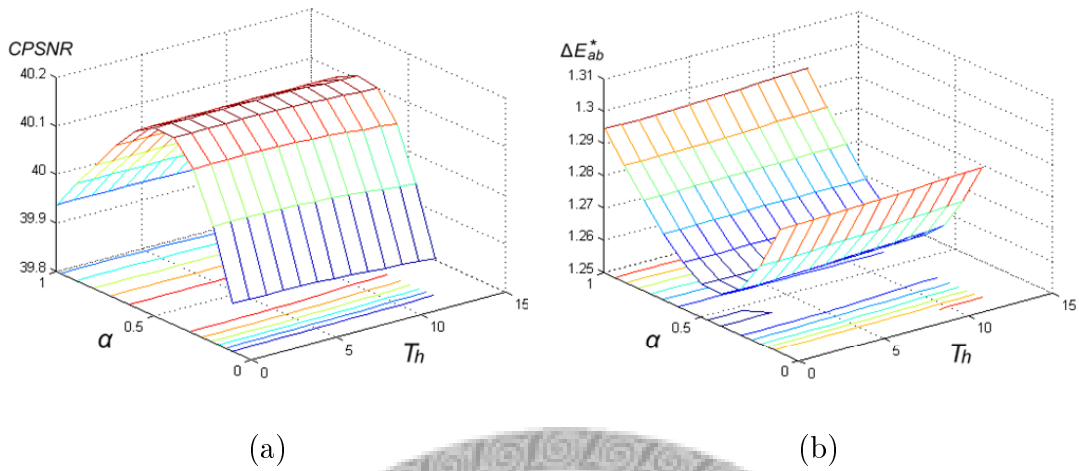


Fig. B.1: Two demosaiced image quality surfaces in terms of  $T_h$  and  $\alpha$  for  $\beta \geq 1$ .

(a) The CPSNR surface. (b) The S-CIELAB  $\Delta E_{ab}^*$  surface.

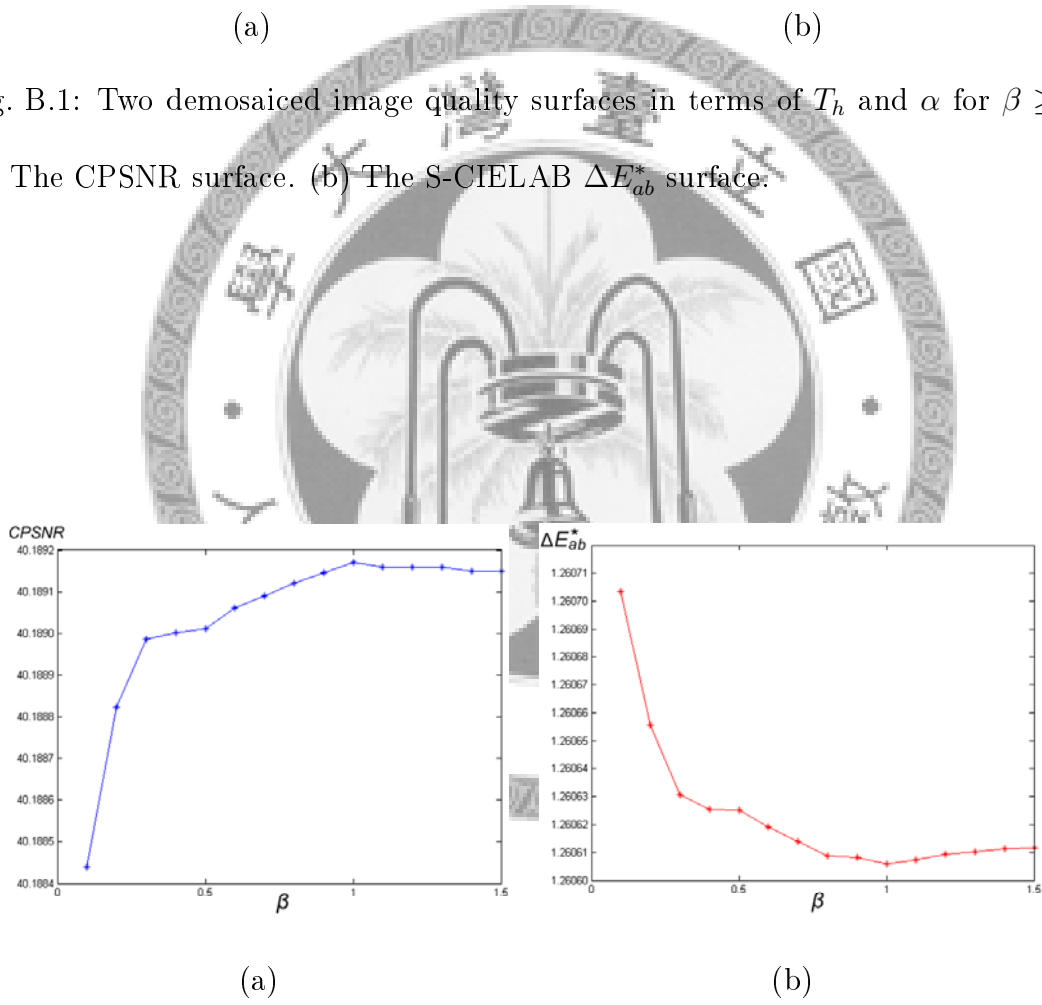
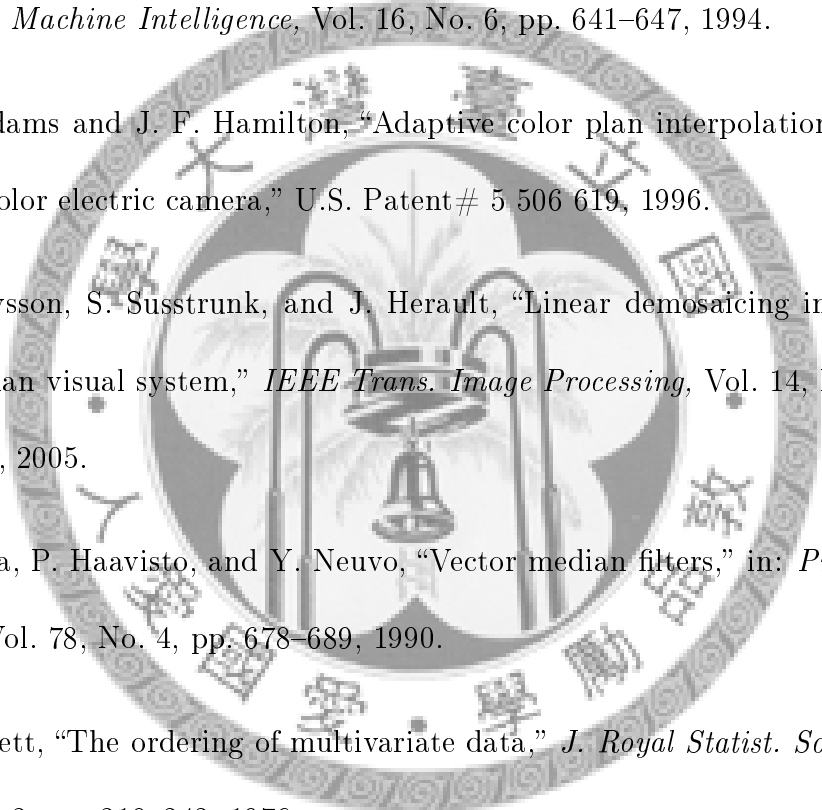


Fig. B.2: Two demosaiced image quality curves in terms of  $\beta$  for  $T_h = 7$  and  $\alpha = 0.5$ .

(a) The CPSNR curve. (b) The S-CIELAB  $\Delta E_{ab}^*$  curve.

# Bibliography

- 
- [1] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, No. 6, pp. 641–647, 1994.
- [2] J. E. Adams and J. F. Hamilton, "Adaptive color plan interpolation in single sensor color electric camera," U.S. Patent# 5 506 619, 1996.
- [3] D. Alleysson, S. Susstrunk, and J. Herault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Processing*, Vol. 14, No. 4, pp. 439–449, 2005.
- [4] J. Astola, P. Haavisto, and Y. Neuvo, "Vector median filters," in: *Proc. of the IEEE*, Vol. 78, No. 4, pp. 678–689, 1990.
- [5] V. Barnett, "The ordering of multivariate data," *J. Royal Statist. Soc. A*, Vol. 139, No. 3, pp. 318–343, 1976.
- [6] S. Battiato, G. Gallo, and F. Stanco, "A locally adaptive zooming algorithm for digital images," *Image and Vision Computing*, Vol. 20, No. 11, pp. 805–812, 2002.
- [7] B. E. Bayer, "Color imaging array," U.S. Patent# 3 971 065, 1976.

- [8] E. J. Candes and 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, pp. 1–16, 1999.
- [9] H. A. Chang and H. H. Chen, "Stochastic color interpolation for digital cameras," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 17, No. 8, pp. 964–973, 2007.
- [10] D. Y. Chen, H. Y. M. Liao, and S. Y. Lee, "Robust video sequence retrieval using a novel object-based T2D-histogram descriptor," *J. of Visual Communication and Image Representation*, Vol. 16, No. 2, pp. 212–232, 2005.
- [11] J. Chen, T. N. Pappas, A. Mojsilovic, and B. E. Rogowitz, "Adaptive perceptual color-texture image segmentation," *IEEE Trans. Image Processing*, Vol. 14, No. 10, pp. 1524–1536, 2005.
- [12] S. C. Cheng and T. L. Wu, "Subpixel edge detection of color images by principal axis analysis and moment-preserving principle," *Pattern Recognition*, Vol. 38, No. 4, pp. 527–537, 2005.
- [13] K. H. Chung and Y. H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Processing*, Vol. 15, No. 10, pp. 2944–2955, 2006.
- [14] K. H. Chung and Y. H. Chan, "A low-complexity joint color demosaicking and zooming algorithm for digital camera," *IEEE Trans. Image Processing*, Vol. 16, No. 7, pp. 1705–1715, 2007.

- [15] K. L. Chung, W. J. Yang, W. M. Yan, and C. C. Wang, "Novel high-quality demosaicing algorithm for color filter array," in: *Proc. of 2007 Computer Vision, Graphics and Image Processing Conf.*, pp. 509–516, 2007.
- [16] K. L. Chung, W. J. Yang, and W. M. Yan, "Efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation," *J. of Visual Communication and Image Representation*, Vol. 19, No. 5, pp. 299–310, 2008.
- [17] K. L. Chung, W. J. Yang, W. M. Yan, and C. C. Wang, "Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection," *IEEE Trans. Image Processing*, Vol. 17, No. 12, pp. 2356–2367, 2008.
- [18] K. L. Chung, W. J. Yang, P. Y. Chen, W. M. Yan, and C. S. Fuh, "New joint demosaicing and zooming algorithm for color filter array," *IEEE Trans. Consumer Electronics*, Vol. 55, No. 3, pp. 1477–1486, 2009.
- [19] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent# 4 642 678, 1987.
- [20] E. Dubois, "Frequency-domain methods for demosaicking of bayer-sampled color images," *IEEE Signal Processing Letters*, Vol. 12, No. 12, pp. 847–850, 2005.
- [21] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Trans. Image Processing*, Vol. 10, No. 10, pp. 1454–1466, 2001.



- [22] J. Fan, W. G. Aref, M. S. Hacid, and A. K. Elmagarmid “An Improved automatic isotropic color edge detection technique,” *Pattern Recognition Letter*, Vol. 22, No. 13, pp. 1419–1429, 2001.
- [23] W. T. Freeman, “Median filter for reconstructing missing color samples,” U.S. Patent# 4 724 395, 1988.
- [24] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, New York, 1992.
- [25] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd Ed., Prentice Hall, New York, 2002.
- [26] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Trans. Image Processing*, Vol. 11, No. 9, pp. 997–1013, 2002.
- [27] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: color filter array interpolation,” *IEEE Signal Processing Magazine*, Vol. 22, No. 1, pp. 44–54, 2005.
- [28] Y. Hel-Or and D. Keren, “Image demosaicing method utilizing directional smoothing,” U.S. Patent# 6 404 918, 2002.
- [29] H. Hibbard, “Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,” U.S. Patent# 5 382 976, 1995.
- [30] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” *IEEE Trans. Image Processing*, Vol. 14, No. 3, pp. 360–369, 2005.

- [31] I. S. Hsieh and K. C. Fan, "Multiple classifier for color image retrieval," *IEEE Trans. Image Processing* Vol. 10, No. 6, pp. 938–950, 2001.
- [32] J. W. Hsieh and W. E. L. Grimson, "Spatial template extraction for image retrieval by region matching," *IEEE Trans. Image Processing*, Vol. 12, No. 11, pp. 1404–1415, 2003.
- [33] R. W. G. Hunt, *Measuring Colour*, 2nd Ed., Ellis Horwood, New York, 1995.
- [34] D. Keren and M. Osadchy, "Restoring subsampled color images," *Machine Vision and Applications*, Vol. 11, No. 4, pp. 197–202, 1999.
- [35] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Trans. Image Processing*, Vol. 8, No. 9, pp. 1221–1228, 1999.
- [36] J. Z. C. Lai and Y. C. Liaw, "Improvement of interpolated color filter array image using modified mean-removed vector quantization," *Pattern Recognition Letters*, Vol. 26, No. 8, pp. 1047–1058, 2005.
- [37] C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," US Patent# 5 373 322, 1994.
- [38] W. Lee, S. Lee, and J. Kim, "Cost-effective color filter array demosaicing using spatial correlation," *IEEE Trans. Consumer Electronics*, Vol. 52, No. 2, pp. 547–554, 2006.
- [39] O. Lezoray and H. Cardot, "Cooperation of color pixel classification schemes and color watershed: a study for microscopic images," *IEEE Trans. Image Processing*, Vol. 11, No. 7, pp. 783–789, 2002.

- [40] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Processing*, Vol. 14, No. 3, pp. 370–379, 2005.
- [41] W. Lu and Y. P. Tan, "Color filter array demosaicking: new method and performance measures," *IEEE Trans. Image Processing*, Vol. 12, No. 10, pp. 1194–1210, 2003.
- [42] L. Lucchese and S. K. Mitra, "Filtering color images in the xyY color space," in: *Proc. of Int. Conf. Image Processing*, Vol. 1, pp. 500–503, 2000.
- [43] L. Lucchese, S. K. Mitra, and J. Mukherjee, "A new algorithm based on saturation and desaturation in the xy chromaticity diagram for enhancement and re-rendition of color images," in: *Proc. of Int. Conf. Image Processing*, Vol. 2, pp. 1077–1080, 2001.
- [44] R. Lukac and K. N. Plataniotis, "Digital camera zooming on the color filter array," *Electronics Letters*, Vol. 39, No. 25, pp. 1806–1807, 2003.
- [45] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicing approach," *IEEE Trans. Consumer Electronics*, Vol. 50, No. 1, pp. 256–261, 2004.
- [46] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image postprocessing using local color ratios," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 14, No. 6, pp. 914–920, 2004.
- [47] R. Lukac, K. Martin, and K. N. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Trans. Consumer Electronics*, Vol. 50, No. 1, pp. 15–24, 2004.

- [48] R. Lukac and K. N. Plataniotis, "Normalized color-ratio modeling for CFA interpolation," *IEEE Trans. Consumer Electronics*, Vol. 50, No. 2, pp.737–745, 2004.
- [49] R. Lukac and K. N. Plataniotis, "Digital zooming for color filter array based image sensors," *Real-Time Image*, Vol. 11, No. 2, pp. 129–138, 2005.
- [50] R. Lukac, K. N. Plataniotis, and D. Hatzinakos, "Color image zooming on the Bayer pattern," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 15, No. 11, pp. 1475–1492, 2005.
- [51] R. Lukac and K. N. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Trans. Consumer Electronics*, Vol. 51, No. 4, pp.1260–1267, 2005.
- [52] R. Lukac and K. N. Plataniotis, "A robust, cost-effective postprocessor for enhancing demosaicked camera images," *Real-Time Image*, Vol. 11, No. 2, pp. 139–150, 2005.
- [53] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A new CFA interpolation framework," *Signal Processing*, Vol. 86, No. 7, pp. 1559–1579, 2006.
- [54] R. Lukac and K. N. Plataniotis, "A Taxonomy of Color Image Filtering and Enhancement Solutions," in *Advances in Imaging and Electron Physics*, (ed.) P.W. Hawkes, Elsevier/Academic Press, Vol. 140, pp.187–264, 2006.
- [55] D. D. Muresan and T. W. Parks, "Demosaicing using optimal recovery," *IEEE Trans. Image Processing*, Vol. 14, No. 2, pp. 267–278, 2005.

- [56] M. K. Mandal, T. Aboulnasr, and S. Panchanathan, "Image indexing using moments and wavelets," *IEEE Trans. Consumer Electronics*, Vol. 42, No. 3, pp. 557–565, 1996 .
- [57] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recognition Letters*, Vol. 18, No. 10, pp. 1065–1071, 1997.
- [58] E. Navon, O. Miller, and A. Averbuch, "Color image segmentation based on adaptive local thresholds," *Image and Vision Computing*, Vol. 23, No. 1, pp. 69–85, 2005.
- [59] Y. S. Park and H. W. Park, "Arbitrary-ratio image resizing using fast DCT of composite length for DCT-based transcoder," *IEEE Trans. Image Processing*, Vol. 15, No. 2, pp. 494–500, 2006.
- [60] G. Paschos, I. Radev, and N. Prabakar, "Image content-based retrieval using chromaticity moments," *IEEE Trans. Knowledge and Data Engineering*, Vol. 15, No. 5, pp. 1069–1072, 2003.
- [61] S. C. Pei and C. M. Cheng, "Extracting color features and dynamic matching for image data-base retrieval," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 9, No. 3, pp. 501–512, 1999.
- [62] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 13, No. 6, pp. 503–513, 2003.

- [63] S. C. Pei, Y. C. Zeng, and C. H. Chang, "Virtual restoration of ancient Chinese paintings using color contrast enhancement and lacuna texture synthesis," *IEEE Trans. Image Processing*, Vol. 13, No. 3, pp. 416–429, 2004.
- [64] W. K. Pratt, *Digital Image Processing*, 2nd Ed., Wiley-Interscience, Hoboken, NJ, USA, 2001.
- [65] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still camera suitable for a 32-bit MCU," *IEEE Trans. Consumer Electronics*, Vol. 44, No. 4, pp. 1342–1352, 1998.
- [66] S. J. Sangwine and R. E. N. Horne, *The Colour Image Processing Handbook*, Chapman & Hall, London, UK, 1998.
- [67] J. Scharcanski and A. N. Venetsanopoulos, "Edge detection of color images using directional operators," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 7, No. 2, pp. 397–401, 1997.
- [68] F. Y. Shih and S. Cheng, "Automatic seeded region growing for color image segmentation," *Image and Vision Computing*, Vol. 23, No. 10, pp. 877–886, 2005.
- [69] J. L. Starck, F. Murtagh, E. J. Candes, and D. L. Donoho, "Gray and color image contrast enhancement by the curvelet transform," *IEEE Trans. Image Processing*, Vol. 12, No. 6, pp. 706–717, 2003.
- [70] J. L. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Processing*, Vol. 11, No. 6, pp. 131–141, 2002.

- [71] M. Stricker and M. Orengo, "Similarity of color images," in: *Proc. of SPIE: Storage and Retrieval for Image and Video Databases III*, Vol. 2420, pp. 381–392, 1995.
- [72] C. Y. Su, "Highly effective iterative demosaicing using weighted-edge and color-difference interpolations," *IEEE Trans. Consumer Electronics*, Vol. 52, No. 2, pp. 639–645, 2006.
- [73] C. Theoharatos, G. Economou, and S. Fotopoulos, "Color edge detection using the minimal spanning tree," *Pattern Recognition*, Vol. 38, No. 4, pp. 603–606, 2005.
- [74] P.J. Toivanen, J. Ansamaki, J. P. S. Parkkinen, and J. Mielikainen, "Edge detection in multispectral images using the selforganizing map," *Pattern Recognition Letter* Vol. 24, No. 16, pp. 2987–2993, 2003.
- [75] P. E. Trahanias and A. N. Venetsanopoulos, "Color edge detection using order statistics," *IEEE Trans. Image Processing*, Vol. 2, No. 2, pp. 259–264, 1993.
- [76] A. Tremeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Trans. Image Processing*, Vol. 9, No. 4, pp. 735–744, 2000.
- [77] H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. Image Processing*, Vol. 11, No. 4, pp. 485–492, 2002.
- [78] C. Y. Tsai and K. T. Song, "Heterogeneity–projection hard-decision color interpolation using spectral-spatial correlation," *IEEE Trans. Image Processing*, Vol. 16, No. 1, pp. 78–91, 2007.

- [79] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicing for digital cameras (patent pending)," *IEEE Trans. Image Processing*, Vol. 13, No. 9, pp. 1263–1274, 2004.
- [80] E. Zaharescu, M. Zamfir, and C. Vertan, "Color morphology-like operators based on color geometric shape characteristic," in: *Proc. of Int. Symposium on Signal Circuit and Syatems*, Vol. 1, pp. 145–148, 2003.
- [81] S. D. Zenzo, "A note on the gradient of a multi-image," *Computer Vision Graphics and Image Processing*, Vol. 33, No. 1, pp. 116–125, 1986.
- [82] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error interpolation," *IEEE Trans. Image Processing*, Vol. 14, No. 12, pp. 2167–2178, 2005.
- [83] L. Zhang and D. Zhang, "A joint demosaicking–zooming scheme for single chip digital color cameras," *Computer Vision and Image Understanding*, Vol. 107, No. 1–2, pp. 14–25, 2007.
- [84] S. Y. Zhu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Comprehensive analysis of edge detection in color image processing," *Optical Engineering*, Vol. 38, No. 4, pp. 612–625, 1999.
- [85] [Online]. Available: <http://www.site.uottawa.ca/~edubois/demosaicking/>.
- [86] [Online]. Available: <http://140.118.175.164/WJYang/paper/Demosaic/>.
- [87] [Online]. Available: <http://140.118.175.164/WJYang/paper/AResizeCFA/>.



# Curriculum Vitae

Wei-Jen Yang received the B.S. degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2004. He is currently pursuing the Ph.D. degree in Computer Science and Information Engineering at National Taiwan University, Taipei, Taiwan. His research interests include color image processing, digital camera image processing, image/video compression, computer vision, pattern recognition, and algorithms.



# Publication List

## Journal Papers

- 1 K. L. Chung, W. J. Yang, P. Y. Chen, W. M. Yan, and C. S. Fuh, "New joint demosaicing and zooming algorithm for color filter array," *IEEE Trans. Consumer Electronics*, accepted, 2009.
2. K. L. Chung, W. J. Yang, W. M. Yan, and C. C. Wang, "Demosaicing of color filter array captured images using gradient edge detection masks and adaptive Heterogeneity-projection," *IEEE Trans. Image Processing*, Vol. 17, No.12, 2008, pp. 2356-2367.
3. K. L. Chung, W. J. Yang, and W. M. Yan, "Efficient edge-preserving algorithm for color contrast enhancement with application to color image segmentation," *J. of Visual Communication and Image Representation*, Vol. 19, No.5, 2008, pp. 299-310.
4. K. L. Chung, W. J. Yang, J. H. Yu, W. M. Yan, and C. S. Fuh, "A novel quality-effective zooming algorithm for color filter array," *J. of Electronic Imaging*, revised, 2009.
5. K. L. Chung, W. J. Yang, W. M. Yan, and C. S. Fuh, "New Joint demosaicing and

arbitrary-ratio resizing algorithm for color filter array using DCT approach,”  
*IEEE Trans. Consumer Electronics*, submitted, 2009.

## Conference Papers

1. K. L. Chung, W. J. Yang, M. J. Chen, W. M. Yan, and C. S. Fuh, “Joint demosaicing and arbitrary-ratio resizing algorithm for Bayer color filter array based on DCT approach,” in *19th International Conf. on Computer Graphics and Vision*, accepted, 2009.
2. J. H. Yu, K. L. Chung, W. J. Yang, W. M. Yan, and C. S. Fuh, “Efficient zooming algorithm for color filter array using gradient edge detection masks and color differences,” in *Proc. of IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Ilan, Taiwan, 2008, Session C3-4, paper# 92, (Excellent Paper Award).
3. K. L. Chung, W. J. Yang, W. M. Yan, and C. C. Wang, “Novel high-quality demosaicing algorithm for color filter array,” in *Proc. of IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Miaoli, Taiwan, 2007, pp. 509-516, (Best Paper Award).
4. W. J. Yang, W. M. Yan, and K. L. Chung, “An efficient edge-preserving algorithm for color contrast enhancement,” in *Proc. of IPPR Conf. on Computer Vision, Graphics, and Image Processing*, Taipei, Taiwan, 2005, pp. 1177-1183, (Excellent Paper Award).