APPLICATIONS OF HALFTONING TECHNIQUES IN 3D PROFILOMETRY AND REVERSIBLE COLOR-TO-GRAYSCALE CONVERSION

XU ZI-XIN

Ph.D

The Hong Kong Polytechnic University

2017

The Hong Kong Polytechnic University

Department of

Electronic and Information Engineering

Applications of Halftoning Techniques in 3d Profilometry and Reversible Color-to-grayscale Conversion

XU ZI-XIN

A thesis submitted in partial fulfilment of requirements for the degree of Doctor of Philosophy

June 2017

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

______________________________ (Signed)

XU ZI-XIN (Name of student)

Abstract

Digital halftoning is a technique originally proposed to render a gray scale image with binary value pixels. It is able to shift the quantization noise to the high frequency band. Accordingly, the quantization noise can be removed through the lowpass filtering effect of human visual systems to make a binary image visually identical to the original gray level image. Digital halftoning was successfully extended to render color images and it has been widely used in printing applications nowadays. However, its application in other areas is still limited. The objective of this work is to explore possible applications of digital halftoning in areas other than printing. This thesis reports some findings that we had and some contributions that we made during our study. In particular, the focus will be on the application of digital halftoning in (1) 3D profilometry and (2) reversible color to grayscale conversion.

Digital fringe projection technique has been widely used in commercial 3D depth map acquisition in the past decades due to its simplicity, reliability and flexibility. When it is used, fringe patterns are projected onto the object for evaluating its depth information. Phase-shifting sinusoidal patterns are popular patterns used in digital fringe projection. To support real-time measurement and get rid of the luminance nonlinearity of a projector, the binary defocusing method has been proposed to replace sinusoidal patterns with binary patterns. In its practical realization, the binary patterns should be good approximations of the sinusoidal patterns and contain only high frequency approximation errors. Since the projector is defocused, the patterns projected onto the object are blurred and the high frequency approximation error can be removed to some extent. Obviously, halftoning can be applied to produce binary fringe patterns based on the target sinusoidal fringe patterns.

From signal processing point of view, the binary fringe patterns used in binary defocusing method can be considered as 1-bit quantization outputs of sinusoidal fringe patterns. Obviously, the projected fringe patterns impact the measurement quality directly and hence it is a good move to improve the quality of a quantized fringe pattern by increasing its quantization levels. By making use of the fact that a digital-light-processing projector can actually project color patterns via three different channels simultaneously, we proposed a method to project octa-level fringe patterns at no extra cost as compared with projecting binary fringe patterns. Accordingly, while the measurement performance can be improved, the advantages of binary defocusing can be maintained. In other words, it still supports real-time measurement and does not need to handle the luminance nonlinearity of a projector. A framework for optimizing octa-level fringe patterns to support this projection method was also proposed in this work.

The extent of defocusing plays a critical role in determining the quality of the defocused fringe patterns. In real situations, it is impossible to control it precisely and hence the fringe patterns should make their performance robust to it. To respond to this issue, conventional fringe pattern generation schemes optimize halftone patterns under different conditions (e.g. different patch sizes and different defocusing extent) and then, from the optimized results, pick the one which is the most robust to defocusing conditions. This pick-the-best-from-the-available approach is passive to some extent and makes the optimization effort grow in multiples.

To provide flexibility and reduce effort, the optimization processes of recent binary fringe pattern generation schemes are generally patch-based by making use of the property that a sinusoidal fringe pattern is periodic. In general, they optimize one single halftone patch to make its defocused output close to a patch of a sinusoidal fringe pattern and then tile it repeatedly to construct all full-size fringe patterns. We have three observations on this common strategy as follows: (1) The distortion of the tiling result with respect to an ideal sinusoidal fringe pattern is periodic and contains strong harmonics, which affects the measuring performance remarkably. (2) The fringe period is bound to be an integer multiple of 3. (3) It introduces extra constraint for the optimization process.

To solve all these problems, we proposed a framework for generating aperiodic octa-level fringe patterns based on optimized patches. The produced fringe patterns can significantly lower the noise floor and suppress the harmonic distortion in the constructed depth map. Accordingly, the achieved depth measuring performance can be significantly improved. Special care is also taken during the optimization of the patches in our framework such that the depth measuring performance is robust to the variation of fringe period and defocusing extent.

Though conventional fringe pattern generation schemes formulate the pattern generation as an optimization problem, the problem is generally solved by iteratively refining an initial estimate due to its unaffordable complexity. This strategy cannot guarantee a global optimum in terms of an objective function. When the refining step is not flexible, the solution can be biased to the initial estimate and its performance can be far from the optimal. Besides, a poor initial estimate can easily guide the optimization process to reach a poor local optimum at the end. In view of this, a better initial estimate and a flexible refining scheme would definitely be helpful to get better fringe patterns.

By fulfilling these two demands and extending the idea of our previous work, we further developed a novel method to generate patch-based octa-level fringe patterns for improving the measuring performance of a 3D surface measuring system.

Specifically, the optimized patches can be flexibly and seamlessly tiled to form octa-level fringe patterns. As compared with the fringe patterns generated with our previously proposed method, their produced depth maps contain no harmonic distortion along any direction instead of just one single direction.

Reversible color-to-grayscale conversion (RCGC) aims at embedding the chromatic information of a full color image into its grayscale version such that the original color image can be reconstructed in the future when necessary. Conventional RCGC algorithms tend to put their emphasis on the quality of the reconstructed color image, which makes the color-embedded grayscale image visually undesirable and suspicious. As an output of this study, a novel RCGC framework that emphasizes the quality of both the color-embedded grayscale image and the reconstructed color image simultaneously is proposed. Its superiority against other RCGC algorithms is mainly achieved by developing a color palette that fits into the application and exploiting error diffusion to shape the quantization noise to high frequency band. The improved visual quality of the color-embedded grayscale image makes the image appears as a normal image. It does not catch the attention of unauthorized people and hence the embedded chromatic information can be protected more securely.

The color palette used in the proposed RCGC framework is critical to the conversion performance. To fit into the application, we proposed a palette generation algorithm to generate an image-dependent palette that bears two properties. First, palette colors are sorted and indexed according to their luminance values such that the index plane looks closely to the luminance plane of the original color image. Second, consecutive colors in the palette form a three-dimensional enclosure in the color space to cover as many pixel colors that have the same luminance values as the involved palette colors as possible. Theoretically, with the halftoning technique, any specific color inside the enclosure can be rendered with the palette colors that form the enclosure.

The aforementioned palette generation algorithm was further improved in an extended study. The idea comes from two observations. First, halftoning does not work properly when the region having the color to be rendered is very small in the image and the color is very different from the available palette colors. Second, from mean square error point of view, using a palette color directly to replace a pixel color can be more efficient than rendering it with halftoning.

While our first proposed palette generation algorithm concerns whether a color can be rendered with halftoning in ideal situations, the improved palette generation algorithm considers whether a color can be effectively rendered in practical situations. We proposed a measure to estimate the effectiveness and appropriateness of using halftoning to render a specific color in a spatial region. The measure is incorporated into the objective function to optimize the color palette. As a result, the color palette can work with our proposed RCGC framework in a better way and achieve a better RCGC performance.

In this thesis, we present some ideas for the applications of digital halftoning in 3D profilometry and reversible color to grayscale conversion. As a tool of noise shaping, digital halftoning can be applied in different areas. We expect there will be more novel application ideas coming in future.

Publications arising from the thesis

International Journal Papers

1. Z.-X. Xu and Y.-H. Chan, “Improving reversible color-to-grayscale conversion with halftoning,” *Signal Processing: Image Communication*, vol. 52, pp. 111–123, 2017.

2. Z.-X. Xu and Y.-H. Chan, “Removing harmonic distortion of measurements of a defocusing three-step phase-shifting digital fringe projection system,” *Optical Lasers Engineering*, vol. 90, pp. 139–145, 2017.

3. Z.-X. Xu and Y.-H. Chan, “High-quality octa-level fringe pattern generation for improving the noise characteristics of measured depth,” accepted to appear in *Optical Lasers Engineering*.

4. Z.-X. Xu and Y.-H. Chan, “A color palette generation scheme for reversible color-to-grayscale conversion with error diffusion,” to be submitted for possible publication in *Signal Processing: Image Communication*.

International Conference Papers

1. Z.-X. Xu and Y.-H. Chan, “Eliminating blocking artifacts of halftoning-based block truncation coding,” Proceedings, 2016 European Signal Processing Conference (EUSIPCO’2016), Aug 29 - Sep 2, 2016, Budapest, Hungary, pp.913-017.

2. Z.-X. Xu and Y.H. Chan, “Optimized multilevel fringe patterns for real-time 3D shape measurement with defocused projector,” Proceedings, 2015 IEEE

International Conference on Image Processing (ICIP’15), Quebec City, Canada, Sep 27-30, 2015, pp.2730-2734.

3. Z.-X. Xu and Y.H. Chan, “A constrained error diffusion method to shape quantization noise in backlight dimming,” Proceedings, 19th International Conference on Digital Signal Processing (DSP 2014), Hong Kong, Aug 21-23, 2014, pp.909-913

Acknowledgement

I am grateful to my supervisor Dr. Yue-Hee Chan, who introduces me to digital halftoning theory and other image processing technologies. His scientific knowledge, his insight and challenging idea was fundamental to this work.

I wish to thank Dr. Daniel P.K Lun for sharing his research idea and simulation results in 3D shape measurement.

I wish to thank Prof. Wan-Chi Siu, Prof. Kin-Man Lam and Prof. Francis C.M Lau for their kind supervision on the course of this research.

I wish to thank Dr Y.H. Fung. He provided me a number of inspiring ideas in the application of halftoning techniques.

I also wish to thank the Department of Electronic and Information Engineering for their help on my living and study in Hong Kong.

I am grateful to the Research Grant Council of the Hong Kong Special Administrative Region, China and the Hong Kong Polytechnic University for their financial support.

Finally, I wish to express my appreciation to my parents and my wife for their invaluable support during my life.

Content

List of Figures

List of Tables

List of Abbreviations

B/W	Black & White
BTC	Block Truncation Coding
LED	Light-emitting Diode
LCD	Liquid-crystal Display
2D	Two Dimensional
3D	Three Dimensional
RCGC	Reversible color-to-grayscale conversion
SE-RCGC	Subband embedding RCGC
VQ-RCGC	Vector Quantized RCGC
HVS	Human Visual System
MED	Multiscale Error Diffusion
FTP	Fourier Transform Profilometry
PSP	Phase Shifting Profilometry
MSE	Mean Square Error
PWM	Pulse Width Modulation
RMS	Root Mean Square
PSNR	Peak Signal to Noise Ratio
SSIM	Structural Similarity Index Measure
CSSIM	Structural Similarity Index Measure for Color Images
SA	Simulated Annealing

Chapter 1.

Introduction

1.1 Overview

Digital halftoning [1, 2] is a technique originally proposed to render a gray scale image with binary value pixels. It is basically a noise shaping technique used to shift the quantization noise to the high frequency band when reducing the number of gray levels of an image. Since our human visual system behaves as a low pass filter, the quantization noise can be removed. It makes a binary image visually identical to the original gray level image.

Digital halftoning was successfully extended to render color images and it has been widely used in printing applications nowadays. However, its application in other areas is still relatively very limited. The objective of this work is to explore possible applications of digital halftoning in areas other than printing. This thesis reports some findings that we had in our recent studies and some contributions that we made during our studies. In order not to diversify the issues to be discussed, the focus of the thesis will be on the application of digital halftoning in (1) 3D profilometry and (2) reversible color to grayscale conversion.

3D shape measurement, which is also named as 3D surface sensing or 3D reconstruction, aims to detect the depth information of an object. 3D shape measurement has a variety of applications including vehicle guidance, geometry checking, industrial monitoring, etc. Among various types of 3D shape measurement methods, the application of optical structured light [3] tends to be a simple yet accurate way to obtain the depth information. Optical structured light methods, also known as active triangulation, can be implemented with a well-calibrated measurement system that is equipped with one or more digital projectors and a high-speed camera. This measurement system has several advantages: 1) its implementation is easy and cheap; 2) the whole measurement scheme can be well controlled by a computer without the need of human supervision; and 3) the measurement accuracy is high. Consequently, structured light system is one of the most successful methods for commercial use.

Given a structured light measurement system, the fringe patterns projected onto the object being measured directly impact on the measurement quality. On the other hand, the measurement speed is bounded by the frame rate of a projector. To release the measurement speed bottleneck, binary fringe patterns were proposed to replace grayscale fringe patterns in 1992 [4] under the belief that a defocused projector can remove the difference between a binary fringe pattern and a grayscale fringe pattern. Since the introduction of this binary defocusing method, various studies [5–24] have been presented in the literature to improve the quality of binary fringe patterns.

The major concern of using binary fringe patterns is how to eliminate the phase errors introduced by the quantized fringe patterns. In Chapter 3, we proposed a method to project octa-level fringe patterns at no extra cost as compared with projecting binary fringe patterns [25]. By increasing the actual number of intensity levels of the projected fringe patterns, the quantization noise floor is efficiently reduced without sacrificing the advantages of binary fringe patterns.

In order to increase the flexibility and to reduce the development effort of the fringe patterns, advanced fringe pattern generation algorithms [6, 10, 18, 23] are generally patch-based in a way that small patches are optimized and then tiled to form the final full-size fringe patterns. The noise of their tiling results is periodic and the consequence is that severe harmonic distortion exists in their measurement results. In Chapter 4, we proposed a tiling method to construct aperiodic patch-based octa-level fringe patterns and an optimization method to produce patches for supporting the tiling method [26]. Some other weaknesses of conventional pattern generation algorithms were also considered in our proposal such that they can be alleviated to a large extent.

In Chapter 5, we extend the idea proposed in [26] from one dimension to two dimensions such that the harmonic distortion can be suppressed in both horizontal and vertical directions. We also propose a better optimization algorithm by improving both the way to get an initial estimate and the way to search for a better estimate.

Reversible color-to-grayscale conversion (RCGC) aims at representing a full color image as a grayscale image without discarding its chrominance information so that the color image can be recovered in future. Sometimes, due to some practical constraints, we are forced to deliver or present an image in grayscale temporarily but we need the color information in future. RCGC is also useful in scenarios where we need to protect the chrominance information from being revealed to unauthorized users [27–31].

Various RCGC algorithms have been proposed since the introduction of RCGC in 2006 [32], and they can be categorized into Subband embedding (SE) based and Vector Quantization (VQ) based algorithms. In both approaches, chrominance information is embedded into the luminance plane of the color image to produce the color-embedded grayscale image. When necessary, the embedded chrominance information is extracted to recover the color image. The key of success is to minimize the distortion in both the color-embedded grayscale image and the recovered color image.

SE-RCGC algorithms [32–37] embed the chrominance information into the luminance plane of an image by replacing some high frequency luminance bands with the downsampled chrominance planes. It introduces visible pattern noise in the color-embedded grayscale image and blurs the recovered color image. VQ-RCGC algorithms [27–31] color-quantizes a color image under a constraint that the index plane is close to the luminance plane of the image, and then embeds the color palette into the index plane. Since the constraint is not easy to fulfill, conventional VQ-based algorithms tend to emphasize the quality of the recovered color image more. Even so, visible artifacts such as color shift and false contour can generally be found in their recovered color images.

To tackle the weaknesses of VQ-based algorithms, in Chapter 6 we proposed a novel VQ-based RCGC framework that emphasizes both the quality of the color-embedded grayscale image and the recovered color image [38]. By incorporating the error diffusion technique [39] and developing a tailor-made color palette to support the use of the error diffusion technique, our algorithm developed under the proposed framework can significantly improve the quality of the color-embedded grayscale image and the recovered color image simultaneously.

In the aforementioned proposal, we assume that any colors in a color image can be ideally rendered with a halftoning technique. In Chapter 7, we consider the practical constraints of halftoning in rendering a color and propose a new palette generation algorithm accordingly. When working under the framework proposed in Chapter 6, the newly generated palette is able to further improve the quality of the color-embedded grayscale image and the recovered color image remarkably in terms of both subjective and objective criteria.

1.2 Contribution of this work

In this thesis, the application of digital halftoning techniques in (1) 3D profilometry and (2) reversible color to grayscale conversion has been extensively discussed. The following contributions reported in this thesis are claimed to be original.

1) We proposed a novel framework that measures 3D objects with octa-level fringe patterns instead of binary fringe patterns. It lowers the noise floor of a measurement result while maintaining all advantages offered by a system that uses binary fringe patterns (Chapter 3).

2) We proposed a patch optimization algorithm that optimizes patches for supporting random tiling along a specific direction (Chapter 4). It also removes some limitations of conventional patch-based fringe patterns and proactively improves the robustness of the fringe patterns to defocusing extent.

3) We proposed a patch optimization algorithm that optimizes patches for supporting random tiling along any directions (Chapter 5). Optimized patches can be used to construct fringe patterns without two-dimensional harmonic distortion.

4) We proposed a better approach to produce an initial estimate and a better strategy to avoid being easily trapped in local minimums when optimizing a patch. (Chapter 5)

5) We proposed a novel VQ-based RCGC framework that significantly improves the output image quality of both the color-embedded grayscale image and the

recovered color image by shifting the noise to the high frequency bands of both images simultaneously with a halftoning technique (Chapter 6).

6) We proposed a tailor-made color palette generation algorithm that produces color palettes for supporting the VQ-based RCGC framework mentioned in 5). It takes the halftoning process to be performed in RCGC into account when optimizing the color palette (Chapter 6).

7) We proposed a tailor-made color palette generation algorithm that produces color palettes for supporting the VQ-based RCGC framework mentioned in 5). It takes the practical constraint of a halftoning process into account when optimizing a color palette (Chapter 7).

1.3 Organization of the thesis

This thesis consists of 8 Chapters and it is organized as follows.

Chapter 1 is a brief introduction of the work discussed in this thesis.

Chapter 2 provides a brief review of the background knowledge on which the present work is based. It covers various issues relevant to digital halftoning, 3D shape measurement and reversible color-to-grayscale conversion.

Chapters 3, 4 and 5 present our work in 3D profilometry. Some ideas and methods are proposed to generate octa-level fringe patterns for improving the accuracy of 3D shape measurement. Chapter 3 introduces the idea of octa-level fringe pattern generation and shows how it can improve measurement accuracy while maintaining the advantages of using binary fringe patterns. Chapter 4 presents a patch-based octa-level fringe pattern generation algorithm that alleviates the harmonic distortion in phase domain and removes some limitations of patch-based fringe patterns. Chapter 5 extends the idea discussed in Chapter 4 to further suppress harmonics in both horizontal and vertical directions. It also provides a method to reach a better optimization result by generating a better initial estimate and adopting a more flexible strategy in the optimization.

Chapters 6 and 7 are dedicated to presenting our work in reversible color-to-grayscale conversion. Chapter 6 presents a VQ-based RCGC algorithm that makes use of a halftoning technique and a matched color palette to significantly improve the quality of color-embedded grayscale images and recovered color images. Chapter 7 provides an optimization algorithm for generating a color palette to further improve the performance of the RCGC algorithm proposed in Chapter 6.

The thesis is concluded in Chapter 8 with a summary of the work that was done in this project. Future possible extensions of the present work are also discussed in this final Chapter.

Chapter 2.

A comprehensive literature review

2.1 Introduction

The objective of this work is to explore possible applications of digital halftoning in areas other than printing. In particular, our focus is on the application in 3D profilometry and reversible color to grayscale conversion. Accordingly, this Chapter provides some background information of digital halftoning, 3D profilometry and reversible color to grayscale conversion. It also reviews some state-of-art works that are relevant to our present works.

2.2 Digital halftoning

Digital halftoning aims to render a grayscale image by distributing binary dots in a way that its visual appearance is as close to the original as possible [1, 2]. Traditionally, digital halftoning has been widely applied in bi-level devices such as black-and-white printers and fax machines. A number of digital halftoning methods have been discussed in the literature. Among them, error diffusion [39–41] is widely used because it can provide good halftones with a reasonable computational cost. In general, digital halftoning algorithms can be categorized into two classes: amplitude modulation (AM) halftoning and frequency modulation (FM) halftoning. AM halftoning converts a continuous image to a binary one by varying the size of printed dots arranged along a regular grid, and FM halftoning produces halftones by varying the density of fixed-size printed dots spatially. There are also hybrid halftoning algorithms in which both dot density and dot size are used to render the gray levels in a spatial region of the image.

In general, FM halftoning produces isolated dots and the halftoning outputs can support a higher spatial resolution. As our human visual system (HVS) actually behaves as a spatial low pass filter, FM halftoning tends to modulate the quantization noise to the high frequency band so that the halftone dots are not visible to human beings. Accordingly, the so-called blue noise model is used as the statistical model to describe the ideal noise characteristics of a halftoning result. A halftoning result having this noise characteristic is visually pleasant and made up with aperiodic dispersed-dot dither patterns.

Ordered dithering [42], which is introduced by Bayer, is one of the simplest ways to generate blue-noise halftones. However, the most popular blue noise halftoning technique is error diffusion. The original version of error diffusion algorithm was proposed by Floyd's and Steinberg's [39]. It is an adaptive method that quantizes image pixels one by one in a raster scanning order. After a pixel’s intensity is quantized, the quantization noise is diffused to its neighbour pixels. Dot diffusion [43] is a variety of error diffusion which is suited to parallel computation. Direct binary search (DBS) [44] is an iterative algorithm that tries to minimize the perceived error between the original continuous-tone image and the halftoning output based on a given HVS model.

Multiscale error diffusion (MED) is an iterative method proposed in [45], and then significantly improved in [46]–[48]. The idea of feature preserving was presented in [47] and the method is called feature preserving multiscale error diffusion (FMED). and it has been proved that feature preserving multiscale error diffusion (FMED) can preserve features of the original grayscale image in spatial domain faithfully and adjust the noise characteristics of its halftoning outputs in spectral domain flexibly [48, 49].

2.2.1 Ordered dithering

Introduced by Bayer [42], ordered dithering is a simple screening method to generate blue noise halftone. Ordered dithering compares the grayscale image with a 2D threshold array called Bayer kernel. Bayer kernel is a group of optimal threshold array of different sizes. The smallest 2×2 kernel is given as:

$$M_1 = \begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} \tag{2.1}$$

A larger Bayer kernel can be calculated by

$$M_{N+1} = \begin{bmatrix} 4M_N & 4M_N + 2E_N \\ 4M_N + 3E_N & 4M_N + E_N \end{bmatrix} \tag{2.2}$$

where E_N is N×N unit matrix.

8×8 Bayer kernel or 16×16 Bayer kernel is commonly used in halftoning. The kernel tiles itself to match the image size, and the image intensity values are compared to the tiled threshold array to generate the final halftone.

2.2.2 Error diffusion

Error diffusion was introduced by Floyd and Steinberg [39]. The algorithm processes the image in a raster scanning order in which pixels are scanned from left to right and top to bottom. For each pixel, the algorithm performs thresholding and the quantization error of that pixel is diffused to its neighbours with a casual filter. The block diagram of error diffusion is given in Figure 2-1.

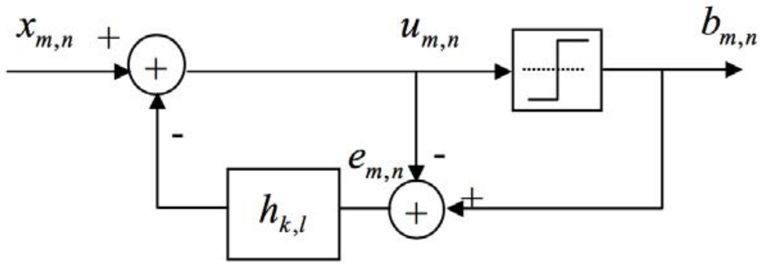


Figure 2-1 The block diagram of error diffusion

As shown in Figure 2-1, the intensity value of pixel (*m*,*n*), say $x_{m,n}$, is modified by the weighted errors diffused from some previously processed neighbouring pixels to produce $u_{m,n}$. $u_{m,n}$ is then quantized to either 0 or 1 as $b_{m,n}$. The quantization error between $u_{m,n}$ and $b_{m,n}$ is diffused to pixel (*m*,*n*)'s neighbouring pixels. Mathematically, the process can be formulated as

$$u_{m,n} = x_{m,n} - \sum_{(k,l)\in\Omega} h_{k,l} e_{m-k,n-l} \tag{2.3}$$

$$b_{m,n} = Q(u_{m,n}) = \begin{cases} 0 & if \quad u_{m,n} < 0.5 \\ 1 & \quad otherwise \end{cases} \tag{2.4}$$

$$e_{m,n} = b_{m,n} - u_{m,n} \tag{2.5}$$

where Ω is the support region of the diffusion filter and $h_{k,l}$ is the $(k,l)^{\text{th}}$ coefficient of the diffusion filter. The coefficients of the diffusion filter suggested by Floyd and Steinberg [39] is

$$h = \begin{bmatrix} 0 & * & 7 \\ 3 & 5 & 1 \end{bmatrix} / 16 \tag{2.6}$$

where * marks the current pixel position. Floyd and Steinberg's method [39] is referred to as standard error diffusion. Its halftoning outputs generally suffer from artifacts such as worm-like texture, pattern noise and directional hysteresis. The output quality can be improved by using some other diffusion filters such as [40, 41], or serpentine scanning order [1]. There are some other remedial solutions. Examples are as follows. Kolpatzik and Bouman [50] optimized the error diffusion system based on the weighted mean square error. Adaptive error diffusion [51] adjusts the error diffusion filter along with the processing pixels to minimize local errors. Li [52] proposed to preserve edge information in the error diffusion halftone.

Tone-dependent error diffusion (TDED) was introduced in [53], and it has recently been improved by Fung and Chan [54, 55]. The idea of TDED is to adjust both the filter weights and the quantization threshold according to the input pixel intensity value. The filter coefficients and the quantization threshold for each grey-level are optimized offline to minimize the perceived error under a particular HVS model.

Standard error diffusion can be applied directly to color halftoning by processing different color channels separately. However, since the information among color channels is usually correlated, various improvements have been made to diffuse vector quantization noise directly [56–60].

2.2.3 Multiscale error diffusion

Since standard error diffusion uses a fixed scanning order and a casual diffusion filter, its halftoning outputs unavoidably suffer from artifacts such as worm-like texture, pattern noise and directional hysteresis. Although a number of remedial solutions have been proposed (e.g. [40], [41] and [50]–[55]), the shortcomings of error diffusion were still not resolved.

A brand-new alternative method, called multiscale error diffusion (MED), was proposed in [45], and hereafter improved by Chan’s group [46, 47]. The concept of MED is to replace the fixed scanning order by an undetermined order. By doing so, a non-casual diffusion filter can be used to diffuse a quantization error and hence directional hysteresis can be removed. Although MED is capable to generate high quality blue noise halftones, its high computation complexity prevents it from some real-time applications.

MED is basically an iterative algorithm. In each iteration cycle, it searches a pixel to assign a dot and then diffuses the quantization error to the pixel's neighbors. In Katsavounidis's algorithm [45], the search is carried out as follows. Repeatedly, it divides the image into 4 non-overlapped regions and selects the brightest region among them until a pixel is located. This searching strategy is referred to as maximum intensity guidance in [46]. The intensity value of the located pixel is quantized to 1 (the maximum intensity value) and its error is diffused with a non-causal filter. The search-and-quantize-and-diffuse step is repeated until the total energy of the processing output is equivalent to that of the given image.

In [46], Chan modified the above algorithm by dividing image into nine overlapped regions. This modification eliminates the blocking artifacts. Chan also solved the error leakage problem caused by the error diffusion process proposed in [45]. The idea of feature-preserving MED (FMED) was proposed in [47]. Instead of using the maximum intensity guidance to locate a pixel for placing a new dot, FMED exploits extreme intensity guidance to avoid the bias caused by only assigning white dots (dots whose intensity values are 1). Conceptually, minority dots in a local region provide features in the region and should be placed first because they probably stand for critical information. A simple trick is also introduced in [47] to further reduce the boundary effect. It has been proved that FMED can generate high quality halftones by preserving spatial features and achieving desired noise characteristics. Since FMED is used in the development of one of our contributions represented in this thesis, the details of FMED are briefly summarized as follows.

Without loss of generality, we assume that the input image X is of size $2^n \times 2^n$ and all its pixel intensity values are bounded in [0,1], where 0 and 1 are the minimum and the maximum intensity values respectively. To start the process, an error plane denoted as E is initialized to be X. The intensity values of pixels (m, n) of X, E and B are, respectively, denoted as $x_{m,n}$, $e_{m,n}$ and $b_{m,n}$, where B is the halftoning output of FMED.

Step 1: Search a pixel in E according to extreme intensity guidance. The search starts with the error image E as the region of interest. Then the region of interest is divided into nine overlapped sub-regions and the sub-region with the largest sum of its all pixel intensity is selected to be the new region of interest. This step is repeated until a particular sub-region of a particular size is reached. If the average energy of this sub-region is large than 0.5, it is marked as a bright local region and the subsequent search in this local region follows minimum intensity guidance policy instead. Otherwise, it is marked as a dark local region and the subsequent search sticks to the original maximum intensity guidance policy. Eventually a particular pixel location is reached. Suppose its pixel location is (i, j).

Step 2: If the sub-region is marked as a bright local region, assign a black dot to (i, j) $(b_{i,j} = 0)$. Otherwise a while dot is assigned $(b_{i,j} = 1)$. The produced quantization error $b_{i,j} - e_{i,j}$ is diffused to pixel (i, j)'s neighbour pixels in E as:

$$e_{m,n} = \begin{cases} 0 & if \quad (m,n) = (i,j) \\ e_{m,n} - \left(b_{i,j} - e_{i,j}\right)\left(d_{m,n} h_{m-i,n-j}\right)/s & if \quad (m-i, n-j) \in \Omega \end{cases} \tag{2.7}$$

where

$$d_{m,n} = \begin{cases} 0 & \text{if pixel } (m,n) \text{ has been assigned a dot} \\ 1 & \text{otherwise} \end{cases} \tag{2.8}$$

and $\quad s = \sum_{(m-i,n-j)\in\Omega} \left(d_{m,n} h_{m-i,n-j}\right)$ (2.9)

Note that Ω is the filter support of a non-casual diffusion filter h and s is a normalizing factor that makes available coefficients in the diffusion filter sum to 1.

These two steps are repeated until the sum of all pixels of E is bounded in absolute value by 0.5. Note that the total number of iterations is bounded by the total number of pixels in the input image.

A significant advantage of FMED is that the noise characteristics of the generated halftones can be well controlled by the non-casual diffusion filter. For example, ideal blue noise characteristics can be achieved with the diffusion filter proposed in [48], while green noise characteristics can be achieved with the diffusion filter suggested in [49, 61].

2.3 Fringe projection 3D shape measurement

3D shape measurement aims to detect the depth information of an object so that its 3D model can be reconstructed with a computer. It has been widely used in applications such as character rigging, surface topography, robot controls, facial animation, and machine vision, etc. Generally, 3D shape measurement can be categorized as contact and non-contact. Due to its simplicity, flexibility and accuracy, non-contact 3D shape measurement has been widely discussed in the literature and has attracted a number of commercial investments.

A non-contact measurement system can be either active or passive. An active system sends one or more reference signals onto the surface of object to carry out a measurement while a passive system does not. In the work presented in this thesis, our focus is on active measurement methods that project structured light signals. A realization example of such an active measurement system is shown in Figure 2-2. This method is referred to as Fringe Projection Profilometry (FPP).

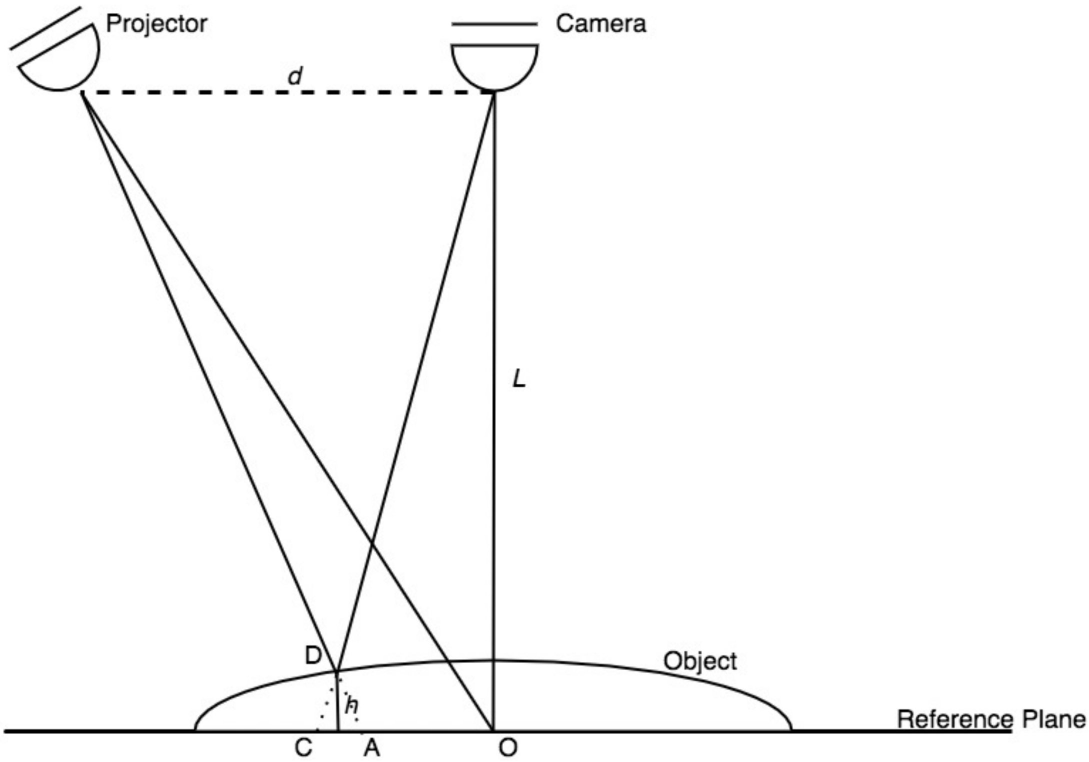


Figure 2-2 Optical geometry of an active 3D measurement system

Two most famous FPP techniques are Fourier Transform Profilometry (FTP) [62, 63] and Phase Shifting Profilometry (PSP) [64]. In either technique, a projector is required to project fringe patterns onto the surface of the measured object and the resulting fringe patterns are captured by a high-speed camera. The depth of the object can then be evaluated based on the phase shift of the fringes caused by the shape of the object. FTP and PSP are different in their ways to calculate the phase map. FTP projects one fringe pattern only while PSP projects at least three fringe patterns with fixed phase differences. As expected, the measure speed of FTP is faster than PSP. However, PSP approaches are less influenced by the environmental noise because the additive and multiplicative interferences can be eliminated by division and subtraction operations. Both approaches face the same practical problem that precise sinusoidal patterns are difficult to produce due to the luminance nonlinearity of a digital projector.

2.3.1 Phase shifting profilometry

In PSP [65], at least three sinusoidal fringe patterns are needed. The method that exploits three fringe patterns is referred to as three-step phase shifting method. It can be easily extended to N-step phase shifting method for any value of N larger than 3. In general, the larger the value of N, the higher the spatial sampling rate can be achieved and it results in higher accuracy but lower measurement speed.

In three-step phase shifting method, the three sinusoidal fringe patterns have a phase shift of $2\pi/3$ from each other. After being projected to the surface of an object, the reflected fringe patterns are captured with a high-speed camera. Let the three captured fringe patterns be denoted as I_k for $k \in \{1,2,3\}$. Their intensity values at pixel (x, y) are given as:

$$I_1(x,y) = A(x,y) + M(x,y)cos(\varphi(x,y) - 2\pi/3) \tag{2.10}$$

$$I_2(x,y) = A(x,y) + M(x,y)cos\big(\varphi(x,y)\big) \tag{2.11}$$

$$I_3(x,y) = A(x,y) + M(x,y)cos(\varphi(x,y) + 2\pi/3) \tag{2.12}$$

where $A(x,y)$ is the average intensity, $M(x,y)$ denotes the amplitude of intensity modulation, and $\varphi(x,y)$ symbolizes the pixel-wise phase to be solved. By solving the three equations, a pixel-wise phase map can be obtained as:

$$\varphi(x,y) = \tan^{-1}(\sqrt{3}\,\frac{I_1(x,y)-I_3(x,y)}{2I_2(x,y)-I_1(x,y)-I_3(x,y)}) \tag{2.13}$$

$\varphi(x,y)$ is a wrapped phase ranging in [-π, π]. Therefore, an unwrapping algorithm [65] is need to eliminate the 2π discontinuities. In a well-calibrated measurement system, the unwrapped phase is linearly proportional to the depth information of the object.

Since a projector generally suffers from luminance nonlinearity, gamma correction [66] is required before its practical usage.

2.3.2 Binary defocusing method

PSP algorithms have two disadvantages. First, the accuracy is limited due to the luminance nonlinearity of digital projectors. Second, the measurement speed is limited because at least three grayscale fringe patterns must be projected in sequence to complete one single measurement. In order to tackle the two drawbacks, the idea of binary defocusing was introduced [4].

The fundamental concept of binary defocusing is that a blurred binary fringe pattern can appear as a grayscale sinusoidal fringe pattern when it is projected by a defocused projector. There is no luminance nonlinearity issue when a binary pattern is projected. Moreover, binary patterns can be easily generated by toggling on and off the LED. In other words, the pulse width modulation procedure used in the digital micromirror device (DMD) of a digital projector can be bypassed. As a result, the frame rate could be increased sharply to the order of kHz. The blurring effect of a defocused projector acts as a low-pass filter to remove the high frequency harmonics of a quasi-sinusoidal binary pattern. However, two new issues are introduced by binary defocusing algorithms: 1) the error introduced by high-order harmonics; and 2) the reduction in depth measurement range.

2.3.2.1 Squared binary fringe patterns

As presented in [4], [16], squared binary fringe patterns are the simplest quasi-sinusoidal binary patterns and they can be obtained by thresholding grayscale sinusoidal fringe patterns. Figure 2-3 shows how a squared binary pattern can be used to approximate a sinusoidal after being blurred.

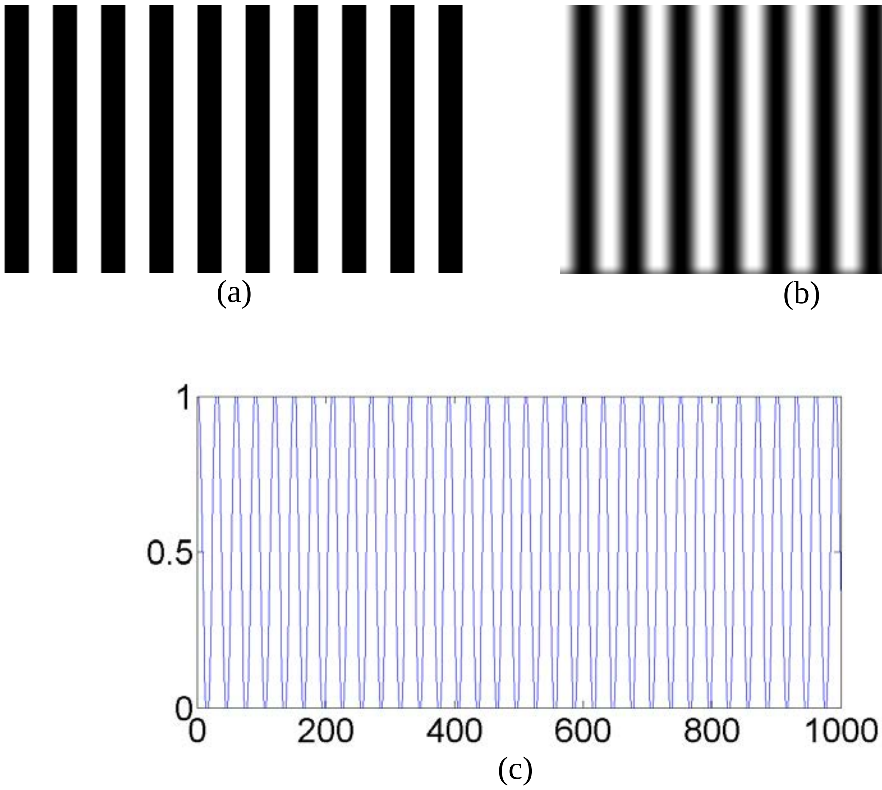


Figure 2-3 (a) a squared binary pattern; (b) the blurred squared binary pattern; (c) the cross-section of (b).

Squared binary pattern is sensitive to the defocusing level. One can observe that the signal shown in Figure 2-3(c) is actually not sinusoidal due to the inappropriate blurring factor used in the simulation. Huge phase error can be introduced if the defocusing level is not ideal. Unfortunately, it cannot be controlled precisely in practical situations.

2.3.2.2 Pulse width modulation-based binary fringe patterns

After the introduction of squared binary method (SBM), endeavours have been made to conquer the challenges of binary defocusing method. A sequence of algorithms adopted the idea of pulse width modulation (PWM) to improve SBM. For example, Ayubi et al. proposed a technique based on sinusoidal pulse width modulation (SPWM) [17]. SPWM is a famous technique used in power electronics to generate sinusoidal signal by filtering binary signals. SPWM is capable to shape the third and higher order harmonics further away from the fundamental frequency. Therefore, the quantization noise is easier to be eliminated by low-pass filtering.

An improved idea called Tripolar SPWM was proposed in [8]. Tripolar SPWM works along with a four-step phase shifting algorithm, which is insensitive to its even-order harmonics. Based on this observation, the author argued to selectively suppress the odd-order harmonics only, as the even-order harmonics have no impact on phase error. This method works better than SPWM when the defocusing level and fringe period are selected ideally.

Optimal pulse width modulation (OPWM) was presented by Wang and Zhang [15]. OPWM selectively eliminates some undesired harmonic frequencies by inserting different types of notches in a conventional binary square wave. A non-linear optimization is required to reach the solution. A comparison study on SBM, SPWM and OPWM was reported in [5].

Zuo et. al. [6, 7] manipulate the 3rd harmonic of a PWM signal in intensity domain. They demonstrated that the 3rd harmonic of a PWM signal does not affect the phase measurement in equation form.

2.3.2.3 Halftoning-based binary fringe patterns

Based on a 1D modulation scheme, PWM-based binary fringe algorithms cannot handle harmonics efficiently especially when the fringe period is small. In such a case, the fundamental frequency is very high and it is difficult to be separated from the harmonics. As a matter of fact, binary fringe patterns are 2D images. In view of this, digital halftoning techniques, which are basically 2D noise shaping techniques, could be efficient alternatives. With halftoning, the quantization noise can be shaped in both horizontal and vertical directions. It makes the noise of the produced binary fringe patterns easier to be filtered away by the blurring effect.

Based on the aforementioned idea, algorithms were proposed to improve the quality of binary fringe patterns. For examples, ordered dithering [42] was adopted to generate binary fringe pattern in [12], [13], and error diffusion [39] was exploited in [21]. Figure 2-4 shows a binary fringe pattern generated with error diffusion. It can be found that intensity values vary along both horizontal and vertical directions. The quantization noise is shaped in both directions.

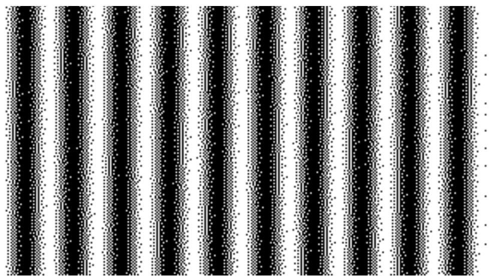


Figure 2-4 Binary fringe pattern obtained by error diffusion

2.3.2.4 Binary fringe pattern optimization

Although halftoning techniques can directly help improving the quality of binary fringe patterns, the performance is still far from optimal. There remain several challenges and an optimization process is required. First, as shown in eqn. (2.13), the final phase map is not a linear function of the fringe patterns. Second, the phase map should be robust to the blurring extent since it is difficult to control the defocusing level precisely in practical situations. A direct application of a halftoning algorithm cannot guarantee that the halftoning result is optimal and hence an optimization process is necessary.

Taken the halftones obtained by ordered dithering or error diffusion as initial estimate, several optimization algorithms have been proposed. For example, a genetic technique is exploited in [21] to realization the optimization. Since the optimization is offline, 10,000 iterations are allowed in the genetic optimization.

Dai and Zhang [12] proposed to optimize the binary fringe patterns by minimizing the resultant phase error directly. Since the phase map is directly correlated to the depth information, the optimization is considered as a proactive optimization.

Although Dai's method can efficiently reduce phase errors, its performance is sensitive to the defocusing level. The method fails to obtain a good phase map when the focal lens of projector is not ideally set. To tackle this problem, optimization based on the quality of fringe patterns was proposed in [11], [23]. These methods tend to minimize the intensity error between the halftone and a target sinusoidal fringe pattern. The optimized binary patterns are close to sinusoids and hence they are more robust to defocusing. As the phase information is not optimized directly, the optimization is classified as a passive optimization.

In addition, Dai et. al. proposed an optimization algorithm in [10] to utilize the symmetry and periodicity of a sinusoidal fringe pattern. Instead of optimizing the whole fringe pattern, a small patch is built and tiled to formulate a binary fringe pattern. Let x-direction be the direction of the sinusoidal variation in a fringe pattern and y-direction be the direction orthogonal to x-direction. A common patch-based optimization framework is briefly reviewed as follows:

Step 1: *Patch formation*. A patch is defined as a dimension of $S_x \times S_y$, where $S_x = {}^{T}\!/_{2}$ is fixed to half of the fringe period T, and S_y varies from 2 to 10 to formulate patches of different sizes.

Step 2: *Patch initialization*. Generate an initial patch by assigning 0 or 1 (the minimum or the maximum intensity value) to its pixel values randomly.

Step 3: *Patch optimization*. Process the initial patch by mutating every pixel in the patch in a raster scanning order. Once a mutation is done, tile the updated patch to form a full-size fringe pattern and compute the difference between the blurred full-size pattern and the ideal sinusoidal fringe pattern. The mutation that minimizes the difference is kept, and the others are discarded. The objective function can be formulated as:

$$\min_{B} J = \mathrm{argmin}_B \|I_k - G \otimes B\|_F \tag{2.14}$$

where *B* is the tiled binary pattern, *G* symbolizes a Gaussian low-pass filter that simulates the defocusing effect, $\|.\|_F$ denotes the Frobenius norm, and $\otimes$ is the 2D convolution operator.

Step 3 is an iterative step that mutates pixels one by one iteratively under the termination criteria is satisfied:

$$J^{(i)} - J^{(i+1)} \Big/ J^{(i)} < 0.01\% \tag{2.15}$$

where $J^{(i)}$ is the objective value *J* obtained after the i^{th} iteration cycle.

Step 4: *Patch variation*. Repeat Step 2 and 3 for a number of times so that a number of good patch candidates are generated.

Step 5: *Patch dimension mutation*. Modify S_y to another value and repeat step 2 to 4. Hence, patch candidates of different dimensions are generated.

Step 6: *Patch Selection*. The best patch is selected from the good patch candidates based on the following two criteria: 1) the resultant phase error is not sensitive to different defocusing levels; and 2) the resultant phase error is consistently small.

Step 7: *Fringe pattern construction*: Once the best patch is determined, binary fringe patterns can be generated by tiling the best batch together.

Based on this patch-based optimization framework, some improvements have been introduced. For examples, the proposal in [23] modifies the objective function shown in eqn. (2.14) by removing the 3rd harmonics because the 3rd harmonics do not impact the phase measurement. In [18], the objective function is modified by taking both mean square error (MSE) and structural similarity index measure (SSIM) into account. Besides, in its pixel mutation process, pixels are visited group by group, rather than one by one.

Patch-based fringe pattern generation algorithms can significantly reduce the optimization effort. However, since the final fringe patterns are constructed by tiling the patch, it introduces severe harmonic distortion to the measured depth map. A solution is required to address this issue.

2.3.3 Summary of binary defocusing methods

In short, binary defocusing technique is capable to implement real-time 3D measurement and its performance is not impacted by the luminance nonlinearity of a projector. However, the use of binary fringe patterns introduces noise into the measurement and hence degrades the measurement accuracy. Though remedial solutions such as [6], [8], [10]–[12], [15], [17], [18], [21] and [23] have been proposed to increase the accuracy, there is still room for further improvement. Its sensitivity to defocusing extent and fringe frequency is another issue. Patch-based fringe pattern generation algorithms such as [10] and [23] can reduce the optimization effort, but the tiling results carries strong harmonic distortion and it can affect the measurement performance seriously.

2.4 Reversible color-to-grayscale conversion

Reversible color-to-grayscale conversion (RCGC) aims at solving the problem of representing a full color image as a grayscale image without discarding its chrominance information. The idea of RCGC was firstly proposed by Queiroz and Braun [32] to tackle the problem of how to recover a color image from a grayscale image.

State-of-art RCGC algorithms can be categorized into Subband embedding (SE) based RCGC [32–37] and Vector Quantization (VQ) based RCGC [27–31]. Both of them try to compress the chrominance information of a color image significantly so that it can be embedded into a grayscale image. There are two significant issues to be resolved: 1) how to retain the chrominance information under a large compression ratio, and 2) how to avoid obvious visual artifacts in the grayscale image in which the chrominance information is embedded.

Since RCGC involves information embedding, a naive approach is to compress the chromatic information and then embed it as binary data into the grayscale version with a reversible watermarking algorithm such as [67]–[69]. However, this approach does not provide a good performance in practical situations because the color information of a natural image increases with the image size and it is too rich for a reversible data-hiding algorithm to handle. For example, even after a 4:2:0 chrominance subsampling process [70] is performed, the average entropy of the chrominance content of the color images in the Kodak true color image set [71] is 2.9 bits per pixel (bpp), which still significantly exceeds the capacity of state-of-art reversible data hiding algorithms, and hence a further compression is required. The chromatic quality of the reconstructed color image can be degraded remarkably at a high compression ratio. Dedicated RCGC algorithms are hence needed to address this specific issue.

2.4.1 Subband embedding-based RCGC

Subband embedding-based RCGC is a kind of RCGC algorithms in which the chromatic information of the original color image is extracted, downsampled and embedded in the high frequency bands of the luminance plane of the original image. The hidden chromatic information can be extracted from the corresponding frequency bands to reconstruct the color image later. The flowchart shown in Figure 2-5 shows the working principle of SE-based RCGC.

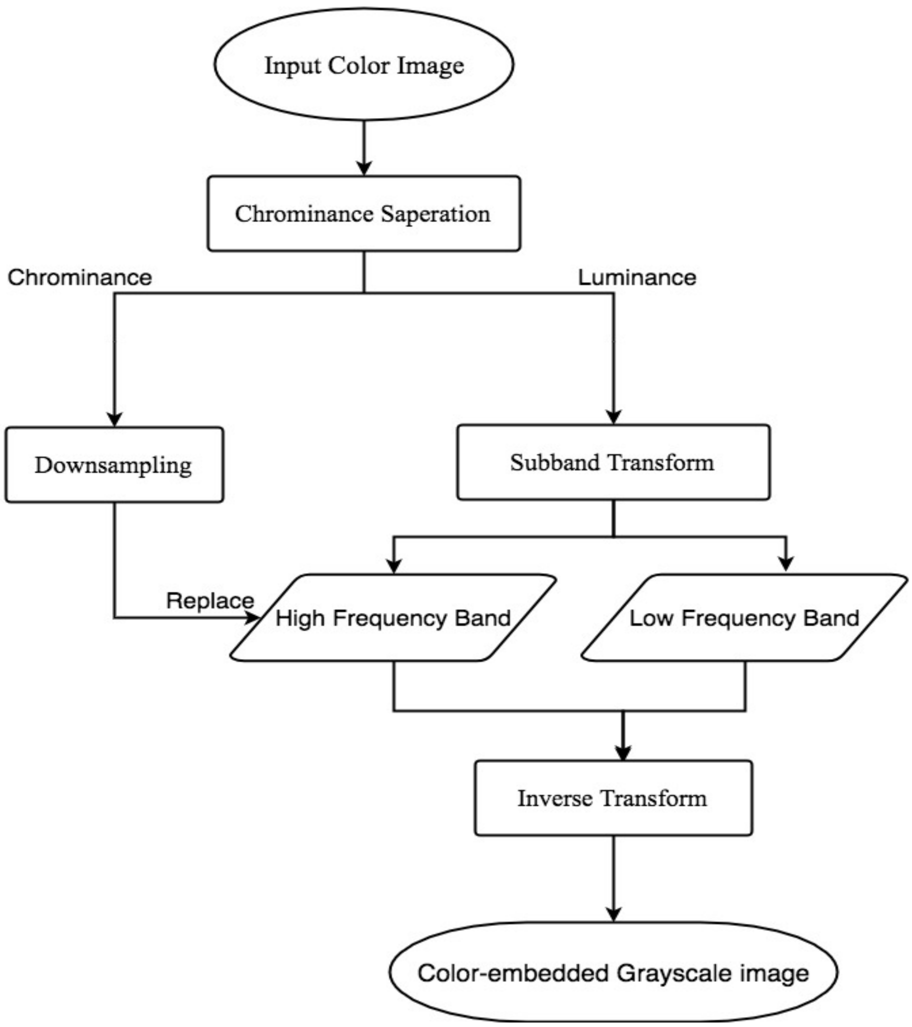


Figure 2-5 Flowchart of SE-based RCGC algorithm

The first SE-based RCGC was proposed by Queiroz and Braun in 2006 [32]. In their approach, the color image is transformed into a luminance-chrominance color space such as YUV or CIELAB. The separated luminance plane Y is then divided into four subbands via a one-level Discrete Wavelet Transform (DWT). The four subbands are named *low-pass*, *vertical high-pass*, *horizontal high-pass* and *diagonal high-pass* bands respectively. The dimension of each subband is only half of that of Y. Accordingly, chrominance planes U and V are downsampled by 2 along each direction and then used to replace subbands *vertical high-pass* and *horizontal high-pass* respectively. The color-embedded grayscale image is obtained by inverse DWT of the modified subbands.

Queiroz and Braun also discussed the impact of printing to this color-embedded grayscale image. Descreening, warping, misregistration and blurring problems may be encountered when printing and scanning a color-embedded grayscale image. To handle these obstacles, they proposed to modify the embedding scheme as follows:

Step 1. Transform the color image to YUV color space to obtain luminance plane Y and chrominance planes U and V.

Step 2. Decompose chrominance planes U and V to four planes U+, U-, V+ and V- as follows.

$$U^+ = (U > 0) \tag{2.16}$$

$$U^- = (U < 0) \tag{2.17}$$

$$V^+ = (V > 0) \tag{2.18}$$

$$V^- = (V < 0) \tag{2.19}$$

Step 3. Transform Y with a two-level DWT to get seven frequency subbands in frequency domain.

$$Y \rightarrow (S_L, S_{h1}, S_{v1}, S_{d1}, S_{h2}, S_{v2}, S_{d2}) \tag{2.20}$$

Step 4. Replace four subbands with downsampled chrominance planes as follows.

$$S_{d1} \leftarrow U^{-},\ S_{h2} \leftarrow U^{+},\ S_{v2} \leftarrow V^{-},\ S_{d2} \leftarrow V^{+} \tag{2.21}$$

The subband replacement process is summarized in Figure 2-6.

Step 5. Take inverse DWT to obtain the color-embedded grayscale image. This image is ready to be printed or faxed.

The recovery of the color image is the reverse of the above steps. During the reconstruction, chrominance planes U and V are obtained by

$$U = |\, S_{h2}| - |S_{d1}| \tag{2.22}$$

$$V = |\, S_{d2}| - |S_{v2}| \tag{2.23}$$

and then upsampled to the size of Y with bilinear interpolation.

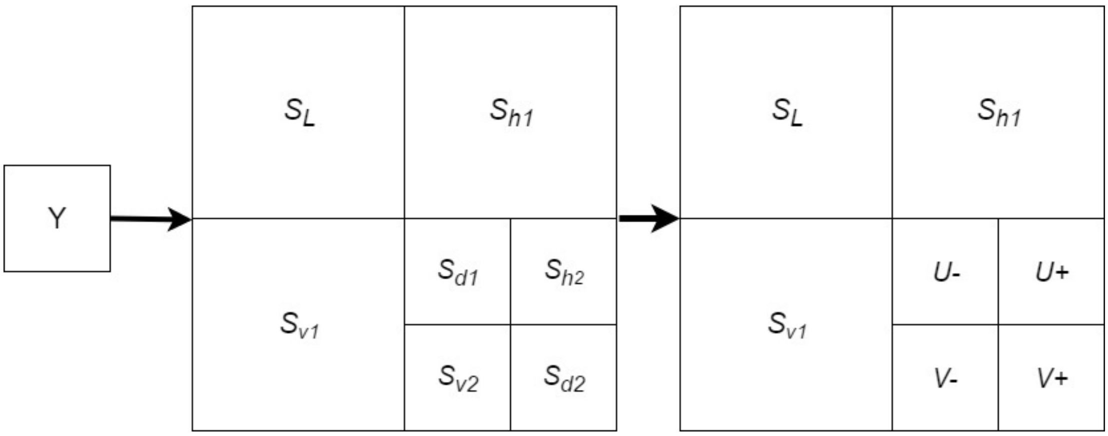


Figure 2-6 Substitution of high frequency band

The idea in [32] was further studied afterwards. Ko et. al. [34] changed the transform to Discrete Wavelet Package Transform (DWPT). They investigated to find out the subbands of the minimal amount of luminance information. Accordingly, the chrominance information is inserted into the less informative subbands to reduce the loss of luminance information.

The chromatic information may be distorted due to the blurring effect of a scanning process. In [35], this problem was tackled by using redundant representation of color information. Specifically, Discrete Cosine Transform (DCT) was applied, and a theoretical study was made to determine bounds for guiding people on how many subbands into which to embed the chrominance.

After being embedded into the luminance plane, the chromatic information is visible as textures in the color-embedded grayscale image. These textures form a highly visible regular pattern in a smooth region because the chromatic information in a smooth region is spatially identical. To alleviate this problem, Horiuchi et. al. [36] presented a modification that partly preserves the high frequency luminance information to reduce the distortion. They also showed a practical application of their approach in color information security.

2.4.2 Vector quantization-based RCGC

Vector quantization-based RCGC algorithms tend to reduce chromatic redundancy with color quantization. In generally, a color-quantized image is visually close to the original color image, and the number of colors in the palette can be adjusted flexibly. As long as the index plane looks like the luminance plane of a color image, it can be interpreted as the grayscale version of the image. Figure 2-7 shows the basic working principle of a VQ-based RCGC algorithm.

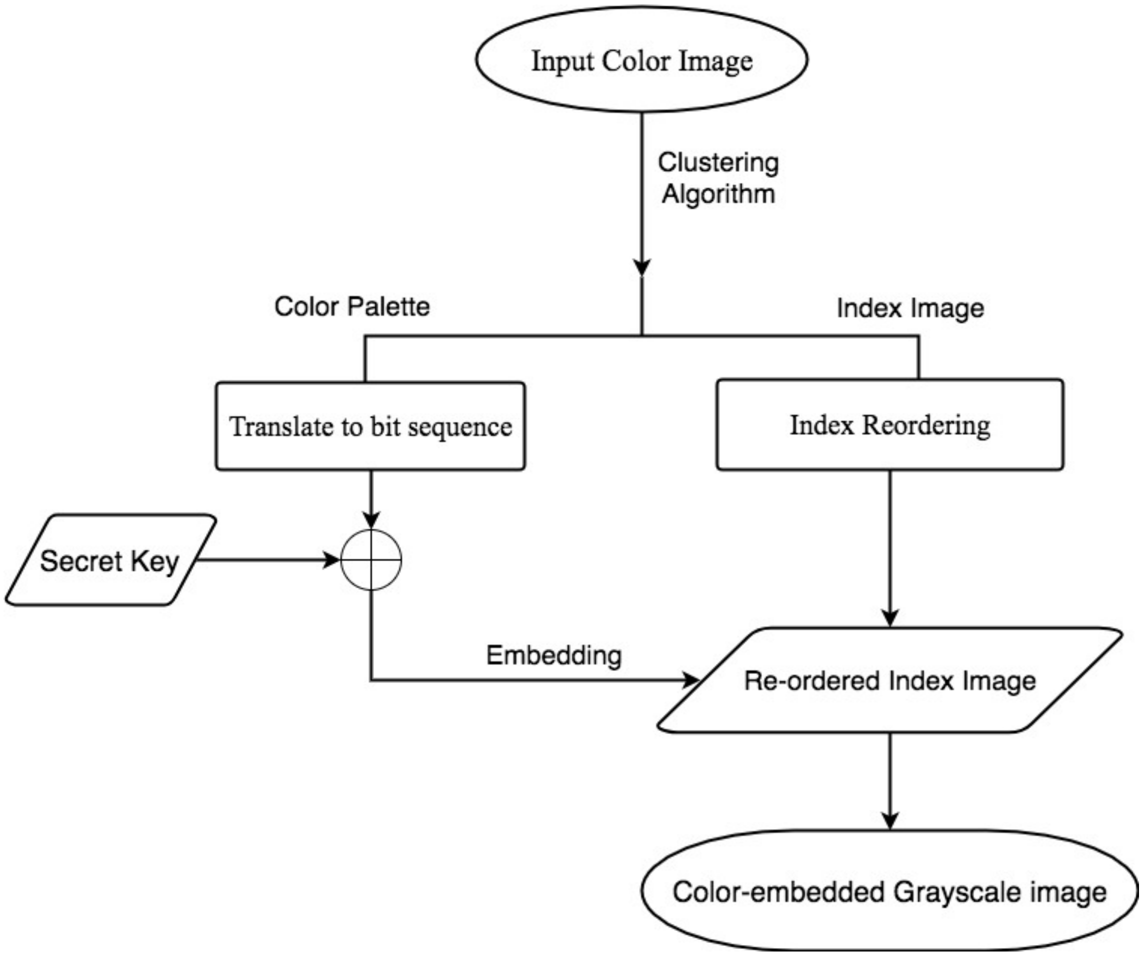


Figure 2-7 Flowchart of VQ-based RCGC algorithm

In [28], Chaumont proposed to generate a color palette by simultaneously minimizing (1) the difference between the color-quantized image and the original image, and (2) the difference between the index image and the luminance plane of the original image. The optimization is performed iteratively in a two-step loop as in a conventional fuzzy c-mean algorithm [72].

A fast method was presented in [33] to re-order the indexes of palette colors according to a function of their color differences and luminances. The color palette is obtained with the popular k-mean clustering method [73]. After index reordering, the color palette is compressed to reduce the bits to be embedded.

Tanaka et. al. introduced a method in [30]. They modified the progress of k-mean clustering by inserting a luminance constraint in each iteration cycle. This constraint forces the index value of a color close to the color's luminance value. Accordingly, the index image can be similar to the luminance plane of the original image.

The algorithm presented in [31] is an improved version of [27] in a way that the number of colors can be adaptively determined. In this algorithm, the color palette is extracted by fuzzy c-mean clustering [72], where the number of colors is determined by the range of existing luminance values in the original image. After the color palette is obtained, the color index is re-ordered by a layer-scanning algorithm. The re-ordering algorithm is a heuristic algorithm that minimizes the errors of both the color-quantized color image and the index image. Finally, the color palette is translated into a bit sequence, protected by a secret key, and embedded into the index image by the least significant bit (LSB) substitution method.

Current VQ-based algorithms generally concern more on the quality of the recovered color image. The color-embedded grayscale image, which is basically the index plane of a color quantization result, can deviate from the luminance plane of the original image remarkably. Besides, their recovered color images generally suffer from color shift and false contour due to its limited number of palette colors.

In the field of color quantization, halftoning techniques have been exploited to alleviate the false contour and color shift in a color-quantized image [74–76]. However, to our best knowledge, halftoning has never been exploited in RCGC to improve the visual quality of the recovered color image, not to mention using halftoning in the way that we introduce in this thesis to shift the noise around the color-embedded grayscale image and the recovered color image.

We note that there are palette generation algorithms (e.g. [75]) proposed to support quantizing a color image with halftoning. However, these algorithms do not take RCGC into account and their generated palettes cannot be used in RCGC. It is because, in these palettes, the index value of a palette color is not linearly correlated with its luminance value. Without this property, it is impossible for an index plane to appear as the luminance plane of the color image.

2.4.3 Summary of RCGC algorithms

In short, state-of-art RCGC algorithms can be classified into either SE-based or VQ-based algorithms. SE-based algorithms replace high frequency subbands with downsampled chrominance planes. The constructed grayscale image is generally blurred since the high frequency luminance content is removed. Besides, the embedded chrominance information appears as visible pattern noise in smooth regions of the recovered color image. VQ-based algorithms train a color palette with a conventional clustering algorithm and then color-quantize a color image with the palette to generate an index plane that appears as a grayscale image. Without the help of halftoning, there are generally color shift and false contour in the recovered color images. Moreover, the index image generated by a VQ-based RCGC algorithm is usually far away from the luminance plane of the original image.

Though halftoning has been proposed to work with color quantization to improve the quality of a color quantized image and this factor has been taken account to develop color palette, it has never been exploited in RCGC to improve the conversion performance and no dedicated color palette generation algorithm has been proposed to support RCGC. RCGC demands a luminance constraint that was not concerned by any conventional color palette generation algorithm before.

Chapter 3.

Real-time 3D shape measurement with octa-level fringe patterns

3.1 Introduction

Chapters 3, 4 and 5 survey the application of halftoning in 3D shape measurement. As mentioned in Chapter 1, binary defocusing algorithms were designed to break the speed bottleneck of 3D shape measurement. However, quantization noise is introduced as the grayscale fringe patterns are replaced by binary images. To address the impact of quantization noise, halftoning algorithms have been applied to this area since 2012 [13], but simply application of state-of-art halftoning algorithms is far from optimal. The following three chapters are devoted to a framework of generating octa-level fringe patterns and further optimization with respect to harmonics in phase domain that has never been addressed in the literature.

This Chapter is organized as follows. In Section 3.2, we formulate the definition of octa-level fringe patterns that simulate the grayscale sinusoidal fringe patterns and explain why these fringe patterns can help improve quality without loss of measurement speed. Then, in Section 3.3, we present an optimization process to further improve the measurement quality. In particular, this optimization is a variant of the patch-based optimization described in [10]. In Section 3.4, simulation and experiment results are provided to evaluate the performance of the proposed algorithm. In Section 3.5, experiment results are presented. Finally, a summary is given in Section 3.5.

3.2 Generation of octa-level fringe patterns

A commercial single chip DLP projector generates full color images by either placing a color wheel or using three individual color light sources. In either approach, three gray level images respectively pass through three color channels (R, G and B) in turns to formulate a full color image. Accordingly, one can produce three different binary patterns, one for each channel, for the projector to project a color halftone image *X*. The luminance channel of *X*, say *L*, can be obtained by

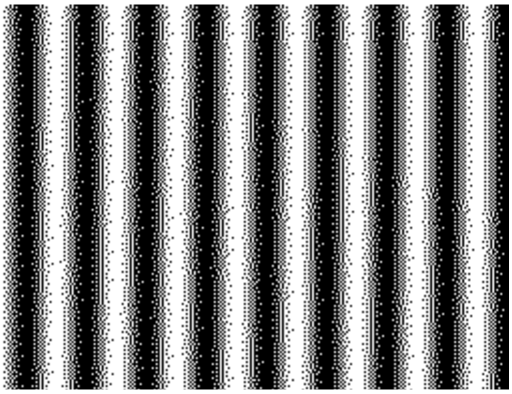
$$L = 0.299B_R + 0.587B_G + 0.114B_B \tag{3.1}$$

where B_R, B_G and B_B are the binary patterns for channel *R*, *G* and *B* respectively.

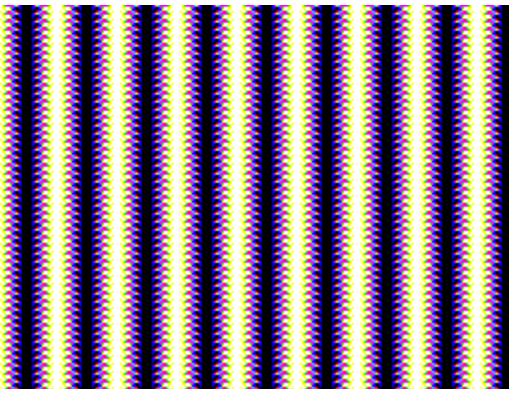
Let $B_c(x, y)$, where $c \in \{R, G, B\}$, be the intensity value of pixel (x, y) of pattern B_c. Since $B_c(x, y) \in \{0,1\}$ for all c, there are altogether $2^3 = 8$ possible intensity levels in *L*. In other words, if we project the color fringe image *X* onto the object and extract the luminance plane of the color image captured by the camera, it will be equivalent to projecting an octa-level fringe pattern onto the object directly.

Note that the actual luminance value associated with a particular color can vary among different projectors in practical situations. However, they can be measured easily though a simple experiment before doing 3D measurements. Eqn. (3.1) can then be adjusted accordingly.

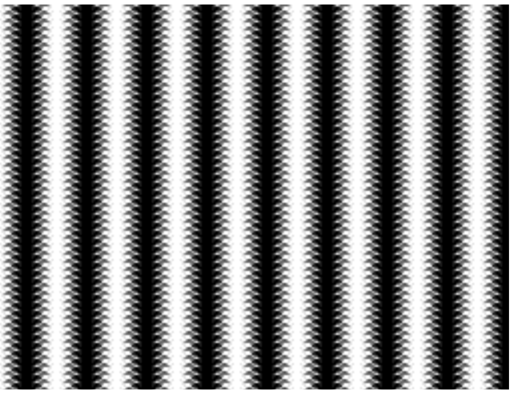
Figure 3-1 illustrates an example which provides a binary fringe pattern, a color halftone fringe pattern and an octa-level fringe pattern that is actually the luminance plane of the color halftone fringe pattern. The octa-level fringe pattern can be considered as a quantization result of a grayscale sinusoidal pattern obtained with a non-uniform 8-level quantizer. Its quantization noise is much lower than a binary pattern



(a)



(b)



(c)

Figure 3-1 (a) a binary fringe pattern that simulates sinusoidal fringe pattern; (b) an color binary fringe pattern that simulates the same pattern; (c) luminance plane of (b).

As compared with binary fringe patterns, the only disadvantage of octa-level fringe patterns is that more preparation effort is required to develop the patterns since it involves the manipulation of three binary planes. However, fringe pattern development is a one-time process and it is done off-line. In practice there is no extra cost for subsequent measurements once the fringe patterns are developed.

Accordingly, octa-level fringe patterns offer the following advantages:

1) Real-time measurement can be supported, as the frame rate can be as high as in the case when binary defocusing technique is used (up to 667 Hz).

2) It is not impacted by projector non-linearity, as only the minimum and maximum intensity levels are used in each color channel.

3) No color-shifting calibration is needed, as there is no restriction on the exact intensity values of the 8 luminance levels. It is different from the case when full-color fringe patterns are used [77].

4) Higher accuracy can be achieved, as octa-level instead of binary fringe patterns are projected.

3.3 Optimization of octa-level fringe patterns

An octa-level fringe pattern can be simply acquired by quantizing a sinusoidal fringe pattern I using a non-uniform quantizer. However, the quantization noise remains a problem although it has been decreased by increasing the number of quantization levels. In previous studies of binary fringe pattern generation, halftoning algorithms such as error diffusion [39] and Bayer dithering [42] were applied to shape the quantization noise. In principle, these halftoning algorithms can also be extended to generate octa-level fringe patterns with a non-uniform quantizer. However, we found that a simple extension of these halftoning algorithms is not able to produce good octa-level fringe patterns. We hence designed an optimization process that is similar to Direct Binary Search (DBS) [78] to optimize an octa-level fringe pattern. The idea stems from the fact that DBS is a powerful halftoning method to produce high-quality halftones by mutating and swapping pixel values one by one.

Consider the case that one wants to generate octa-level fringe patterns S_k for $k \in \{1,2,3\}$ to approximate sinusoidal fringe patterns I_k with a defocused projector. As I_1, I_2 and I_3 are shifted versions of each other, all we need is actually to generate an octa-level pattern S to approximate a sinusoidal fringe pattern I. To achieve this goal, one can generate three binary patterns, say $B_R{}^*$, $B_G{}^*$ and $B_B{}^*$, by minimizing the following cost function:

$$\begin{aligned}\{B_R{}^*, B_G{}^*, B_B{}^*\} &= \arg\min_{\{B_R, B_B, B_G\}} \ J \\ &= \arg\min_{\{B_R, B_B, B_G\}} \left\| I - G^{(t)} \otimes S \right\| \end{aligned} \tag{3.2}$$

where S is constructed with binary pattern B_C for c∈{R,G,B} with eqn.(3.1), $G^{(t)}$ symbolized a $t \times t$ Gaussian low-pass filter that simulates the defocusing effect. The coefficients of the Gaussian filter are given as:

$$G^{(t)}(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{3.3}$$

where σ is the standard deviation of Gaussian distribution which is often set to t/3 to simulate the blurring effect of a defocused projector [10–12, 23, 79], t is the filter size where a larger *t* symbolizes more defocused projector and vice versa.

The optimization is a NP-hard problem as B_C is not continuous. Moreover, S is an 8-level pattern instead of a binary pattern so we cannot simply switch on/off pixel values when doing a search. To solve this problem, we proposed a patch-based optimization algorithm that is modified from [10]. Its detailed procedure is given as below:

Let P be a color patch of size $N_x \times N_y$, where N_x is fixed to be half of the fringe period of the target sinusoidal pattern. $P(x,y)$ denotes the color of pixel (x,y) in patch P. A color is represented as a vector belonging to set $\Omega=\{(r,g,b)|r,g,b\in\{0,1\}\}$, where r, g and b are the values of the red, green and blue components of the color.

Step 1: *Patch initialization*:

Initialize color patch P by randomly assigning a color in Ω for each of its pixel.

Step 2: *Patch optimization*:

Step 2.1: Raster scan the patch and process the patch pixels one by one as follows. For each patch pixel, replace its color with each one of the 8 colors in Ω to form a new patch P', tile patch P' to construct a color pattern Y the size of which is the same as that of I, split the three primary color channels of Y to construct an octa-level pattern S with eqn. (3.1), update patch P to be the patch that provides the minimum value in eqn. (3.2) among the eight tested patches. Step 2.1 is completed when the whole patch is scanned once.

Step 2.2: If the total improvement in step 2.1 is larger than 0.01% in terms of $\|I - G^{(t)} \otimes S\|$, go back to step 2.1. Otherwise the most updated P is the optimal patch of size $N_x \times N_y$.

Step 3: *Fringe pattern generation*:

Construct a full-size octa-level fringe pattern by tiling the optimized color patch and then extracting the luminance plane of the tiling output. This pattern forms fringe pattern I_2. The other two octa-level fringe patterns, namely I_1 and I_3, can be obtained by shifting I_2 spatially by $-2\pi/3$ and $2\pi/3$, respectively.

In our realization, by changing N_y from 4 to 10, we optimized color patches of different sizes with the above procedures. For each patch, a set of octa-level fringe patterns can be derived. In all available sets of octa-level fringe patterns, the best set is selected based on the criteria that its achieved phase root mean square error is small and stable in different defocusing levels. Specifically, phase root mean square error is defined as

$$\Delta\varphi^{(t)} = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(\varphi(x,y) - \varphi_{H^{(t)}}(x,y))^2} \tag{3.4}$$

where $\varphi(x,y)$ is the phase obtained by projecting grayscale sinusoidal fringe patterns and can be calculated with eqn. (2.13), $\varphi_{H^{(t)}}(x,y)$ is the phase obtained by projecting octa-level fringe patterns with a defocused projector that is simulated with a $t\times t$ Gaussian filter, and $M\times N$ is the size of a full-size fringe pattern. In formulation, $\varphi_{H^{(t)}}(x,y)$ is computed as

$$\varphi_{H^{(t)}}(x,y) = \tan^{-1}\left(\frac{\sqrt{3}\left(G^{(t)}\otimes S_1(x,y) - G^{(t)}\otimes S_3(x,y)\right)}{2G^{(t)}\otimes S_2(x,y) - G^{(t)}\otimes S_1(x,y) - G^{(t)}\otimes S_3(x,y)}\right) \tag{3.5}$$

where $G^{(t)}\otimes S_k(x,y)$ denotes the intensity of pixel (x,y) of $G^{(t)}\otimes S_k$ for $k\in\{1,2,3\}$.

To guarantee that the optimized patches can provide a stable performance in different defocusing levels, the best patch selection is formulated as:

$$P_{best} = \arg\min_P \sum_{t=5}^{13}\Delta\varphi^{(t)} \tag{3.6}$$

The optimization is time-consuming, but once the optimal patch is determined, patterns of different sizes can be easily generated by tiling the patch.

3.4 Performance evaluation

The performance of the proposed algorithm is evaluated in this Section. Firstly, in Section 3.4.1, the parameters for generating the octa-level fringe patterns used in our evaluation are provided. Secondly, in Section 3.4.2, some state-of-art binary defocusing algorithms used for comparative studies are introduced. Finally, in Section 3.4.3, the evaluation results are provided and analyzed.

3.4.1 Simulation parameters

Simulation was carried out to evaluate the phase error performance of the proposed defocused octa-level fringe patterns. While optimizing the octa-level fringe patterns, the defocusing effect was modeled as a 5×5 (i.e. t=5 in eqn. (3.3)) Gaussian low-pass filter, with its standard derivation equal to 1/3 of its size. In our first simulation, fringe period is fixed to be 60 pixels. In the other simulations, the performance achieved with various fringe periods and different defocusing levels are thoroughly analyzed.

To convert a phase map to a depth map of an object, it may be necessary to unwrap the phase map. In all our simulations, Goldstein's branch-cut unwrapping algorithm [65] was exploited for this purpose.

3.4.2 Algorithms used for comparison

To our best knowledge, at the moment, we are the only research group that proposes generating octa-level fringe patterns for 3D measurement. There is no octa-level fringe pattern generation algorithm for comparison.

Accordingly, several binary defocusing algorithms [10–12] were selected to serve as references for comparison. In particular, *opt-p* [12] is an algorithm that optimizes binary fringe patterns by minimizing the resulting phase error with respect to the phase map that can be achieved with the ideal continuous tone sinusoidal fringe patterns. *opt-i* [11] optimizes binary fringe patterns by minimizing their errors from the ideal grayscale sinusoidal fringe patterns in the intensity domain. *bpatch* [10] is a patch-based method that utilizes the symmetry and periodicity of a grayscale sinusoidal fringe pattern in the optimization. In our simulations, all these algorithms were realized using the same parameters as described in their corresponding papers.

There are some other algorithms for generating binary fringe patterns. That only the aforementioned algorithms (*opt-p, opti-i* and *bpatch*) are selected for comparison is because they adopt two-dimensional modulation techniques to produce fringe patterns. Their performance is theoretically better as compared with the algorithms that generate patterns with 1D modulation techniques (e.g. [5], [8], [15] and [17]).

Besides, since the fringe patterns of *opt-p, opti-i* and *bpatch* are generated with an optimization scheme, they outperform the fringe patterns generated with the algorithms that do not have an optimization step (e.g.[4], [13] and [16]). In short, *opt-p*, *opti-i* and *bpatch* are good representatives of the state-of-art binary fringe pattern generation methods exploited in 3D measurement.

3.4.3 Simulation results

In the first simulation, phase maps were derived with the fringe patterns obtained with various algorithms. From each of them, a cross-section of the absolute phase error along the x-direction is extracted and plotted in Figure 3-2. The x-direction corresponds to the direction of the sinusoidal variation in a fringe pattern. The plot covers three fringe periods. The absolute phase error is formulated as:

$$|\Delta\varphi(x,y)| = \left|\varphi(x,y) - \varphi'(x,y)\right| \tag{3.7}$$

where $\varphi'(x,y)$ is the (x,y) th pixel of the pixel-wise phase map obtained by projecting defocused binary fringe patterns or octa-level fringe patterns while $\varphi(x,y)$ is that obtained by projecting ideal sinusoidal fringe patterns.

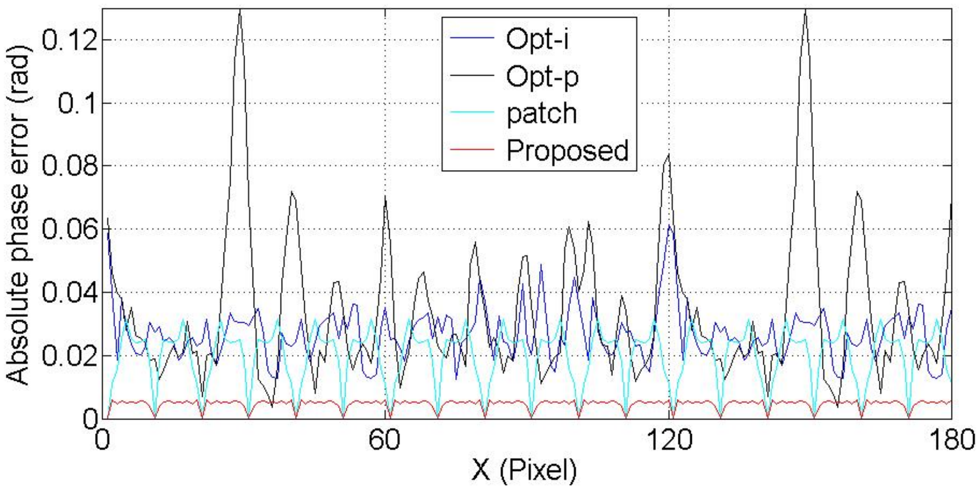
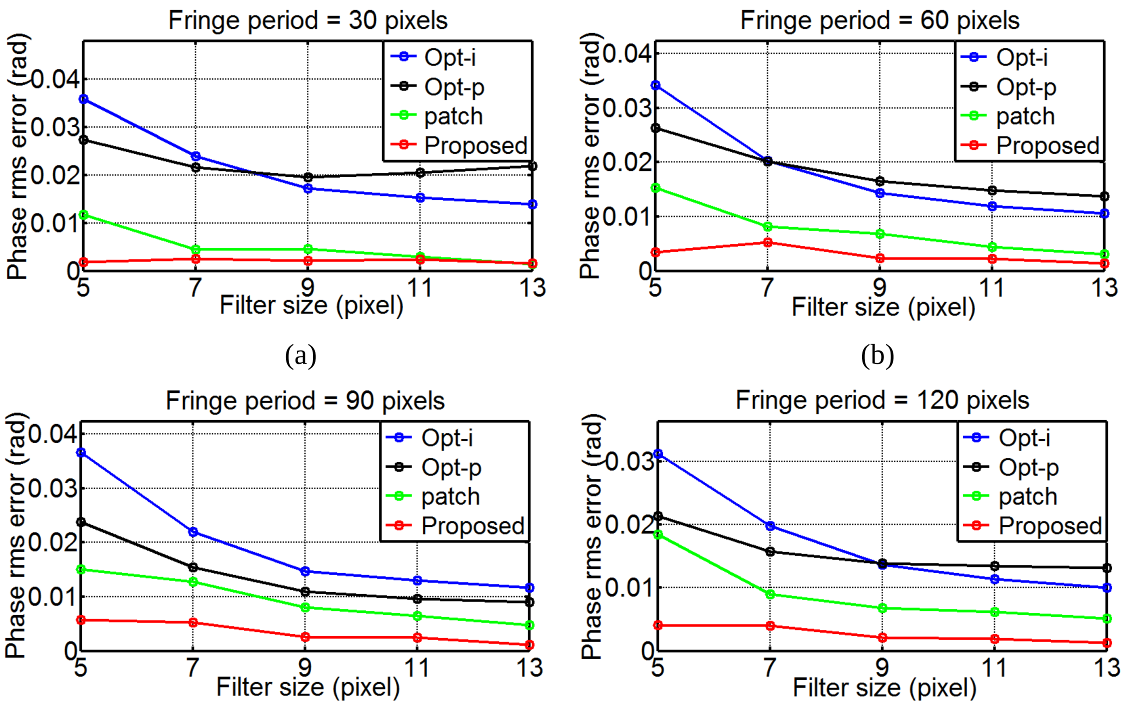


Figure 3-2 Absolute phase error of method *opt-i*, *opi-p*, *bpatch*, and the proposed

From Figure 3-2, one can see that the proposed patterns produce very small phase error with little fluctuation as compared with the binary patterns. Another observation is that the phase errors of *bpatch* and the proposed patterns are periodic, which is due to their patch-based characteristics.

In practical situations, it may not be possible for one to control the defocusing level of a projector precisely. Therefore, in the second simulation, we evaluated the performance of the four methods under different defocusing conditions. Fringe patterns of various fringe periods were involved in this simulation. The performance is measured in terms of phase root mean square (rms) error, which is defined in eqn. (3.4).

The different defocusing levels are simulated with Gaussian filters of sizes 5×5, 7×7, 9×9, 11×11 and 13×13 pixels (i.e. $G^{(t)}$ for t =5, 7, 9, 11 and 13) respectively. Their standard derivations are one third of their filter sizes. As a reminder, we note that all patterns were optimized based on the assumption that the Gaussian filter is of size 5×5, so some evaluation conditions are different from the assumption made during the optimization of the fringe patterns. Figure 3-3 shows the performance curves of different patterns.



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

Figure 3-3 Simulated phase rms errors with different fringe patterns under different defocusing levels when fringe period is (a) 30, (b) 60, (c) 90 and (d) 120 pixels.

From Figure 3-3, one can see that the proposed octa-level patterns are more robust than binary patterns whatever the fringe period is. The performance of the proposed patterns is always better and its advantage is more significant when the defocusing level is low. We note that stronger defocusing implies more low-pass filtering effect, and hence more high frequency noise can be removed. Accordingly, performance of the defocused binary patterns can be closer to the target sinusoidal pattern. This explains why all binary patterns perform better when the defocusing is stronger. In real situation, defocusing cannot be too strong because it lowers the local spatial contrast and hence the noise immunity as well. By increasing the number of luminance levels, the proposed octa-level fringe patterns can approximate sinusoidal patterns well even at a very low defocusing level.

The third simulation is for evaluating the performance of the proposed fringe patterns in measuring a cone-like object. The fringe period is 30 pixels. The defocusing process is again modeled as a 5×5 Gaussian filter with standard derivation equal to 5/3 in this simulation. Figure 3-4 shows the results of measuring a cone with the proposed octa-level patterns and the binary patterns obtained with *opt-i*. By comparing their phase errors shown in Figure 3-4(g) and Figure 3-4(h), one can see the significant improvement achieved by the proposed octa-level fringe patterns. In fact, the maximum phase error is reduced from 0.2211 to 0.0403 rad.

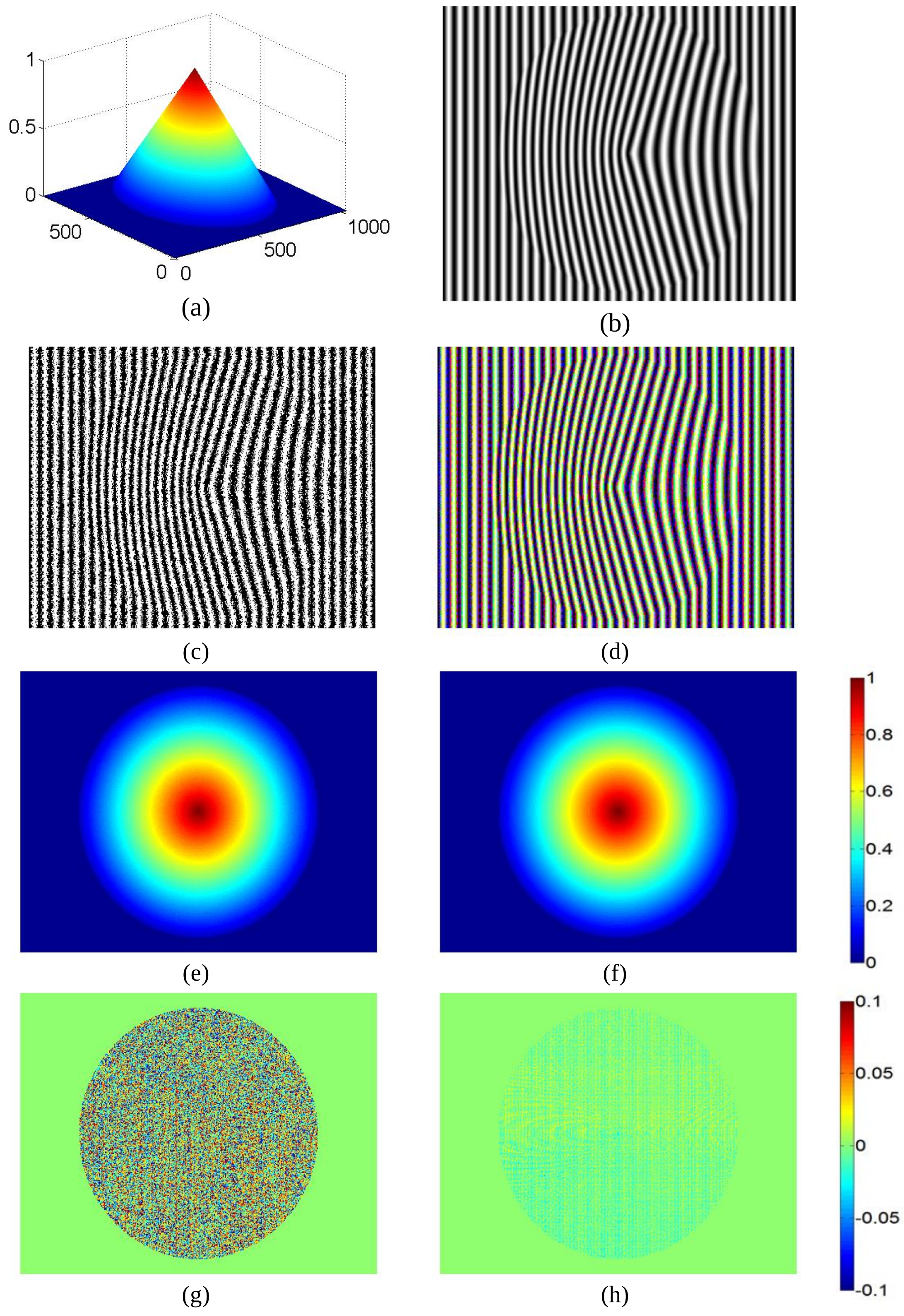


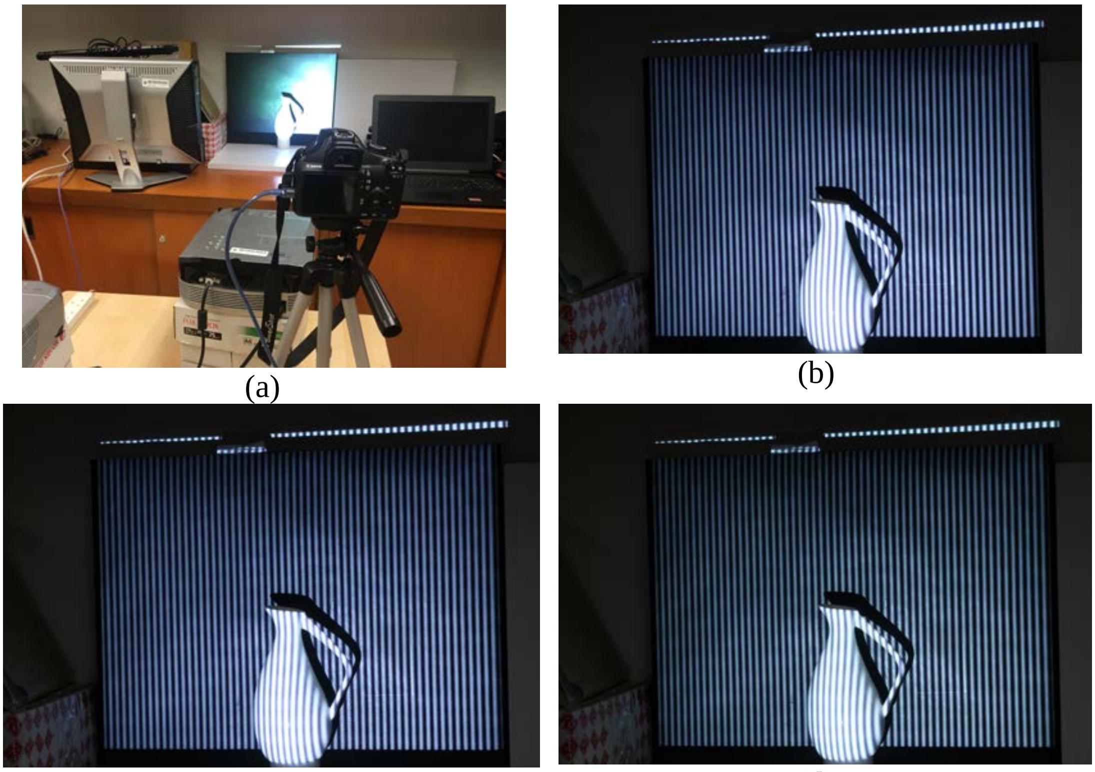
Figure 3-4 Simulation results for measuring a cone. (a) object, (b) projected view of a sinusoidal fringe pattern, (c) projected view of a defocused binary pattern (opt-i), (d) projected view of a defocused octa-level fringe pattern (the proposed), (e) phase map obtained with opt-i, (f) phase map obtained with the proposed octa-level fringe patterns, (g) phase error of (e), (h) phase error of (f).

3.4.4 Experimental results

Besides simulation, we set up a real 3D shape measurement system to evaluate the proposed 3D measuring method. The system consists of a digital-light-processing projector (Texas MP723) and a CCD camera (Canon 400D). The size of a projected fringe pattern is 1024×768 pixels. The fringe period is 18 pixels. The reference plane is placed around 0.5 meters away from the projector. Goldstein's branch-cut unwrapping algorithm [65] was applied to obtain unwrapped phase information. Figures 3-5 (a) shows our experiment system, and figures 3-5 (b), (c) and (d) illustrate one of the captured fringe patterns of grayscale sinusoidal fringe patterns, binary fringe patterns and octa-level fringe patterns, respectively.

Figure 3-6 shows how our octa-level fringe patterns increase the number of quantization levels. With a slight defocused projector, one can observe the signal of *bpatch* [10] includes several quantization levels but the signal of the proposed octa-level fringe pattern (named *cpatch* hereafter) is smooth enough to a sinusoidal signal.

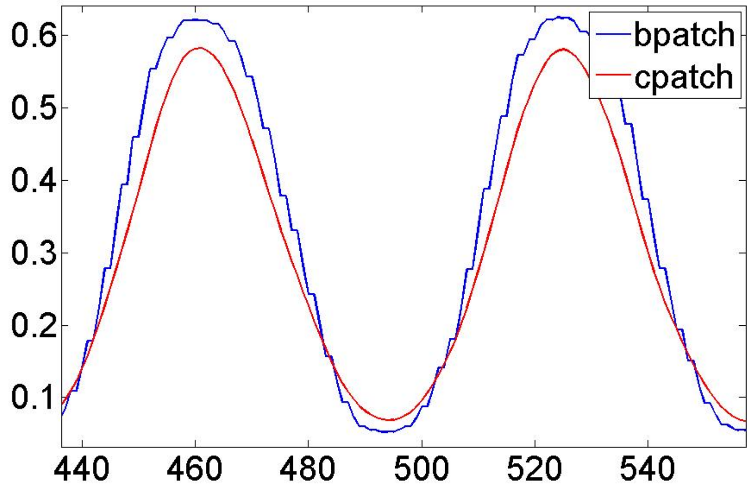
Figures 3-7 (a) and (b) show, respectively, the depth maps of a paper airplane obtained with three-step phase-shifting sinusoidal fringe patterns and the proposed octa-level fringe patterns. Figures 3-8 (b) and (c) show, respectively, the depth maps of a jug with the same fringe patterns. One can observe that the measure quality of the proposed algorithm is approaching the grayscale phase-shifting algorithm. By considering that no gamma calibration is required and real-time realization is feasible when the proposed method is used, the proposed fringe pattern is useful in real world measurement systems.



(a) (b)

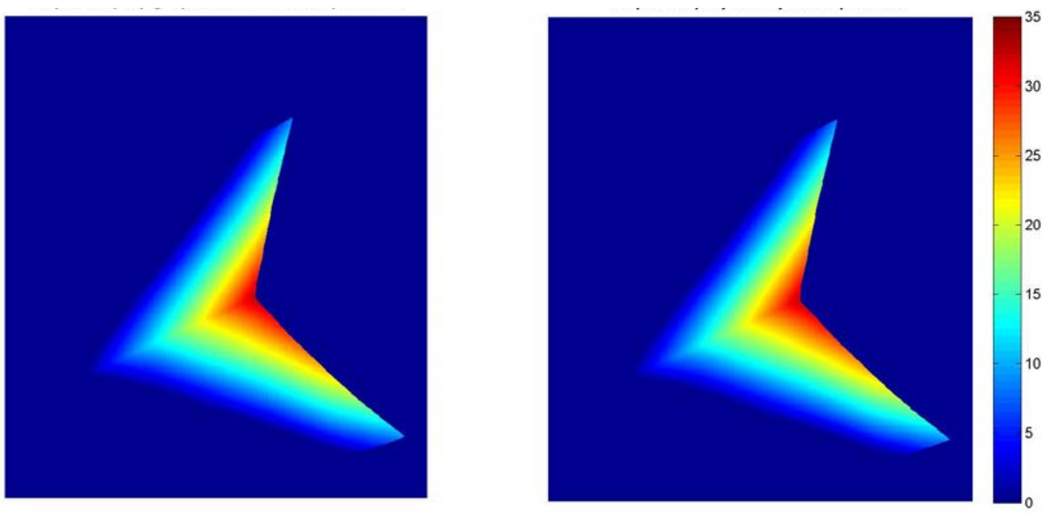
(c) (d)

Figure 3-5 (a) our experiment system; one of the captured images of (b) grayscale fringe patterns, (c) binary fringe patterns and (d) octa-level fringe paterns.



(a)

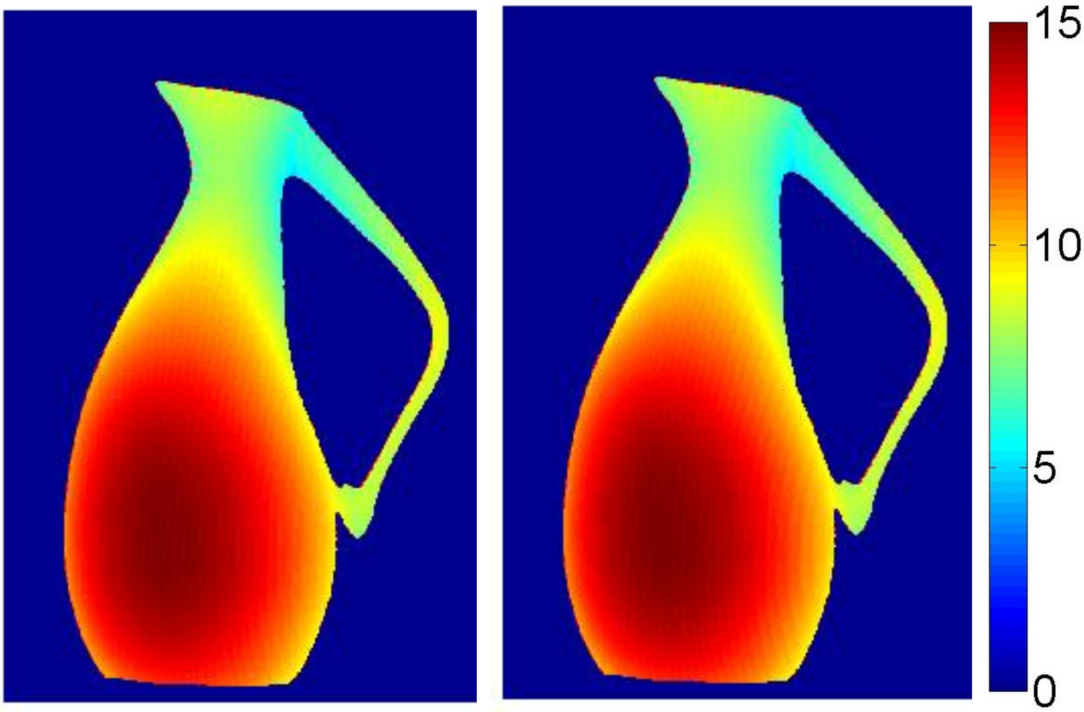
Figure 3-6 Improvement of quantization levels by the proposed octa-level fringe patterns



(a) (b)

Figure 3-7 (a) depth map of a paper airplane obtained with sinusoidal fringe patterns and (b) depth map of a paper airplane obtained with our proposed octa-level patterns.

(a)



(b) (c)

Figure 3-8 (a) picture of a jug; (b) depth map obtained with sinusoidal fringe patterns and (c) depth map obtained with our proposed octa-level patterns.

3.5 Summary

In this Chapter, a framework of generating octa-level fringe patterns is presented. With the help of a defocused projector, octa-level fringe patterns can achieve the same measurement speed and ease the calibration demand during measurements as binary fringe patterns do. In contrast, the quality of octa-level fringe patterns is much higher than binary fringe patterns because of the increased quantization levels. We demonstrated the advantages of the proposed method as compared with the binary fringe projection methods in both simulation and experimental results. The work described in this chapter was published in [25] and contributed to the publication of [26].

For the sake of reference, the method used in this Chapter to generate octa-level fringe patterns is referred to as *cpatch* hereafter.

Chapter 4.

Removing the harmonic distortion introduced by patched-based fringe patterns

4.1 Introduction

In Chapter 3, we present how to generate octa-level fringe patterns that can significantly reduce the noise floor of binary defocusing fringe patterns. The resultant fringe patterns are patch-based, which is similar to those produced in *bpatch* [10]. Although patch-based optimization is computationally efficient, it tends to generate periodic fringe patterns that carry periodic noise patterns in intensity domain. The periodic noise in intensity domain eventually affects the corresponding phase map constructed for the object being measured. As shown in Figure 3-2, periodic phase error can be found in the phase maps generated with the fringe patterns produced with *bpatch* and *cpatch* (i.e. the proposed octa-level fringe pattern generation method introduced in Chapter 3). The periodic phase error introduces regular patterns and contributes strong harmonic noise in the final depth map.

To get an optimized halftone pattern the performance of which is robust to the amount of defocusing, conventional methods optimize halftone patterns under different conditions (e.g. different patch sizes [10], [23] and [25]) and then, from the optimized results, pick the one which is the most robust to defocusing conditions. This pick-the-best-from-the-available approach is passive to some extent and makes the optimization effort grow in multiples.

Though three fringe patterns are required to realize a three-step phase-shifting algorithm in 3D measurement, conventional patch-based binary fringe pattern generation methods generally optimize only one single patch and then tile it to form a full-size fringe pattern. The other two full-size fringe patterns are obtained by shifting the tiling results by $\pm 2T/3$ pixels, where T is the fringe period. In a digital pattern, the shift must be an integer and hence the fringe period is bound to be an integer multiple of 3. This arrangement also introduces a harder constraint for the optimization process to optimize a patch.

In this Chapter, we proposed a different optimization framework which generates multiple patches for tiling so that the periodicity of phase error can be effectively suppressed. The contribution of this work includes:

1) It is able to generate aperiodic fringe patterns the produced phase error of which carries almost no harmonic distortion in the direction along which the fringe patterns provide a sinusoidal magnitude variation.
2) It is capable to generate multilevel fringe patterns of arbitrary fringe period.
3) It releases a constraint for the optimization and theoretically it is able to achieve a better optimization result.
4) It takes proactive action to find optimized patches that are robust to defocusing extent.

The organization of this Chapter is as follows. In Section 4.2, we formulate an alternative objective function that allows us to achieve the aforementioned contributions by optimizing it. The realization of the optimization is presented in Section 4.3. Simulation and experimental results for performance evaluation are provided in Section 4.4. Finally, a brief summary is given in Section 4.5.

4.2 Formulation of the optimization problem

Obviously, we need 3 octa-level fringe patterns, say L_1, L_2 and L_3, to approximate sinusoidal fringe patterns S_1, S_2 and S_3 respectively with the defocusing method such that the approximation error in phase domain is minimized. Besides the minimum error criterion, we would also like to achieve the following criteria:

1. The approximation performance is not sensitive to amount of defocusing.
2. The octa-level fringe patterns of any desirable sizes can be flexibly and easily constructed on site whenever necessary.
3. The octa-level fringe patterns carry no low frequency harmonics and noise.

To achieve these goals, we propose to optimize two sets of three octa-level patch patterns, each of which is constructed with 3 binary patch patterns based on the color to luminance conversion defined in Chapter 3, by taking all the aforementioned criteria into account. Once they are obtained, by randomly tiling the two sets of patch patterns spatially, one can construct three aperiodic octa-level fringe patterns of any desirable size whenever necessary to do 3D shape measurement.

The success relies on how to optimize the two groups of octa-level patch patterns such that they can seamlessly connected to each other. Let $P_{(s,k)}$ be the k^{th} octa-level patch pattern of set $s\in\{0,1\}$, where $k\in\{1,2,3\}$. In our approach, for each k, we tile patch patterns $P_{(1,k)}$ and $P_{(2,k)}$ as shown in Figure 4.1 to form an octa-level pattern, P_k, in which all possible neighboring combinations of $P_{(1,k)}$ and $P_{(2,k)}$ are included, and then optimize P_k for $k\in\{1,2,3\}$ together to minimize a cost function. Note that the occurrences of connection combinations $P_{(1,k)}P_{(1,k)}$, $P_{(1,k)}P_{(2,k)}$, $P_{(2,k)}P_{(1,k)}$ and $P_{(2,k)}P_{(2,k)}$ are identical in P_k and hence there should be no bias to favor a particular connection combination after the optimization.

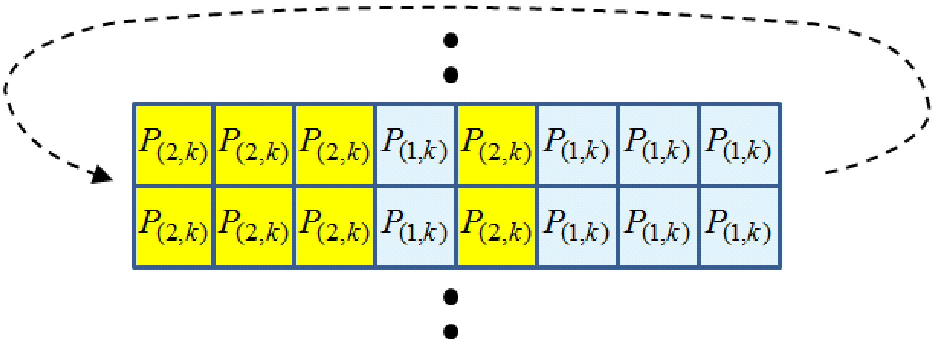


Figure 4-1 A circular connected octa-level pattern with two octa-level patches for optimization.

Let H be a Gaussian low pass filter that models the defocusing effect of a projector. The phase map obtained with the defocused octa-level fringe patterns P_k for $k\in\{1,2,3\}$ is given as

$$\varphi_{S^{(t)}}(x,y)=\tan^{-1}\left(\frac{\sqrt{3}\left(G^{(t)}\otimes S_1(x,y)-G^{(t)}\otimes S_3(x,y)\right)}{2G^{(t)}\otimes S_2(x,y)-G^{(t)}\otimes S_1(x,y)-G^{(t)}\otimes S_3(x,y)}\right) \quad (4.1)$$

where $G^{(t)}\otimes S_k(x,y)$ denotes the intensity value of the $(x,y)^{\text{th}}$ pixel of $G^{(t)}\otimes S_k$ for k=1,2,3. The approximation error in phase domain can hence be defined as

$$\Delta\varphi^{(t)}=\sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(\varphi(x,y)-\varphi_{S^{(t)}}(x,y)\right)^2} \quad (4.2)$$

where $M\times N$ is the size of P_k, $\Delta\varphi^{(t)}$ is actually the phase root mean square (rms) error achieved with the fringe patterns when the defocusing process can be modeled as a Gaussian low-pass filter defined in eqn. (3.3). It is considered as a good measurement of approximation quality in [6], [10]–[12], [18], [23], [25] and [80].

Different amount of defocusing should be taken into account during the optimization such that the approximation performance can be robust to it. In this work, we model slight defocusing and severe defocusing, respectively, with a 5×5 Gaussian filter and an 11×11 Gaussian filter with their standard derivations equal to

1/3 of their sizes. Octa-level patterns P_k for $k \in \{1,2,3\}$ are then optimized in parallel to minimize cost function

$$J = \Delta\varphi^{(5)} + \Delta\varphi^{(11)} \tag{4.3}$$

The two defocusing levels serve as the upper and the lower bounds of a range of defocusing levels. By taking them into account when defining objective function (4.3), the optimization considers both the phase rms error and the robustness to defocusing levels. In contrast to those conventional fringe pattern generation methods (e.g. [6], [10], [18] and [23]) that adopt the pick-the-best-from-the-available approach to passively tackle the robustness issue, we optimize patches proactively to solve the problem.

Recall that P_k is constructed with patch patterns as shown in Figure 4.1 and $P_{(s,k)}$ is actually the luminance plane of a color patch whose red, green and blue channels (denoted as $B_{R(s,k)}$, $B_{G(s,k)}$ and $B_{B(s,k)}$ hereafter) are all bi-level patches. Optimizing P_k for $k \in \{1,2,3\}$ is hence equivalent to solving the following optimization problem:

$$\min_{\substack{B_{R(s,k)}, B_{G(s,k)}, B_{B(s,k)} \\ for\ s \in \{1,2\} and\ k \in \{1,2,3\}}} J \tag{4.4}$$

subject to the constraints that P_k is constructed with $P_{(s,k)}$ for s=1,2 as shown in Figure 4-1 and that

$$P_{(s,k)} = 0.299 B_{R(s,k)} + 0.587 B_{G(s,k)} + 0.114 B_{B(s,k)} \tag{4.5}$$

Notably, the optimization procedures are different from other patch-based optimization schemes (c.f. [6], [10], [18], [23] and *cpatch*) as follows:

1) We develop two sets of patches instead of one single patch such that one can construct full-size fringe patterns with them to eliminate periodic phase error and improve measurement quality.
2) We explicitly optimize three different patches belonging to each set while the conventional approaches optimize only one single patch and then shift it by ±1/3 period to generate the other two patches, which removes the constraint that the fringe period must be an integer multiple of 3.
3) We explicitly take different defocusing conditions into account when constructing the objective function (in eqn. (4.3)) for optimization so that the optimized patterns are automatically robust to defocusing extent.

4.3 Realization of the optimization

Let $C_{(s,k)}$ for $k \in \{1,2,3\}$ and $s \in \{1,2\}$ be 6 color patches of size $N_x \times N_y$ each, where N_x is fixed to be the fringe period of the target sinusoidal pattern. $C_{(s,k)}(x,y)$ denotes the color of pixel (x,y) in patch $C_{(s,k)}$. A color is represented as a vector belonging to set $\Omega=\{(r,g,b)|r,g,b \in \{0,1\}\}$, where r, g and b are the binary values of the red, green and blue components of the color. Octa-level patch $P_{(s,k)}$ is the luminance plane of $C_{(s,k)}$.

Step 1. *Initialization of* $P_{(s,k)}$: Generate grayscale sinusoidal fringe patterns S_1, S_2 and S_3 of size $8N_x \times 5N_y$ each. Serpentine scan S_k for $k \in \{1,2,3\}$ separately. For each scanned pixel, quantize its intensity value to the nearest luminance value of the colors in Ω and then diffuse the quantization error with the diffusion filter suggested in [39]. Note the quantization is a vector quantization and this octa-level error diffusion scheme is shown in Figure 4-

2. After the last pixel is processed, chop two connected $N_x \times N_y$ segments from the quantization result of S_k to form the initial versions of $P_{(1,k)}$ and $P_{(2,k)}$ respectively.

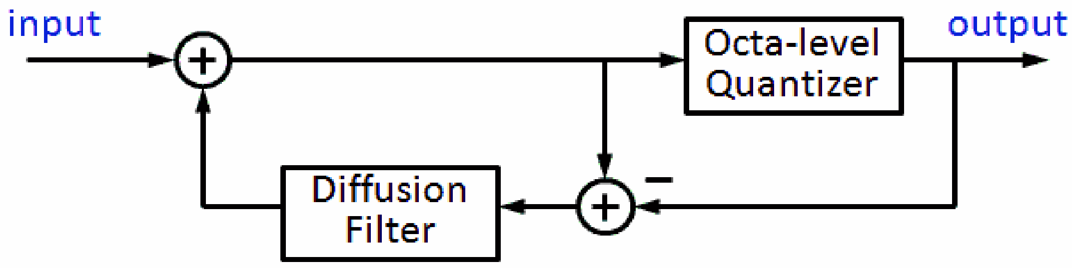


Figure 4-2 Scheme of octa-level error diffusion

Step 2. *Refining* $P_{(1,k)}$: Fix $P_{(2,k)}$ as its most updated version in this step. Raster scan patches $P_{(1,k)}$ for all k in parallel at the same pace and process their pixels sequentially according to the scanning order. Assume that the pixel location being scanned is (x,y). Let $P_{(1,k)}(x,y)$ be the luminance value of the $(x,y)^{th}$ pixel of $P_{(1,k)}$. Since $P_{(1,k)}$ is an octa-level pattern for each k, there are 8^3=512 possible value combinations of $P_{(1,k)}(x,y)$ for $k \in \{1,2,3\}$. For each combination, construct a candidate patch set $\Lambda=\{P_{(1,k)}|k=1,2,3\}$, in which all other pixels of $P_{(1,k)}$ remain the same as the most updated $P_{(1,k)}$, and tile $P_{(1,k)} \in \Lambda$ with the fixed $P_{(2,k)}$ as shown in Figure 4-1 to form a set of fringe patterns P_1, P_2 and P_3. Among all 512 candidate sets of $\{P_{(1,k)}|k=1,2,3\}$, the one used to construct the fringe patterns that minimizes J is the newly updated set of $P_{(1,k)}$ for all k. It continues until all pixels are scanned and processed.

Step 3. *Refining* $P_{(2,k)}$: Do step 2 again but exchange the roles of $P_{(1,k)}$ and $P_{(2,k)}$.

Step 4. *Termination analysis*: If the total improvement in steps 2 and 3 is larger than 0.01% in terms of J, go back to step 2. Otherwise the most updated $P_{(1,k)}$ and $P_{(2,k)}$ are considered as the optimal patches.

Step 5. *Finalizing Fringe patterns*: Randomly tile optimized patches $P_{(1,k)}$ and $P_{(2,k)}$ horizontally to form a patch row and then repeat the patch row vertically to form full-size octa-level fringe patterns L_k for $k \in \{1,2,3\}$ under the condition that L_1, L_2 and L_3 share the same random tiling pattern.

The optimization is time-consuming, but it is offline. Once the optimal patches $P_{(1,k)}$ and $P_{(2,k)}$ are determined, fringe patterns of different sizes can be easily generated by randomly tiling the patches.

4.4 Performance evaluation

This Section provides some simulation results and experimental data for evaluating the performance of the fringe pattern generation method presented in this Chapter.

4.4.1 About the simulations

Simulation was carried out to evaluate the phase error performance of the proposed defocused octa-level fringe patterns. Different from the previous optimization algorithms [10–12, 18, 21, 23, 25], the fringe patterns developed with the approach introduced in this Chapter is not optimized for a particular defocusing level. Instead, the objective function of the optimization is defined based on two defocusing levels which serve as the bounds of a range of defocusing levels.

During the optimization process in our simulation study, these two defocusing levels were simulated with a 5×5 Gaussian low-pass filter and an 11×11 Gaussian low-pass filter, with their standard derivations equal to 1/3 of their sizes. Weighting of the two terms are set to be equal. The optimization process terminates once the marginal improvement in one iteration cycle is smaller than 0.01% in terms of cost function (4.3).

Octa-level fringe patterns of various fringe periods are generated to evaluate the performance. In particular, patterns of fringe periods that are not integer multiply of three are also generated. Fringe patterns of these periods cannot be generated by conventional patch-based optimization algorithms [6, 10, 18, 23, 25].

Several representative binary defocusing algorithms [10–12] as well as the method discussed in Chapter 3 [25] were implemented for comparison. They are referred to as *opt-p* [12], *opt-i* [11], *bpatch* [10] and *cpatch* [25] in this Section. All these algorithms were implemented using the same parameters as suggested in their corresponding papers.

In our simulations, the defocusing process was modeled as a 5×5 Gaussian filter with its standard derivation equal to 5/3 unless else specified. Accordingly, fringe patterns *opt-p*, *opt-i*, *cpatch* were optimized for this defocusing condition while fringe patterns *bpatch* and the proposed were optimized to handle various defocusing conditions as in their original designs.

4.4.2 Simulation results

In the first simulation, the phase maps obtained with various algorithms were derived and their mean absolute phase errors were plotted in Figure 4-3. The period of the target sinusoidal fringe pattern is 60 pixels, and the plot covers two fringe periods. The mean absolute phase errors are computed as

$$|\Delta\varphi|_M(x) = \frac{1}{K_y}\sum_y \left|\varphi(x,y) - \varphi'(x,y)\right| \tag{4.6}$$

where $\varphi(x,y)$ and $\varphi'(x,y)$ are, respectively, the $(x,y)^{\text{th}}$ pixel values of phase maps φ and φ', and $K_y$is the total number of pixels along y-direction. Phase maps φ and φ' are, respectively, obtained when grayscale fringe patterns and the evaluated fringe patterns are used. Mean absolute phase error is more reliable as it is less fluctuated.

As shown in Figure 4-3, the phase errors achieved with octa-level fringe patterns (*cpatch* and the proposed) are reduced significantly as compared with those achieved with binary fringe patterns (*opt-p*, *opt-i* and *bpatch*). As compared with *cpatch*, the mean absolute phase error of the proposed fringe patterns is aperiodic.

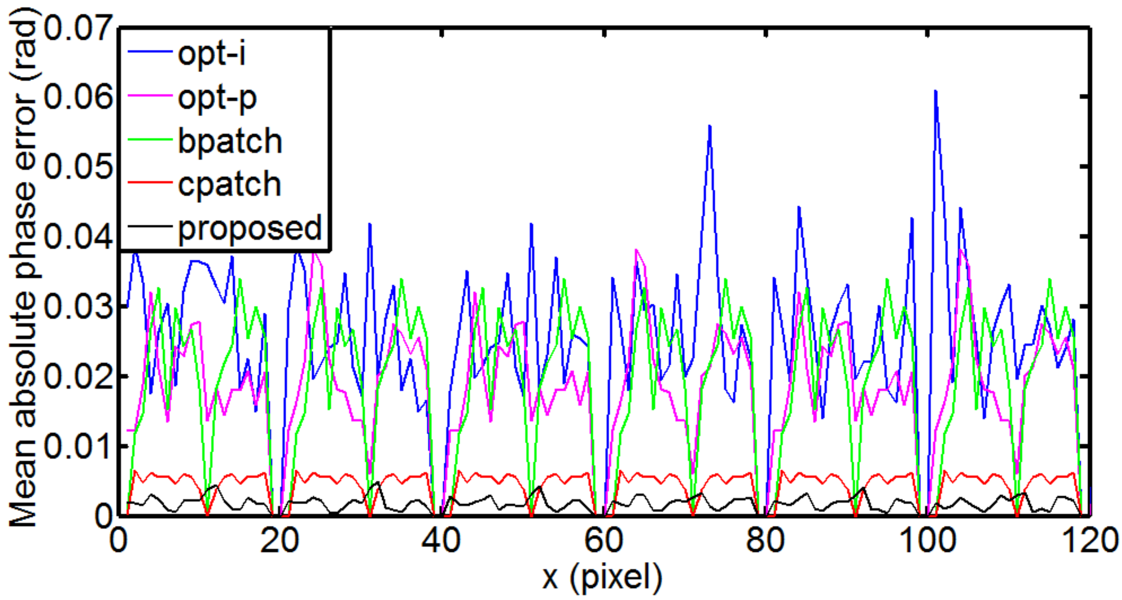


Figure 4-3 Mean absolute phase errors achieved with *opt-i*, *opi-p*, *bpatch*, *cpatch* and the proposed

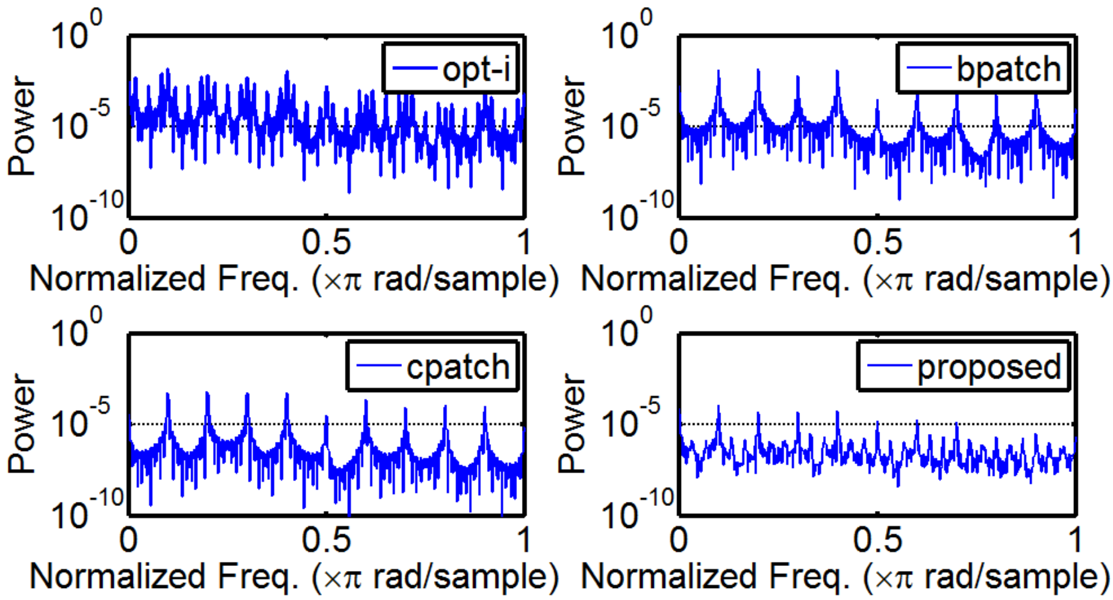


Figure 4-4 Power spectral densities of the phase errors associated with *opt-i*, *bpatch*, *cpatch* and the proposed

For further evaluating the harmonics in phase domain, Figure 4-4 shows the 1D power spectral densities of the phase errors achieved with different fringe patterns. The spectrum associated with *opt-p* is not included because it is similar to the one associated with *opt-i*.

The total noise energy of *opt-i* is the largest. Strong harmonic distortion can be found in the plots of *bpatch* and *cpatch* due to their patch-based optimization schemes. In contrast, the phase errors associated with *opt-i* and the proposed are more like white noise. The noise floor of ours is much lower than *opt-i* and *bpatch*.

Figure 4-5(a) shows the performance achieved with the evaluated fringe patterns in terms of phase rms error when their fringe periods vary from 30 to 120 pixels. Note that the fringe periods of other evaluated fringe patterns must be an integer multiple of 3 while the proposed method do not have this limitation. One can see that the proposed fringe patterns perform well consistently when the fringe period changes. As expected, octa-level fringe patterns (*cpatch* and the proposed) perform better than binary ones (*opt-i*, *opi-p*, *bpatch*). The difference between *cpatch* and the proposed fringe patterns shows the advantage of using fringe patterns with less harmonic distortion.

The same set of fringe patterns evaluated to produce Figure 4-5(a) were also evaluated under different defocusing conditions to investigate whether their performances are robust to defocusing. In our study, different amount of defocusing is achieved by filtering the fringe patterns with a $t \times t$ Gaussian filter with its standard derivation equal to $t/3$, where $t \in \{7,9,11\}$. The simulation results shown in Figures 4-5(b)-(d) verify that the performance of the proposed fringe patterns is robust to defocusing.

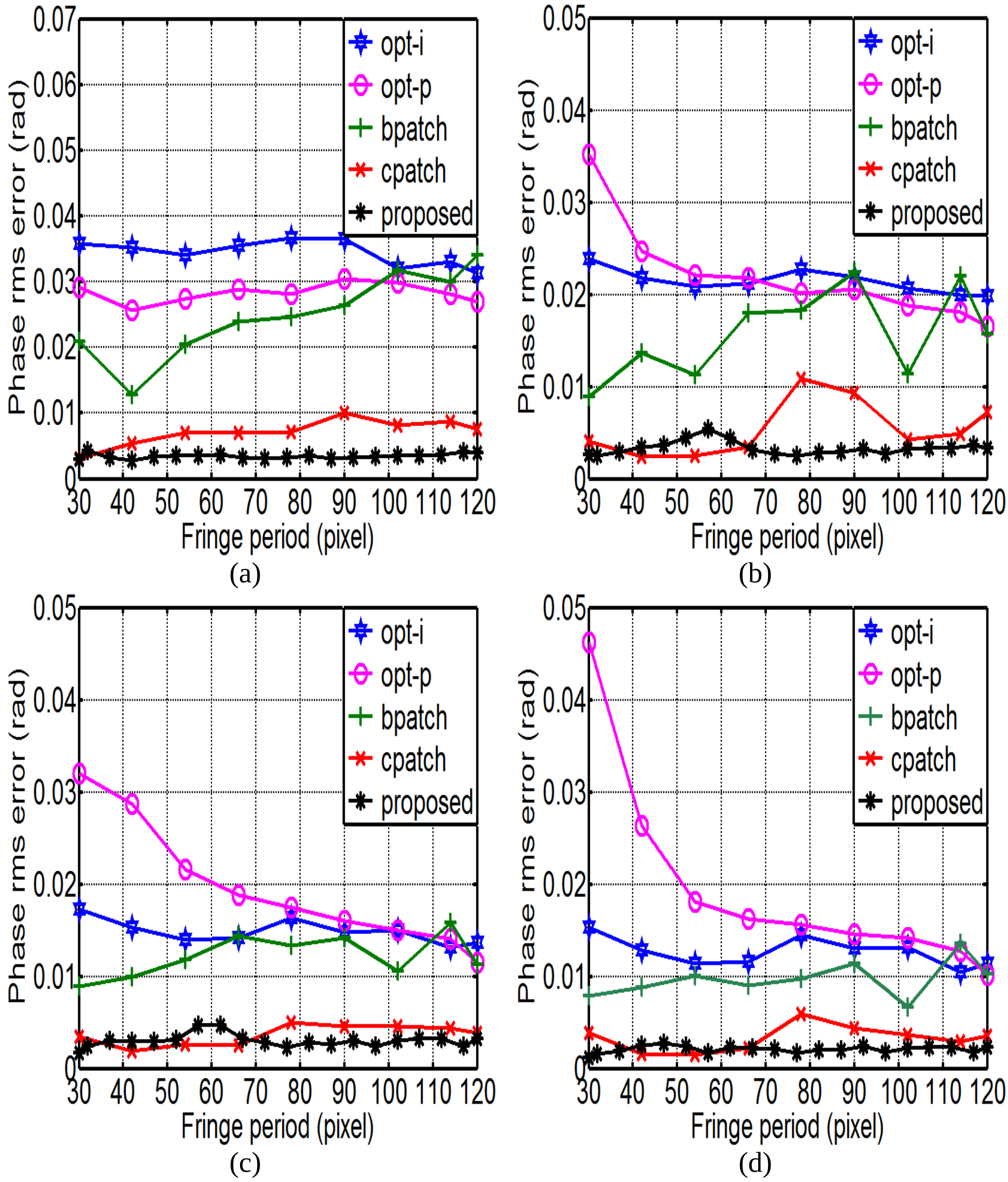


Figure 4-5 Simulated phase rms errors achieved with different fringe patterns of different fringe periods when the defocusing level is simulated by a Gaussian filter of size (a) 5×5, (b) 7×7, (c) 9×9, or (d) 11×11 pixels

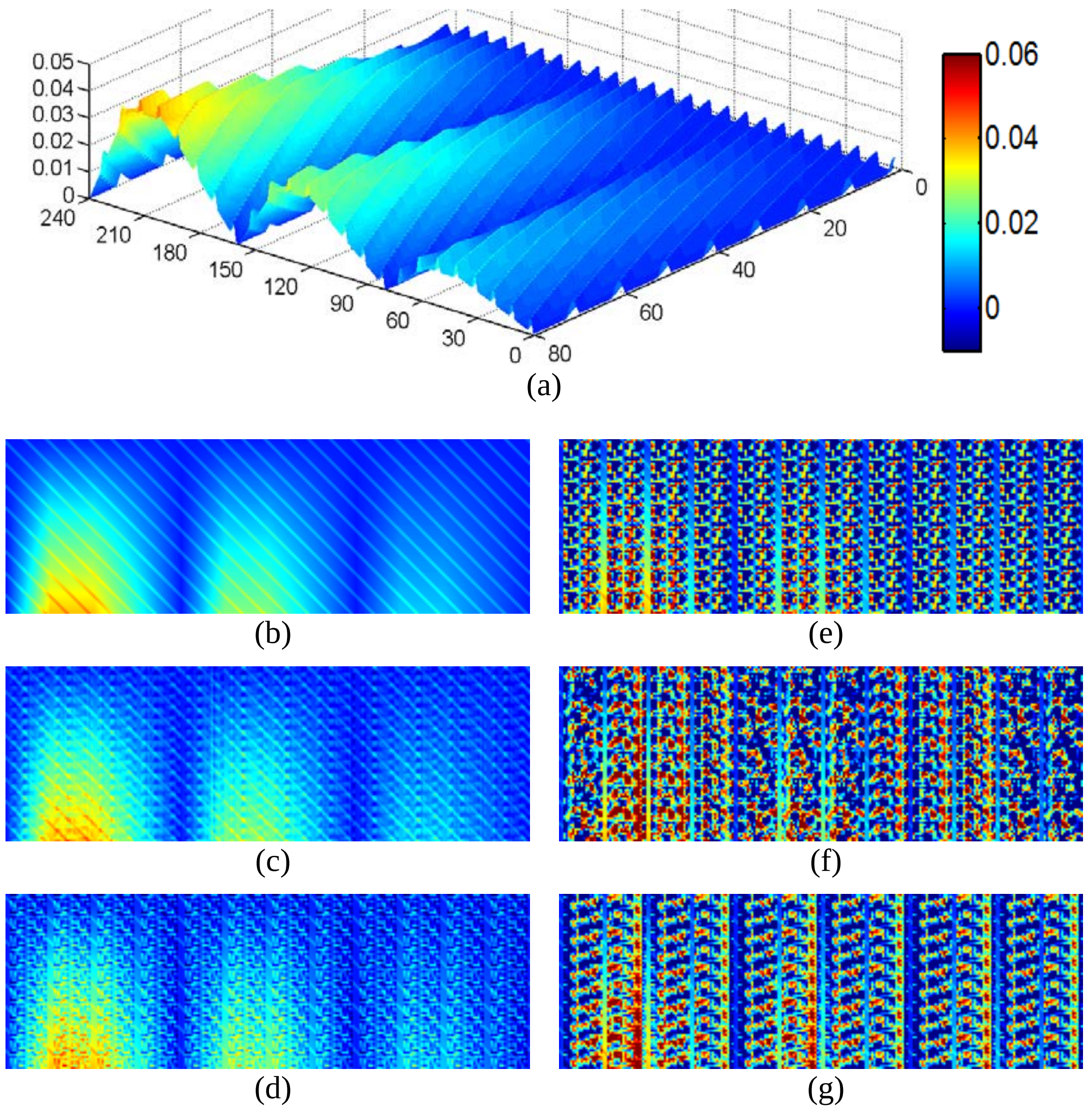


Figure 4-6 Simulation results for measuring an object. (a) 3D plot of the object; (b) ideal unwrapped phase map; and unwrapped phase maps obtained with (c) proposed, (d) *cpatch*, (e) *bpatch*, (f) *opt-i* and (g) *opt-p*.

We also evaluated the algorithms by projecting the fringe patterns onto a complex 3D object. The simulation results are shown in Figure 4-6. In the simulation, the common fringe period of all fringe patterns is 60 pixels, and Goldstein's branch cut unwrapping algorithm [65] was exploited to obtain unwrapped phase information. The defocusing process is modeled as a 5×5 Gaussian filter with standard derivation equal to 5/3. Obviously, the unwrapped phase maps obtained with octa-level fringe patterns are more accurate. Note that the gain is at no cost to a certain extent as only a binary pattern is manipulated in each color channel. After suppressing the harmonic distortion, the unwrapped phase map obtained with our fringe patterns can preserve the parallel ridges much better than the one obtained with *cpatch*.

4.4.3 Experimental results

We also set up a real 3D shape measurement system to evaluate the proposed algorithm. The system consists of a digital-light-processing projector (Texas MP723) and a CCD camera (Canon 400D). The size of a projected fringe pattern is 1024×768 pixels. The fringe period is 18 pixels. The reference plane is placed around 0.5 meters away from the projector. Goldstein's branch cut unwrapping algorithm [65] was applied to obtain unwrapped phase information.

The object to be measured is shown in Figure 4-7(a). Figure 4-7(b) illustrates the ideal depth map which was obtained by grayscale nine-step phase-shifting algorithm [81]. To tackle the gamma nonlinearity of the projector, active gamma correction [66] was done before the grayscale sinusoidal patterns were projected. No gamma correction is needed for projecting binary or octa-level fringe patterns.

Figures 4-7(c), (d), (e) and (f) show, respectively, the depth maps (unit in cm) obtained with *bpatch*, *opt-i*, *cpatch* and the proposed aperiodic octa-level fringe patterns. All are similar to the depth map obtained with ideal sinusoidal fringe patterns (shown in Figure 4-7(b)). However, their depth rms errors are, respectively, 0.0324 cm, 0.0334 cm, 0.0252 cm and 0.0175 cm. The proposed fringe patterns perform better than binary patterns. Remind that no gamma calibration for the projector is required and real-time realization is feasible when the proposed method is used.

To evaluate the improvement of harmonic elimination, we conducted another experiment and plotted the power spectral densities of the phase errors associated with *bpatch*, *cpatch* and the proposed fringe pattern *cpatch1D* in Figure 4-8. It can be clearly noticed that *bpatch* and *cpatch* produced severe harmonics but *cpatch1D* did not.

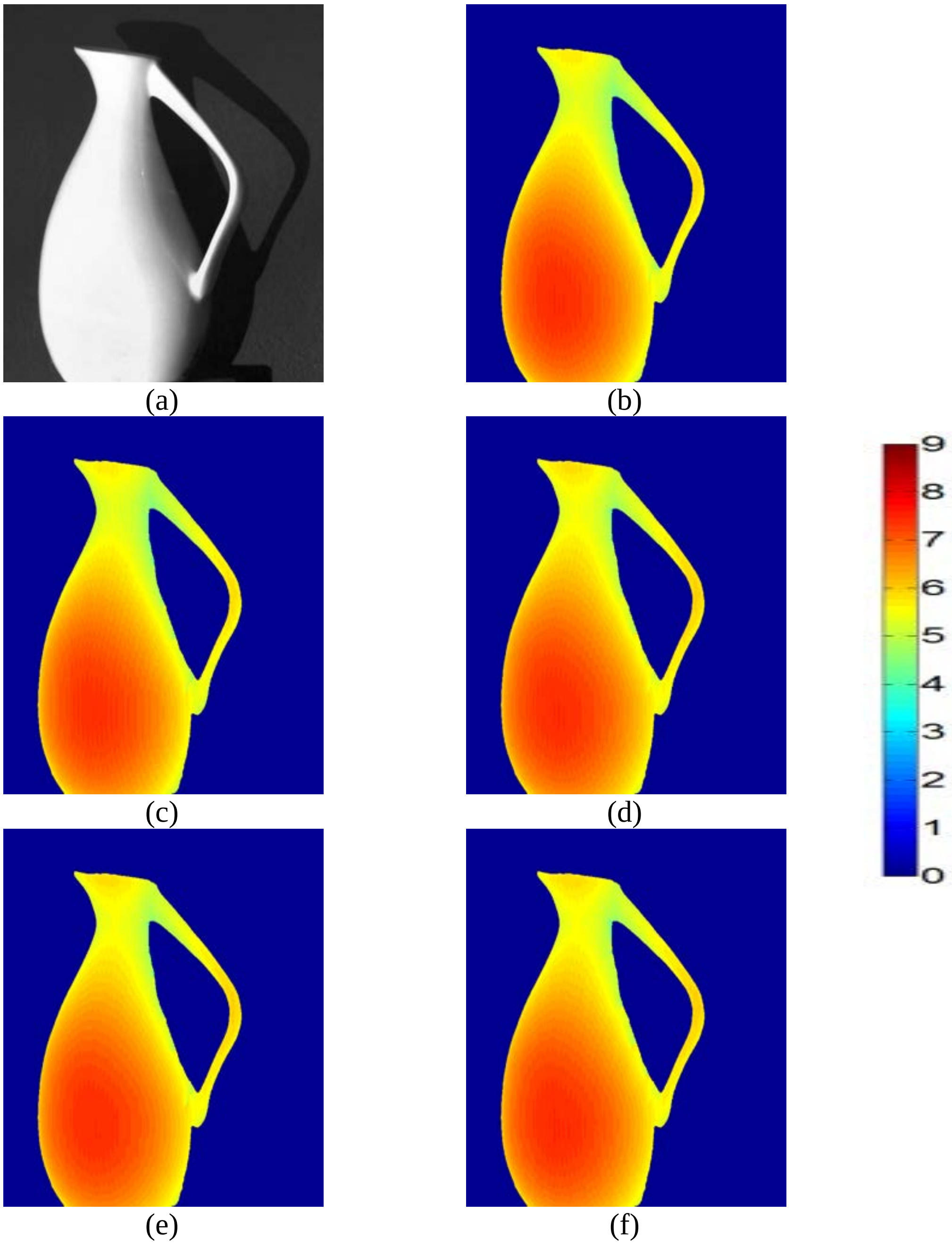


Figure 4-7 Experimental results for measuring a jug: (a) object and the depth maps obtained with (b) sinusoidal fringe patterns, (c) *bpatch*, (d) *opt-i*, (e) *cpatch*, and (f) the proposed fringe patterns.

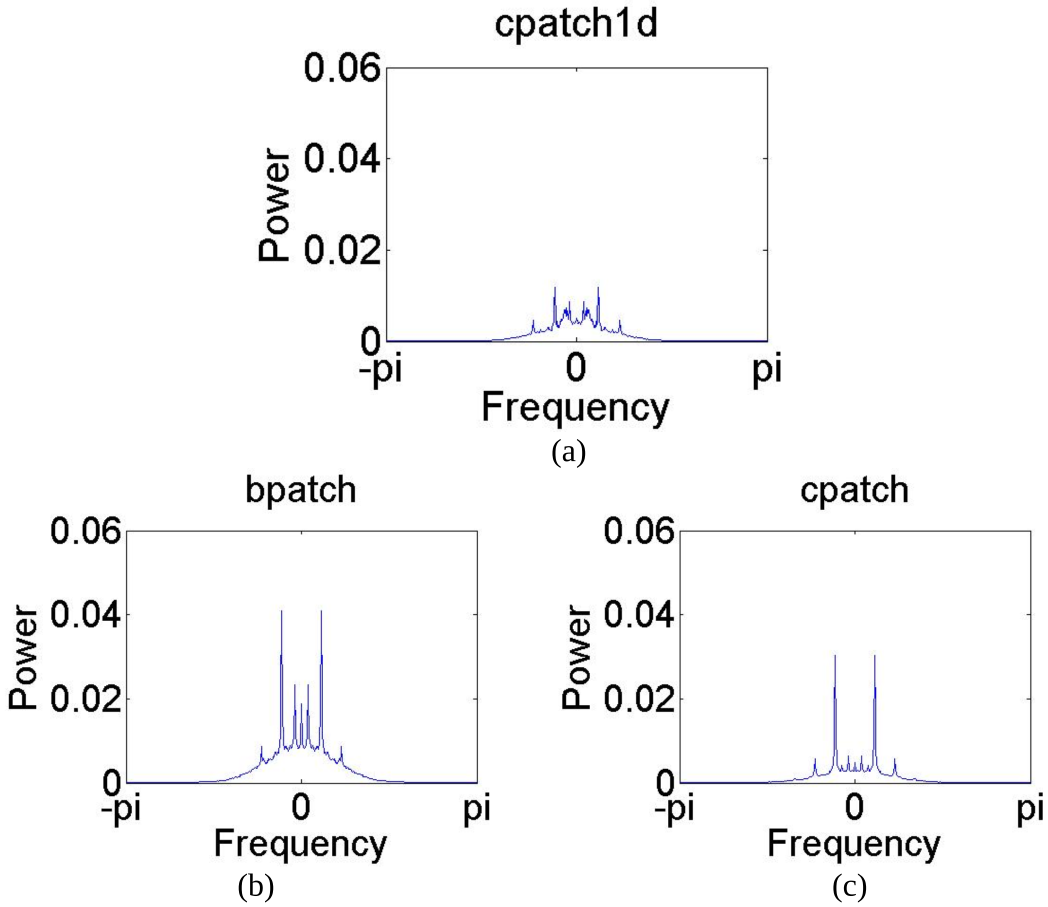


Figure 4-8 Power spectral densities of the phase errors associated with (a) the proposed fringe pattern *cpatch1D* , (b) *bpatch*, and (c) *cpatch.*

4.5 Summary

In this Chapter, we propose a framework for generating aperiodic octa-level fringe patterns for real time 3D shape measurement and an algorithm to optimize patches that can be used to support the proposed tiling method. As compared with conventional patch-based frameworks, it is able to produce fringe patterns of arbitrary fringe period and higher gray-level resolution without introducing harmonic distortion. The achieved depth measuring performance can be significantly improved and is also robust to fringe period and defocusing extent. The gain is almost at no cost because a measuring system exploiting the proposed octa-level fringe patterns shares the same advantages with the systems using binary fringe patterns. This work was published in a journal paper [26].

For the sake of reference, the method used to generate octa-level fringe patterns is referred to as *cpatch1D* hereafter.

Chapter 5.

Improving octa-level fringe patterns by 2D tiling optimization and multiscale error diffusion

5.1 Introduction

Conventional patch-based fringe pattern generation methods tile patches regularly to produce full-size fringe patterns, which introduces strong harmonic distortion to the depth map of the measured object in 3D measurement. In Chapter 4, we address this issue and reduce the harmonic distortion along a specific direction significantly. However, harmonic distortion can still be found in the other orthogonal direction.

Another observation is that, though conventional fringe pattern generation methods formulate the pattern generation as an optimization problem, the problem is generally solved by iteratively refining an initial estimate due to its unaffordable complexity. The solution is generally not the global optimum but a local optimum in terms of an objective function. If the refining step is not flexible, the solution will be biased to the initial estimate and its performance can be far from the optimal. In view of this, a better initial estimate and a flexible refining scheme would definitely be helpful to get better fringe patterns.

This Chapter presents a novel method to generate patch-based octa-level fringe patterns for improving the measuring performance of a 3D surface measuring system.

As compared with the octa-level fringe pattern generation method proposed in Chapter 4, its contributions include:

1. A new method is proposed to optimize patches such that they can be flexibly and seamlessly tiled to form octa-level fringe patterns the achieved phase error of which contains almost no harmonic distortion along any direction.
2. A new approach is proposed to get a better initial estimate of the fringe patterns for further refinement such that fringe patterns of better noise characteristics can be obtained after the optimization process.
3. A more flexible and efficient approach is proposed to refine the fringe patterns such that the optimization process can converge faster to a solution of better noise characteristics.

The organization of this Chapter is as follows. In Section 5.2, we propose a modified feature-preserved multiscale error diffusion (FMED) technique for generating initial octa-level fringe patterns with desired noise characteristics. In Section 5.3, we extend the tiling idea of Section 4.3 into the two-dimensional space and design a dedicated 2D tiling pattern for optimization. In Section 5.4, we introduce a fast convergence optimization that can dramatically increase the efficiency of octa-level fringe pattern generation. Simulation and experimental results for performance evaluation are given in Section 5.5. Finally, a brief summary is provided in Section 5.6.

5.2 Initial fringe pattern generation for optimization

Conventional binary defocusing methods (e.g. [10]–[12], [18], [21], [23], [25] and [26]) first generate initial binary fringe patterns and then refine them to produce the final binary fringe patterns for approximating sinusoidal patterns. This two-stage process is also adopted to develop octa-level fringe patterns in our work in Chapters 3 and 4.

Obviously, the initial fringe patterns play a significant role in reaching the final optimization result because the optimization step is imperfect in a way that it can only provide local optimum solutions. Conventionally, halftoning methods such as error diffusion [39] and Bayer dithering [42] are exploited to obtain the initial fringe pattern (e.g. [11], [12], [21] and [26]), or patches are randomly assigned initial values(e.g. [10], [23] and [25]). None of these approaches can provide ideal initial fringe patterns. For instance, error diffusion [39] diffuses the quantization error to a pre-defined direction with a fixed causal diffusion filter and hence causes directional hysteresis [1,10, 11].

The subsequent refinement step cannot guarantee a globally optimized approximation result. It implies that the approximation performance of the final octa-level fringe patterns is initial-estimate dependent. In general, we can expect that a better initial estimate can lead to a better final fringe pattern. In view of this, a modified version of feature-preserving multiscale error diffusion (FMED) is adopted here to generate our initial octa-level patterns as FMED has already been proven to be good at producing binary and multi-level halftoning results that possess ideal noise characteristics [47–49].

Let I_k for k∈{1,2,3} be the sinusoidal fringe patterns used in three-step phase shifting profilometry and L_k be the corresponding octa-level fringe pattern that approximates I_k. Without loss of generality, we assume that I_k is scaled and offsetted such that its minimum and maximum are, respectively, 0 and 1. Fringe patterns L_k for k∈{1,2,3} are generated separately. For each k, an error image E_k is initialized to be

I_k at the beginning. Pixels of L_k are then picked iteratively to determine their intensity values until all pixel values of L_k are determined. The steps in each iteration cycle are as follows:

1) Select a pixel in L_k via the extreme error intensity guidance (EEIG) based on the most updated E_k. One may refer to Section 2.2.3 for the details of EEIG (Section 2.2.3). Let the selected pixel position be (m,n).

2) Quantize $E_k(m,n)$ to the nearest intensity value in

$$\Psi = \{0.299r + 0.587g + 0.114b | r,g,b \in \{0,1\}\} \tag{5-1}$$

and assign the quantized value to $L_k(m,n)$.

3) Update E_k by diffusing the quantization error, which is

$$Q(m,n) = E_k(m,n) - L_k(m,n), \tag{5-2}$$

to the neighborhood of $E_k(m,n)$ with an adaptive non-causal diffusion filter as follows.

$$E_k(i,j) = \begin{cases} 0 & \text{if } (i,j) = (m,n) \\ E_k(i,j) + \frac{f(i-m,j-n)\cdot D(i,j)\cdot Q(m,n)}{K} & \text{if } (i-m,j-n) \in \Omega \end{cases} \tag{5-3}$$

where

$$f(p,q) = \begin{cases} 0.0717 & \text{if } |p| = |q| = 1 \\ 0 & \text{if } p = q = 0 \\ 0.1783 & \text{if } |p| + |q| = 1 \end{cases} \tag{5-4}$$

is a filter coefficient of a non-causal filter whose support is Ω={(p,q) | p,q∈{-1, 0, 1}},

$$D(i,j) = \begin{cases} 0 & \text{if } L_k(i,j)\text{'s value has been determined} \\ 1 & \text{else} \end{cases} \tag{5-5}$$

and

$$K = \sum_{(i-m,j-n)\in\Omega} \left(f(i-m,j-n)\cdot D(i,j)\right) \tag{5-6}$$

When K=0, a filter with a larger support window is exploited as suggested in [48] to allow the algorithm to proceed.

The constructed octa-level fringe patterns L_k are good estimates of I_k for $k \in \{1,2,3\}$ indeed. Some simulation results will be provided in Section 5.5 to verify this fact.

If we do not opt for patch-based octa-level fringe patterns, we can directly use L_k for $k \in \{1,2,3\}$ as the initial estimate to start an iterative optimization process. However, since we aim for patch-based fringe patterns, we extract patches from L_k to construct patch-based initial fringe patterns for optimizing the patches. The details will be discussed in the following sections.

5.3 Formulation of the optimization problem

Patch tiling tends to produce regular and periodic patterns that introduce harmonic distortion. To solve this problem, special care is taken during the optimization in *cpatch1D* [26] to make sure that a tiling output is aperiodic along a specific direction. In this Section, we extend the idea to further improve the optimization objective function such that both horizontal and vertical periodicity can be avoided when tiling the optimized octa-level patches.

Let $P_{(s,k)}$ be the k^{th} octa-level patch of set $s \in \{0,1\}$, where $k \in \{1,2,3\}$. Each patch is of size $N_x \times N_y$, where N_x is the period of the target sinusoidal patterns and N_y is an integer value. In this approach, for each k, we tile patches $P_{(1,k)}$ and $P_{(2,k)}$ as shown in Figure 5-1 to form an octa-level pattern P_k and then optimize P_k for $k \in \{1,2,3\}$ simultaneously to minimize an objective function.

The pattern P_k shown in Figure 5-1 bears two important properties:

1) The occurrence of $P_{(1,k)}$ and $P_{(2,k)}$ is identical in pattern P_k The optimization does not favor one particular group of patches.

2) The occurrences of all possible horizontal and vertical connection combinations between any two patches randomly and independently picked from $\{P_{(s,k)} | s=0,1\}$ are identical. There is no bias to favor a particular connection combination during the optimization,

The two properties guarantee that the optimized patches can be seamlessly connected to each other when they are tiled horizontally and vertically.

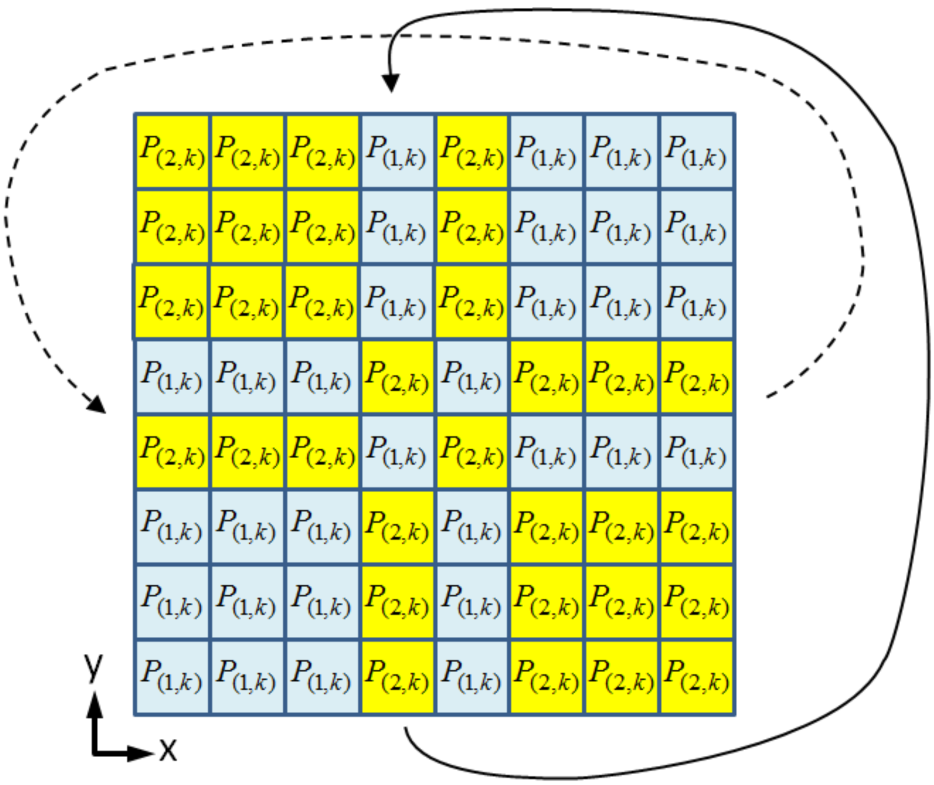


Figure 5-1 2D circular connected octa-level pattern P_k with two octa-level patches for optimization

To make the approximation performance robust to amount of defocusing, the objective function to be minimized in the optimization is selected as

$$J = \Delta\varphi^{(5)} + \Delta\varphi^{(11)} \tag{5.7}$$

where

$$\Delta\varphi^{(t)} = \sqrt{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(\varphi(x,y) - \varphi_{G^{(t)}}(x,y))^2} \tag{5.8}$$

is the phase rms error achieved when the amount of defocusing can be modeled as the filtering effect of Gaussian filter $G^{(t)}$. In eqn. (5.8), $M\times N$ is the size of P_k, $(= 8N_x\times 8N_y$ for the pattern shown in Figure 5.1) and $\varphi_{G^{(t)}}$ is the phase map obtained when the sinusoidal fringe patterns are replaced with the octa-level fringe patterns defocused by filter $G^{(t)}$.

Here, we note that, in formulation, objective function (5.7) and its supporting function (5.8) are identical to the objective function used in the fringe pattern generation method proposed in Chapter 4 (eqn. (4.3)) and its corresponding supporting function (eqn. (4.1)). However, the way that we construct the octa-level fringe patterns for being optimized is different in the two approaches. This difference introduces different constraints in the optimization and hence leads to different optimization results.

In summary, our optimization problem to be solved is formulated as

$$\min_{P_{(s,k)} \text{ for } s\in\{1,2\}, k\in\{1,2,3\}} J \tag{5.9}$$

subject to the constraints that P_k is constructed with $P_{(s,k)}$ for s=1,2 as shown in Figure 5-1 and that

$$P_{(s,k)} = 0.299B_{R_{(s,k)}} + 0.587B_{G_{(s,k)}} + 0.114B_{B_{(s,k)}} \tag{5.10}$$

where $B_{R_{(s,k)}}$, $B_{G_{(s,k)}}$ and $B_{B_{(s,k)}}$ are three $N_x\times N_y$ binary patterns representing the intensity variations in color channels R, G and B respectively.

5.4 Realization of the optimization

When solving an optimization problem to develop a binary or octa-level fringe pattern, conventional algorithms generate an initial estimate of the pattern and then refine it iteratively to get a sub-optimal solution. The adjustment is realized by replacing the current pixel value with the 'best' value that is determined through an exhaustive search among all possible pixel values. This process is very time-consuming. Besides, the non-flexible static scanning order adopted in each refinement iteration cycle makes the final solution easily trapped in a sub-optimal local minimum that is very close to the initial estimate. Patch-based algorithms (e.g. [6], [10], [18], [23], [25] and [26]) alleviate the computation efficiency a bit by downsizing the problem to a problem that optimizes a single patch. However, the trapping issue is even severe in such a case and a number of patches must be tested before the 'best' patch is reached. When the same approach is exploited to optimize an octa-level fringe pattern, it is more complex because it involves manipulation of three binary fringe patterns.

In this Section, we propose to adjust pixels according to a necessity-oriented order instead of a location-oriented order. The introduced flexibility and adaptability make the optimization converge much faster and avoid being trapped in a local optimum before exploring sufficient possibilities. This optimization algorithm is referred to as necessity oriented optimization (NOO) and its details are as follows:

Step 1. Initialization of $P_{(s,k)}$ for $s\in\{1,2\}$:

Generate three octa-level patterns L_k for $k\in\{1,2,3\}$ to approximate three sinusoidal fringe patterns with the FMED-based technique presented in Section 5.2. Chop two connected $N_y\times N_x$ segments from each of them to form initial patches $P_{(s,k)}$ for $s \in \{1,2\}$. P_k is then constructed with $P_{(1,k)}$ and $P_{(2,k)}$ as shown in Figure 5-1.

Step 2. Refining $P_{(s,k)}$ for $s \in \{1,2\}$ based on a necessity-oriented strategy:

As shown in Figure 5-1, P_k can be partitioned into 64 blocks of size $N_y \times N_x$ and each of them is either patch $P_{(1,k)}$ or $P_{(2,k)}$. Accordingly, these blocks are divided into two groups based on the criterion whether they are $P_{(1,k)}$ or $P_{(2,k)}$. Once a pixel in a patch is adjusted, all the blocks in the same corresponding group will be affected.

Let $P_k(m,n)$ be the $(m,n)^{\text{th}}$ block of P_k for $0 \le m,n < 8$ and $P_{k(m,n)}(p,q) \equiv P_k(mN_x + p, nN_y + q)$ be the $(p,q)^{\text{th}}$ pixel in block $P_k(m,n)$. In formulation, the grouping can be done based on the index of a block and it is given as

$$\Lambda_s = \{(m,n) | 0 \le m,n < 8 \text{ and } P_{k(m,n)} \text{ is } P_{(s,k)}\} \quad \text{for } s \in \{1,2\} \tag{5.11}$$

For a specific patch combination, the contribution of the $(i,j)^{\text{th}}$ pixels of patches $P_{(s,1)}$, $P_{(s,2)}$ and $P_{(s,3)}$ to absolute phase error ε is roughly given as

$$\varepsilon(i,j;s) = \sum_{(m,n) \in \Lambda_s} \sum_{t=5,11} \left| \varphi(mN_x + i, nN_y + j) - \varphi_{G_t}(mN_x + i, nN_y + j) \right| \tag{5.12}$$

We sort triplets in set $T = \{(i,j;s) | 0 \le i < N_x, 0 \le j < N_y \text{ and } s \in \{1,2\}\}$ according to the value of $\varepsilon(i,j;s)$ in descending order and then pick the triplets from the sorted triplet sequence one by one. When triplet $(i,j;s)$ is picked, the $(i,j)^{\text{th}}$ pixels of patches $P_{(s,k)}$ for k=1,2,3 are adjusted simultaneously. The picking stops when $\varepsilon(i,j;s) < \varepsilon(i_o,j_o;s_o)/10$, where $(i_o,j_o;s_o)$ is the first triplet in the sorted sequence.

Assume that the currently picked triplet is $(i',j';s')$. Since $P_{(s',k)}(i',j')$ is an octa-level value, there are altogether 8^3=512 possible value combinations of $P_{(s',k)}(i',j')$ for $k \in \{1,2,3\}$. For each combination, construct a candidate set of patches $P_{(s',1)}$, $P_{(s',2)}$ and $P_{(s',3)}$ (denoted as $P^c_{(s',1)}$, $P^c_{(s',2)}$ and $P^c_{(s',3)}$ respectively) in which we have $P^c_{(s',k)}(i,j) = P_{(s',k)}(i,j)$ for all $(i,j) \neq (i',j')$, and update fringe patterns P_k for all k by replacing blocks $P_{k(m,n)}$ with $P^c_{(s',k)}$ for $(m,n) \in \Lambda_{s'}$.

Among all 512 candidate sets of patches, the one used to construct the fringe patterns that minimize J is selected to be the most updated $P_{(s',1)}$, $P_{(s',2)}$ and $P_{(s',3)}$.

Step 3. Terminate the iteration under the following criterion:

Go back to step 2 if the total improvement in step 2 is larger than 0.01% in terms of J. Otherwise the most updated $P_{(1,k)}$ and $P_{(2,k)}$ for $k \in \{1,2,3\}$ are considered as the optimal patches.

Step 4. Generating the final octa-level fringe pattern:

Randomly tile optimized patches $P_{(1,k)}$ and $P_{(2,k)}$ horizontally and vertically to form full-size octa-level fringe patterns L'_k for $k \in \{1,2,3\}$ under the condition that L'_1, L'_2 and L'_3 share the same random tiling pattern. This step can be done on site to cater for different systems.

5.5 Performance evaluation

This Section provides some simulation results and experimental data for evaluating the performance of the fringe pattern generation method presented in this Chapter.

5.5.1 Evaluation of initial octa-level fringe pattern

As described in Section 5.2, modified FMED is applied to generate initial fringe patterns in this work. The quality of the obtained initial fringe patterns is evaluated here.

Figures 5-2(a) and (b) show two octa-level fringe patterns and their corresponding noise spectra. They were respectively produced with the error diffusion method used in *cpatch1D*[55] and the proposed FMED-based method to approximate a sinusoidal fringe pattern whose period is 60 pixels. From their corresponding noise spectra shown in Figures 5-2(a) and (b), one can see that the former carries serious harmonic distortion while the latter does not. Besides, most of the noise in the fringe pattern shown in Figure 5-2(b) is high frequency noise, which can actually be removed by the defocusing process when the fringe pattern is projected onto an object during measurement. In fact, by modeling the defocusing process as a 5×5 Gaussian filtering process, the intensity rms errors of the defocusing outputs of the fringe patterns shown in Figure 5-2(a) and (b) are, respectively, 0.0115 and 0.0077 with respect to the grayscale sinusoidal pattern. The octa-level fringe pattern produced with the proposed FMED-based approach is obviously superior.

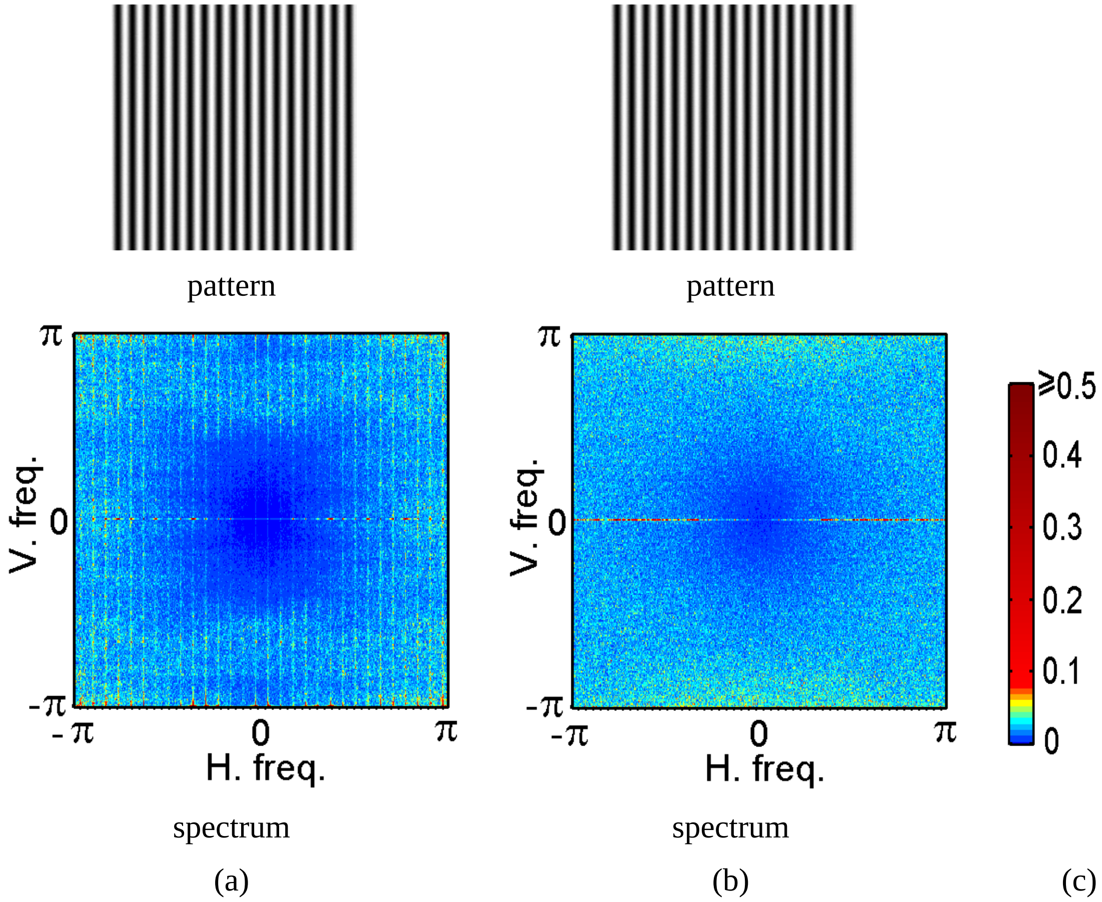


Figure 5-2 Preliminary octa-level fringe patterns obtained with different methods and their noise spectra: (a) error diffusion-based [21, 26] and (b) FMED-based. The magnitudes of all components in both spectra are normalized with respect to the strongest component in error diffusion-based approach's spectrum for easier comparison. The color bar shown in Figure 5-2(c) is shared by both spectra to map magnitude values into colors.

5.5.2 Evaluation of convergence speed

The proposed necessity-oriented patch-based fringe pattern optimization scheme can speed up the optimization process besides improving its optimization result. Figure 5.3 shows the difference in convergence speed when adopting the raster-scanned ordering used in [10]–[12], [18], [21], [23], [25] and [26] instead of the necessity-oriented ordering in step 2 of NOO. Each curve in the plot shows how the value of the objective function defined in eqn.(5.7) decreases with number of iterations in our simulation.

The simulation data are obtained under the condition that the fringe period is 60 pixels and the patch size is 20×60 pixels. The convergence speed is much faster when the necessity-oriented ordering is used. In 5300 iterations, the faster approach can achieve the performance that the slower one achieves after 16,000 iterations.

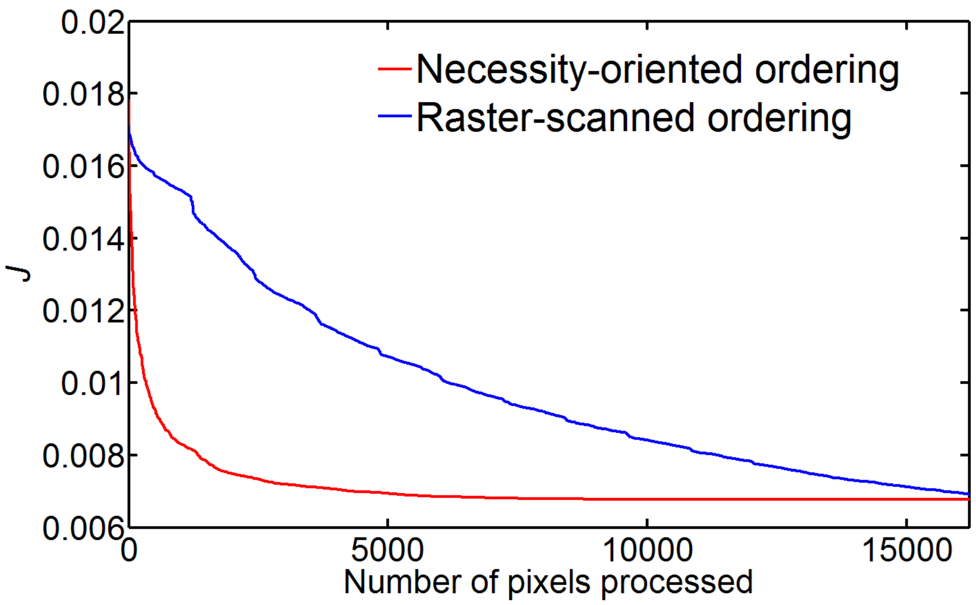


Figure 5-3 Impacts of different ordering schemes on convergence speed

5.5.3 Simulation results

Simulations were carried out to evaluate the performance of the octa-level fringe patterns proposed in this Chapter (referred to as *cpatch2D* hereafter.). The performance was measured in terms of the phase error achieved with the defocused fringe patterns. Binary fringe patterns generated with *bpatch* [10], as well as the octa-level fringe patterns proposed in Chapters 3 and 4 (i.e. *cpatch* [25] and *cpatch1D* [26] respectively) were also realized for comparison. All these algorithms were implemented using the same parameters as described in their corresponding papers.

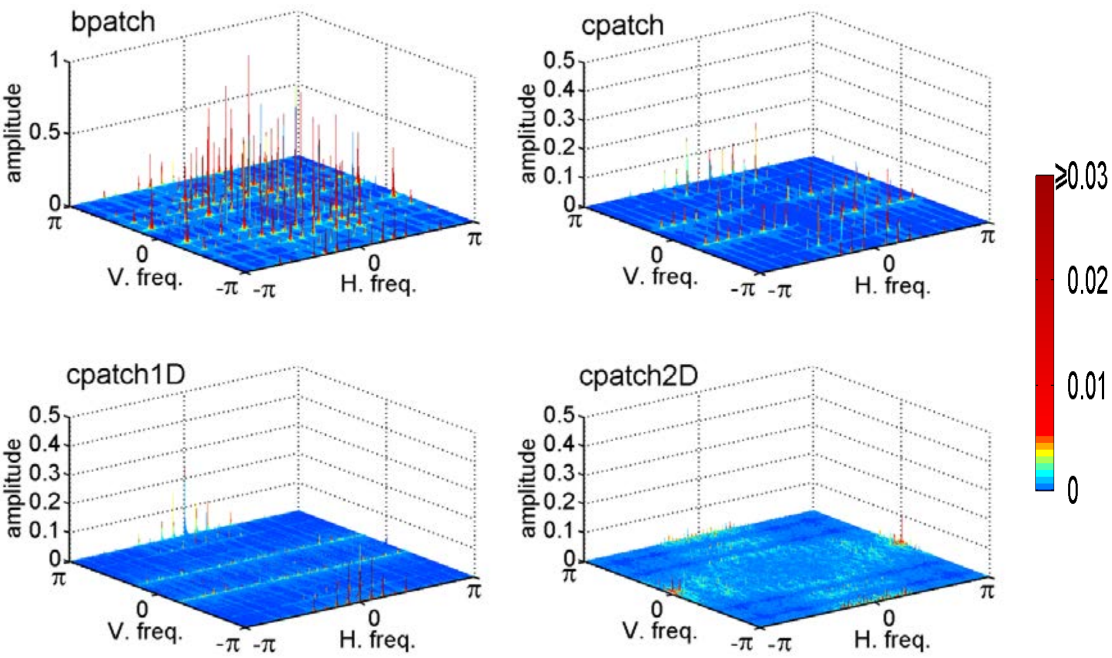


Figure 5-4 Normalized noise spectra of the phase maps associated with *bpatch*, *cpatch*, *cpatch1D* and *cpatch2D*.

The first simulation was for investigating whether the proposed method could effectively eliminate harmonic distortion in their phase maps two dimensionally. In the simulation, all fringe patterns were of size 1024×768 pixels and the fringe period of the target sinusoidal fringe patterns to be approximated was 60 pixels. Figure 5-4 shows the normalized amplitude noise spectra of the phase maps achieved by *bpatch*, *cpatch*, *cpatch1D* and *cpatch2D* under the ideal defocusing condition (i.e. the assumed defocusing level in the optimization schemes). The magnitudes of the components in all 4 plots are normalized with respect to the strongest component in all 4 plots for easier comparison. The advantage of the proposed patterns is obvious in the plots. The outstanding regular peaks in the plot of *bpatch* reflect that there are strong harmonic components in its phase error. The situation is gradually improved by *cpatch* and *cpatch1D*. *Cpatch1D* can only handle the horizontal periodicity in a tiled fringe pattern, so there are still vertical harmonic components. In contrast, the harmonic distortion in both directions is almost eliminated in the phase map achieved by *cpatch2D*.

Figure 5-5 shows a cross-section of an ideal sinusoidal fringe pattern along the x-direction and the corresponding cross sections of the defocused patterns of *cpatch1D* and *cpatch2D*. Only *cpatch1D* and *cpatch2D* are compared here as *bpatch* and *cpatch* are not designed to handle the harmonic distortion of a tiled fringe pattern. One can see that the *cpatch2D* fringe pattern is closer to the ideal sinusoidal signal after defocusing. This superiority is achieved with the proposed necessity-oriented optimization scheme and the better initial patterns used in the optimization scheme. Note that both *cpatch1D* and *cpatch2D* attempt to minimize the phase error achieved by the fringe patterns instead of the approximation error between an octa-level fringe pattern and the target sinusoidal fringe pattern. Theoretically, when phase-domain optimization is adopted, the resultant fringe patterns are not necessary to be ideally sinusoidal as long as the phase error is minimized under a specific defocusing condition [11]. However, the lower the approximation intensity difference, the more robust to the defocusing extent the measuring performance is [11]. This advantage helps *cpatch2D* to perform better in real experiments when the defocusing condition cannot be controlled perfectly.

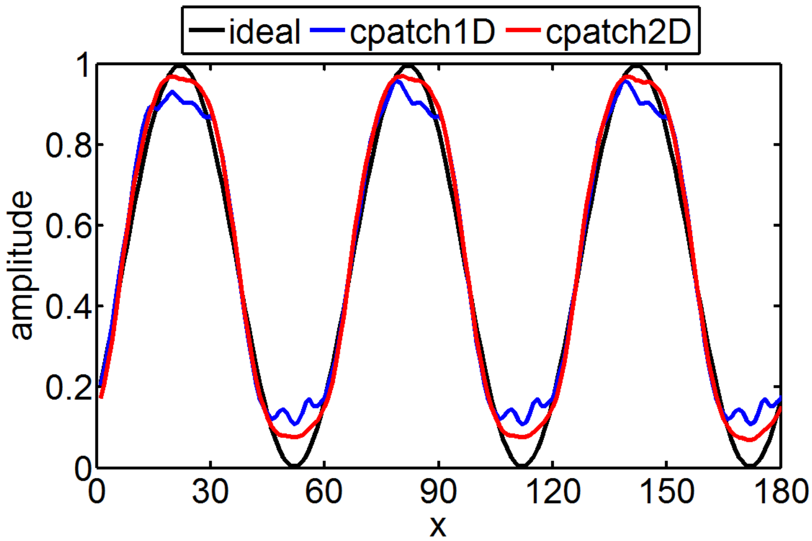


Figure 5-5 Cross-sections of defocused fringe patterns *cpatch1D* and *cpatch2D*. The defocusing is simulated with a 5×5 Gaussian filter with a standard deviation of 5/3.

Figure 5.6 shows some plots for quantitative comparison. The results were achieved with the evaluated fringe patterns in various defocusing conditions. One can see that *cpatch*, *cpatch1D* and *cpatch2D* outperform *bpatch*. This is expected because octa-level fringe patterns are better than binary fringe patterns. Since *cpatch1D* and *cpatch2D* proactively suppress harmonic distortion, they are able to achieve smaller phase rms error than *cpatch*. By comparing *cpatch1D* and *cpatch2D*, one can notice that phase rms errors of the two are similar when the extent of defocusing is slight. However, *cpatch2D* performs better when the defocusing extent is sufficiently large (e.g. the blurring effect can be modeled with Gaussian filter $G^{(t)}$ where t>9). By considering that *cpatch2D* can eliminate harmonic distortion in all directions as shown in Figure 5-4, *cpatch2D* is the best of the four algorithms.

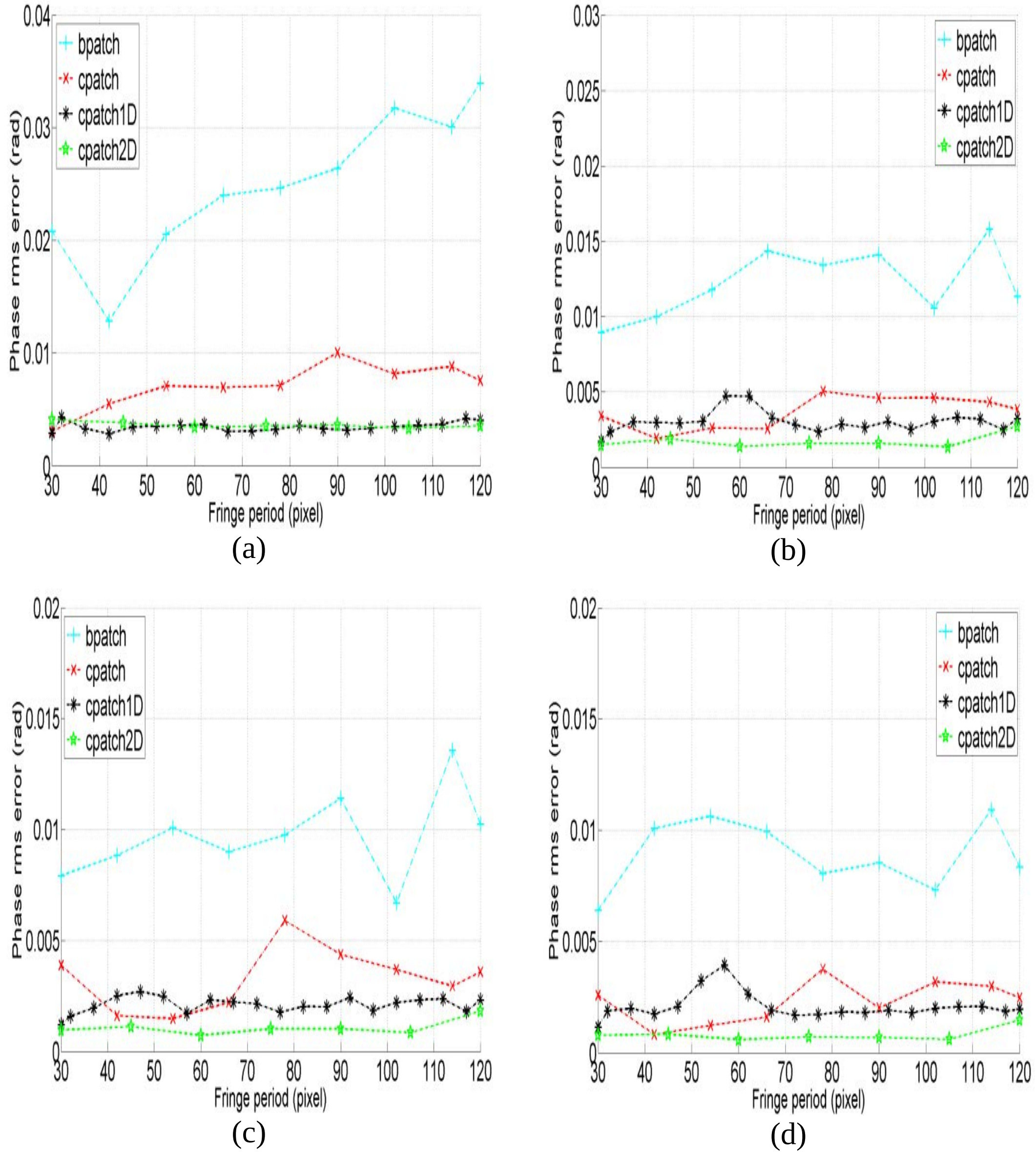


Figure 5-6 Simulated phase rms errors achieved with different fringe patterns of different fringe periods when the defocusing levels are simulated by Gaussian filters of sizes (a) 5×5, (b) 9×9, (c) 11×11 and (d) 13×13.

Figure 5-7 shows some simulation data for measuring a complex three-dimensional surface. In this simulation, the fringe period of all fringe patterns was 60 pixels, and Goldstein's branch cut unwrapping algorithm [65] was exploited to acquire the unwrapped phase. The defocusing process was modeled as a 5×5 Gaussian low-pass filter with standard derivation equal to 5/3.

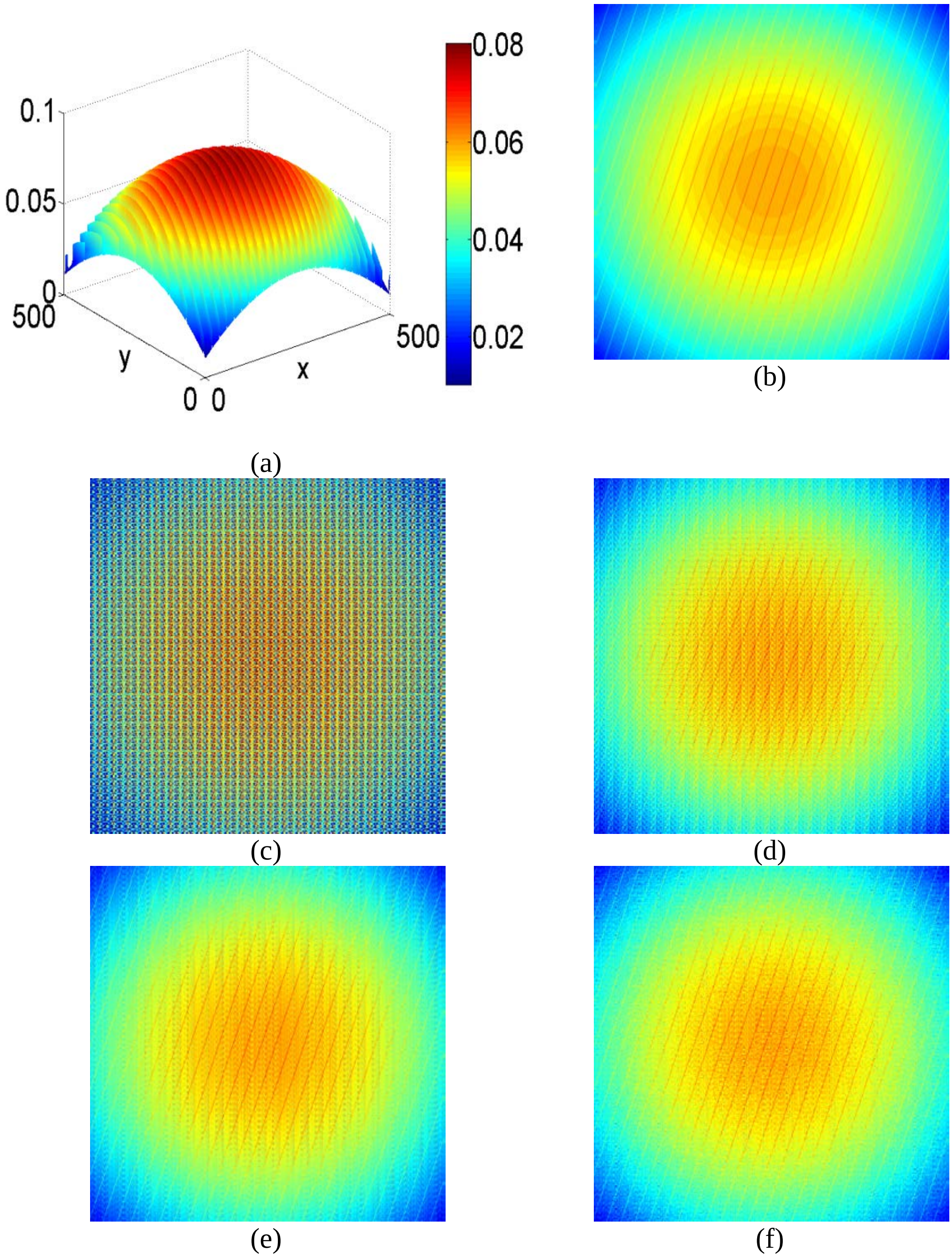


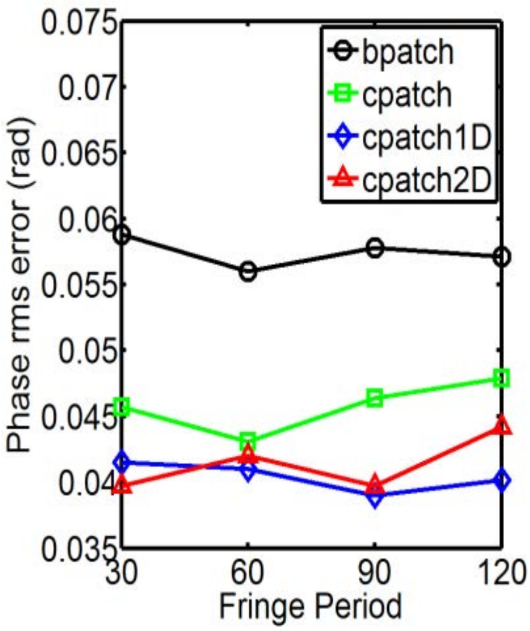
Figure 5-7 Simulation results for measuring an object. (a) 3D plot of an object. (b) ideal unwrapped phase map of (a); and unwrapped phase maps obtained with (c) *bpatch*, (d) *cpatch*, (e) *cpatch1D* and (f) *cpatch2D*. The horizontal and the vertical dimensions are x- and y-dimensions respectively in (b)-(f).

In Figure 5-7, we can observe that the object is flooded in the noise in *bpatch*'s result. The harmonic distortion in *cpatch*'s and *cpatch1D*'s results provides misleading messages as a set of false parallel ridges are mixed with the true ones. In contrast, the object surface can be reported more faithfully in our result. As a matter of fact, the corresponding phase rms errors of Figures 5-7(c), (d), (e) and (f) are, respectively, 0.0221, 0.0089, 0.0048 and 0.0042 (in rad).

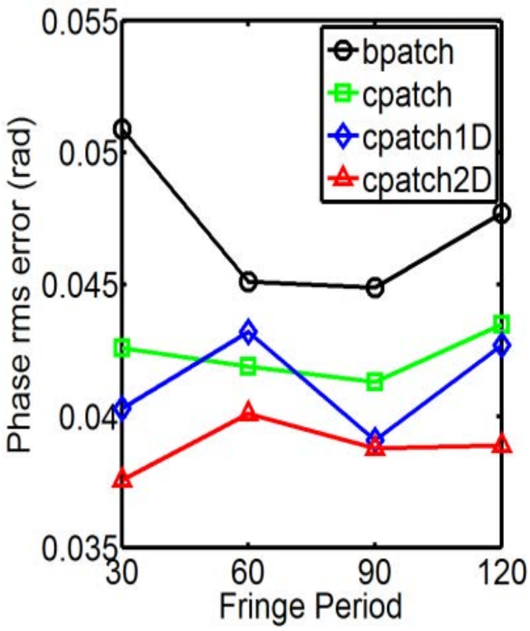
5.5.4 Experimental results

A real 3D shape measurement system was set up to verify the performance of the proposed technique. The system consists of a DLP projector (Texas MP723) and a charge-coupled-device (CCD) camera (Canon 600D). The size of a projected fringe pattern is 1024×768 pixels. The reference board was placed around 50 cm away from the projector. Throughout the experiment, Goldstein's branch-cut unwrapping algorithm [65] was applied to obtain unwrapped phase information for constructing the depth map. When continuous-tone phase-shifted fringe patterns were used, active gamma correction [66] was applied to reduce the impact of gamma nonlinearity of the projector.

In the first experiment we measured a flat black board 50 cm away from the projector. The ground truth is obtained by measuring the board with the continuous-tone nine-step phase-shifted fringe pattern [81]. When binary or octa-level fringe patterns were projected, we adjusted the focal lens of the projector to change the defocusing levels such that corresponding data can be obtained to verify their robustness to defocusing levels. The phase rms errors achieved are illustrated in Figure 5-8.



(a)



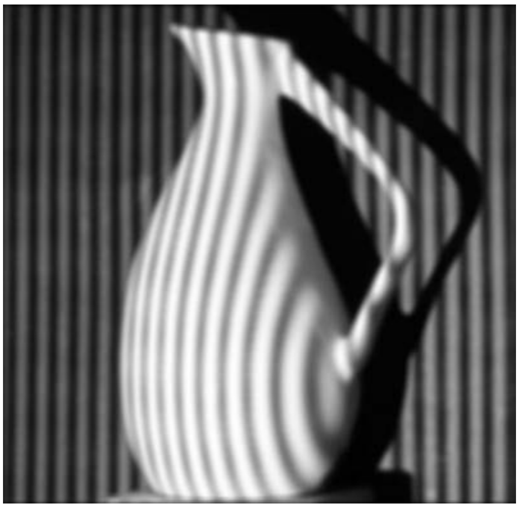
(b)

Figure 5-8 Experimental phase rms errors obtained with (a) a slightly defocused projector and (b) a more defocused projector when different fringe patterns are used.

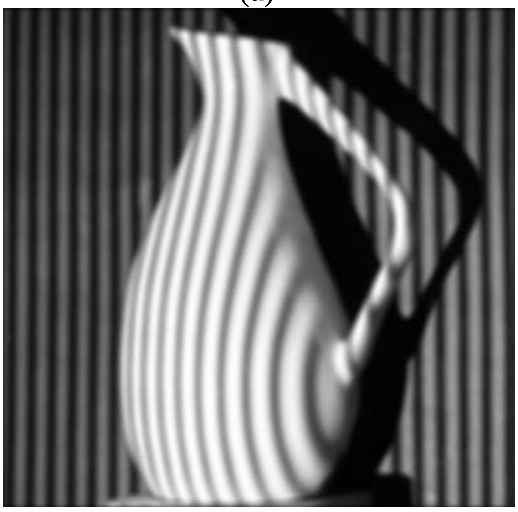
As expected, due to the existence of environmental noise, the phase rms error measured in real experiments is higher than that achieved in simulations. Although it is true, one can still see in Figure 5-8 that the phase rms errors of octa-level fringe patterns (*cpatch*, *cpatch1D* and *cpatch2D*) are smaller than that of binary patterns (*bpatch*) due to the increase of gray levels. When the projector is slightly defocused, the performance of *cpatch1D* and *cpatch2D* is similar. However, when the projector is more defocused, *cpatch2D* outperforms *cpatch1D*. This observation confirms that *cpatch2D* is more robust to defocusing.

Another experiment was conducted to measure a real object. For octa-level fringe patterns (*cpatch*, *cpatch1D* and *cpatch2D*), only the luminance plane is captured in the camera. The phase information of the object is constructed by comparing the phase map of the reference board and the phase map of the object. The final depth map is established according to the unwrapped phase information of the object.

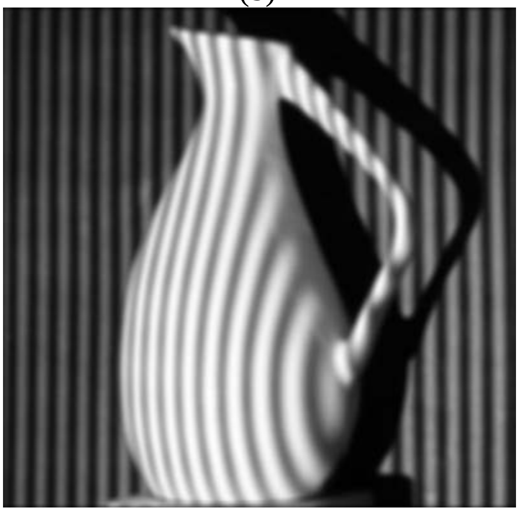
(a)



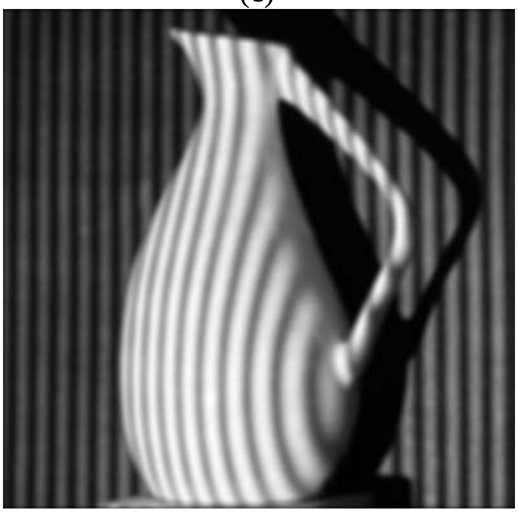
(b)



(c)



(d)



(e)

Figure 5-9 Images captured when different fringe patterns are projected onto a measured object. (a) no projection, (b) *bptach*, (c) *cpatch*, (d) *cpatch1D* and (e) *cpatch2D*.

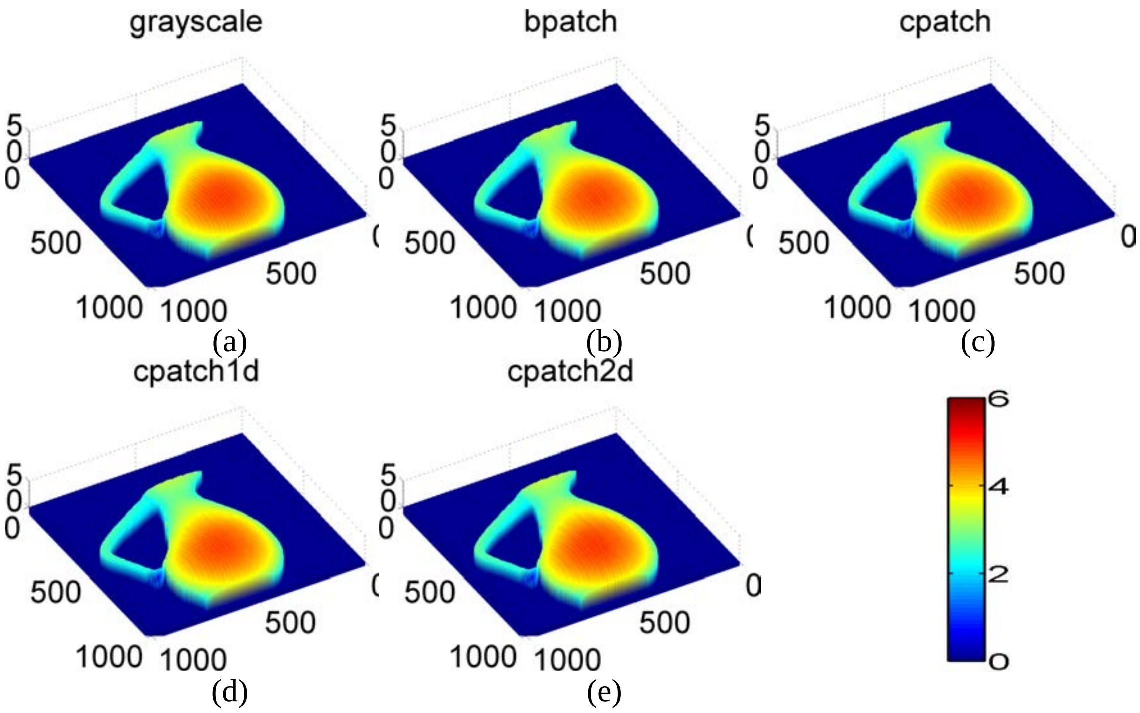


Figure 5-10 Experimental results for measuring a jug object when fringe period = 18 pixels. Depth maps obtained with (a) nine-step grayscale sinusoidal fringe patterns, (b) *bptach*, (c) *cpatch*, (d) *cpatch1D* and (e) *cpatch2D*.

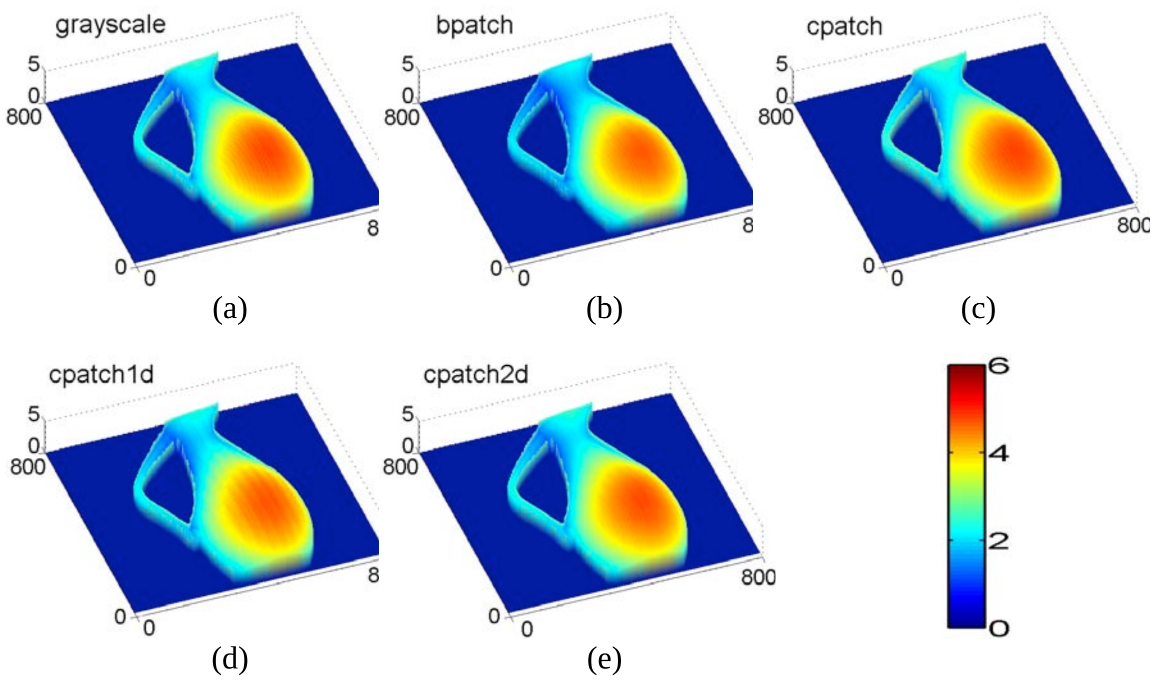


Figure 5-11 Experimental results for measuring a jug object when fringe period = 30 pixels.. Depth maps obtained with (a) nine-step grayscale sinusoidal fringe patterns, (b) *bptach*, (c) *cpatch*, (d) *cpatch1D* and (e) *cpatch2D*.

Figure 5-9 shows the captured grayscale images when different fringe patterns were projected onto the surface of the object. Their corresponding constructed depth maps are illustrated in Figure 5-10 and Figure 5-11 where different fringe periods are applied.

The depth map obtained with the nine-step phase-shifted algorithm [81] is used as the reference for comparison and it is shown in Figure 5-10(a) and Figure 5-11(a). Figures 5-10(b), (c), (d) and (e) show, respectively, the depth maps obtained with *bpatch*, *cpatch*, *cpatch1D* and *cpatch2D*. The depth map shown in Figures 5-10 adopted fringe patterns of fringe period = 18 pixels where depth map shown in Figures 5-11 applied fringe patterns of fringe period = 30 pixels. Compared with Figures 5-10, one can observe stripes in Figures 5-11, which are introduced by gamma non-linearity of the experiment system. The corresponding depth rms error in the depth maps in Figures 5-11 are, respectively, 0.1133, 0.0671, 0.0583 and 0.0240 cm. Note that no gamma calibration for the projector is required when these fringe patterns were projected.

5.6 Summary

In this Chapter, we propose a method to generate high quality octa-level fringe patterns for real time 3D shape measurement. By adopting better approaches to formulate the optimization problem, produce initial fringe pattern estimates for optimization and refine the fringe patterns during the optimization process, the developed fringe patterns can achieve a better measuring performance. Specifically, the phase map produced with the optimized fringe patterns contains almost no harmonic distortion along any direction and little low frequency noise. Though the optimization is carried out in the phase domain, the generated fringe patterns are close to ideal sinusoidal patterns after being defocused, which makes them robust to defocusing extent in real world situations. Moreover, the optimization process is much faster than the one used in the state-of-art octa-level fringe pattern generation algorithm [25, 26]. The work described in this Chapter was accepted to be published in *Optical Lasers Engineering* in 2017.

Chapter 6.

Application of halftoning in reversible color-to-grayscale conversion

6.1 Introduction

Both Chapters 6 and 7 are dedicated to investigating how halftoning can work with color quantization to realize reversible color-to-grayscale conversion (RCGC) effectively. Reversible color-to-grayscale conversion (RCGC) aims at embedding the chromatic information of a full color image into its grayscale version such that the original color image can be recovered in the future when necessary. Conventional RCGC algorithms tend to put their emphasis on the quality of the recovered color image, which makes the color-embedded grayscale image visually undesirable and suspicious.

This Chapter presents a novel RCGC framework that emphasizes the quality of both the color-embedded grayscale image and the recovered color image simultaneously. Its superiority against other RCGC algorithms is mainly achieved by developing a color palette that fits into the application and exploiting error diffusion to shape the quantization noise to high frequency band. The improved quality of the color-embedded grayscale image makes the image appears as a normal image. It does not catch the attention of unauthorized people and hence the embedded chromatic information can be protected more securely.

Chapter 6 is organized as follows. In Section 6.2, we propose the framework of our proposed RCGC algorithm. In Section 6.3, a tailor-made color palette is designed for supporting our RCGC algorithm. Section 6.4 shows how the palette works with halftoning in RCGC. In Section 6.5, we show how to embed the color palette in the index image without degrading the image quality. In Section 6.6, a matched method for suppressing visible halftoning artefacts is introduced. Simulation results for performance evaluation are provided in Section 6.7. Finally, a summary is given in Section 6.8.

6.2 Framework of the proposed RCGC algorithm

The proposed RCGC algorithm is based on color quantization. However, unlike those conventional algorithms [27–31], it makes use of the error diffusion technique [39] to shift the quantization noise to high frequency region such that the noise can be less visible to human eyes. This noise shaping effect makes significant contributions to the performance of the proposed method. On one hand, it allows us to put more effort on improving the quality of the grayscale image as human can tolerate more high frequency noise in the recovered color image. On the other hand, besides the visual quality, the objective quality of the recovered color image can also be guaranteed as its high frequency quantization noise can be removed by low-pass filtering.

Let C is the original color image to be processed and $C(i,j)$ denotes the color of its pixel (i,j). Without loss of generality, $C(i,j)$ is represented as a vector in (r,g,b) format, where r, g and b are the intensity values of the red, green and blue components of the color and they are normalized to be in [0,1]. A color palette is first derived based on image C to bear a property that the color index value of a palette color is highly correlated with the luminance value of the palette color. The details of its derivation are provided in Section 6.3.

After obtaining the color palette, the color image C is processed with luminance-constrained color quantization and vector error diffusion as shown in Figure 6-1(a). In particular, the process scans the image with serpentine scanning and processes the scanned pixels one by one. $C'(i,j)$, $\hat{C}(i,j)$ and $E_c(i,j)$ are all intermediate processing results each of which represents a color. $Y(i,j)$ is the luminance value of color $C'(i,j)$. In the work presented in this Chapter, the luminance value of a color in (r,g,b) format is computed as 0.299r +0.587g+0.114b as in the commonly used *RGB*-to-*YUV* conversion [70] that is defined as

$$\begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{6.1}$$

Based on $Y(i,j)$ and the pre-derived palette, color $C'(i,j)$ is quantized to $\hat{C}(i,j)$, a particular palette color whose corresponding color index value is denoted as $I_{\hat{C}}(i,j)$. $E_c(i,j) = C'(i,j) - \hat{C}(i,j)$ is actually the color quantization error. It will be diffused to the not-yet-processed neighboring pixels of pixel (i,j) in C. After processing all pixels, the index plane composed of $I_{\hat{C}}(i,j)$ for all (i,j) forms the resultant grayscale image. The details of how luminance-constrained color quantization works with error diffusion to provide color to grayscale conversion result will be provided in Section 6.4.

The color palette can be converted to bit sequence, secured with a private key and embedded into the grayscale image or stored separately with it. The details of how to embed the palette in the grayscale image will be described in Section 6.5.

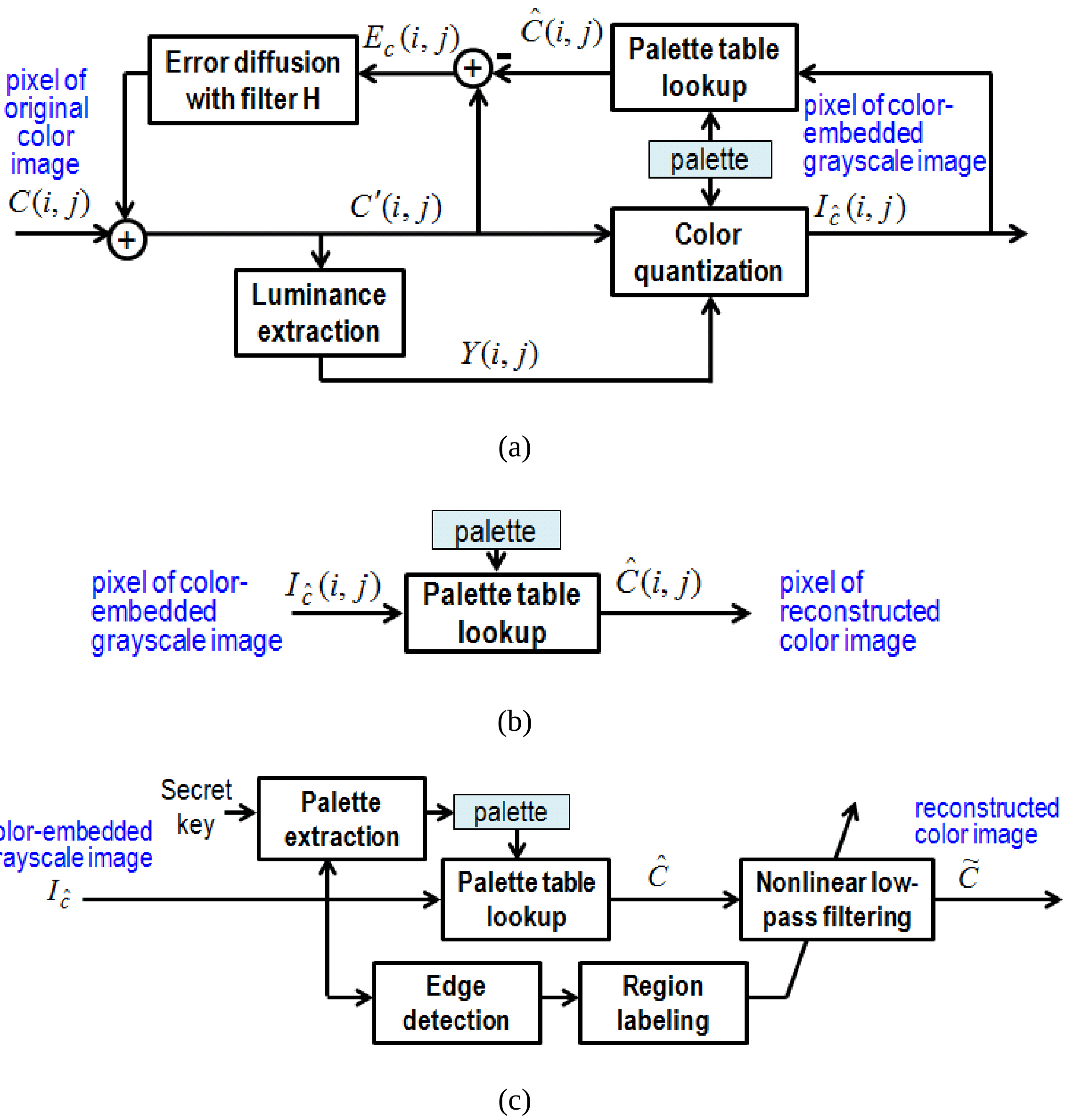


Figure 6-1 Block diagrams of (a) color-to-grayscale conversion, and color image reconstruction (b) without and (c) with halftoning artifacts suppression

When necessary, pixel values of the grayscale image are used as indices to fetch palette colors to reconstruct the color image as shown in Figure 6-1(b). An optional nonlinear low-pass filtering can be further applied to remove the high frequency noise in the recovered color image as shown in Figure 6-1(c). The details of the optional noise removal scheme will be discussed in Section 6.6.

6.3 Color palette generation

To fit into our application, the developed palette should bear two properties. First, palette colors should be sorted according to their luminance values and indexed in a way that their luminance values are roughly proportional to their index values. This allows the index plane to appear as a grayscale image. Second, consecutive colors in the palette should form a three-dimensional enclosure in the color space to cover as many pixel colors $C(i,j)$ that have the same luminance values as the involved palette colors as possible. Theoretically, with the halftoning technique, any specific color inside the enclosure can be rendered with the palette colors that form the enclosure as long as the smooth region having the color to be rendered is large enough in image *C*.

There are a number of studies on how to generate a color palette based on a given color image, such as k-mean clustering [73], fuzzy c-mean clustering [72], median-cut [82] and octree decomposition [83]. Some studies also make an assumption that halftoning will be carried out in color quantization [74, 75, 84]. However, none of them is developed to have the aforementioned properties because their palettes are optimized to provide a high quality color-quantized image and no reversible grayscale image is involved. A tailor-made palette is hence developed as follows to fit into our application.

Figure 6-2(a) shows the RGB color cube in *RGB* color space. All pixel colors that appear in color image *C* should be in the RGB cube. To derive our palette, we stand the cube on the black vertex (0,0,0), with the white vertex (1,1,1) directly above it, as show in Figure 6-2(b). This can be accomplished by rotating the cube about the *b*-axis by π/4 and then about the *r*-axis by θ=arcsin $(\sqrt{2/3})$, where θ is the angle between the achromatic axis (aligned with the line from (0,0,0) to (1,1,1) in the RGB cube shown in Figure 6-2(a)) and the *r*-, *g*- or *b*- axis.

After the rotations, a color in (*r*,*g*,*b*) format is mapped to (x, y, z) format as:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} & 0 \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{6.2}$$

where $z \in [0, \sqrt{3}]$ provides the achromatic information of the color and (*x*,*y*) forms a vector that provides the chromatic information of the color. For reference purpose, we denoted a color in (*r*,*g*,*b*) format as $\vec{p}$ and its representation in (*x*,*y*,*z*) format as $\vec{p}' = T\vec{p}$, where T is the transformation matrix given in eqn. (6.2).

Without loss of generality, we assume that a palette is consisted of 256 colors and they are indexed from 0 to 255. To determine the k^{th} palette color, we collect all pixel colors the luminance values of which are in the range of $B_l(k) = \max\,(0, \frac{k-2}{255})$ and $B_u(k) = \max\,(0, \frac{k+3}{255})$, project them onto the two-dimensional *x*-*y* plane as shown in Figure 6-2(b), compute their signed magnitudes of a selected primary color component, and find out the one which gets the maximum. In our derivation, six primary color components including red (R), yellow (Y), green (G), cyan (C), blue (B) and magenta (M) are considered. Figure 6-2(c) provides an example that shows how to compute the signed magnitude of a pixel color's primary red component after it is transformed and projected onto the *x*-*y* plane, where $\vec{p}'_{xy} = (x, y)$ is a vector formed by the components *x* and *y* of $\vec{p}' = T\vec{p}$.

The pixel color having the maximum signed magnitude of a selected primary color component is the outermost pixel color in $\Omega_k = \{\vec{p}\,|\vec{p} \in \text{color image } C$ and $B_l(k) < \text{luminance of } \vec{p} \le B_u(k)\}$ in terms of the distance from the z-axis along a specific direction. To make every six contiguous palette colors form an enclosure to enclose more pixel colors in Ω_k as possible, when picking the primary color component for deriving the k^{th} palette color, we pick the six chromatic vertices (i.e. R, Y, G, C, B and M) in the RGB cube in turns as k increases. In formulation, the k^{th} palette color $\overrightarrow{c_k}$ is given as

$$\overrightarrow{c_k} = \arg\max_{\vec{p}\in\Omega_k}\left(\vec{p}'_{xy}\cdot\vec{u}_{mod(k,6)}\right) \quad \text{for } k=0,1\ldots255 \tag{6.3}$$

where $\cdot$ is the dot product operator, *mod*(k,6) means k modulo 6, and $\vec{u}_i$ for i=0,1…5 are the unit direction vectors shown in Figure 6-2(c).

With this arrangement, the k^{th} palette color is the outermost color from the origin of the *x*-*y* plane along a specific direction among all image pixel colors whose luminance values belong to $[B_l(k), B_u(k)]$ as shown in Figure 6-3. Besides, every 6 consecutive palette colors form an enclosure that contains almost all image pixel colors that have the same luminance levels with the palette colors. The enclosure may not be able to enclose all image pixel colors, but those outside the enclosure are very close to the enclosure boundary. That means, for any given image pixel color $\vec{p}$, we can use 6 palette colors to render it and their luminance errors from $\vec{p}$ are bounded in [-2/255, 3/255].

This capability cannot be achieved with conventional color palettes even though error diffusion is used to render $\vec{p}$. It is because there may not be palette colors having similar luminance levels and there can be a lot of image pixel colors far outside the enclosure formed with the palette colors having similar luminance levels (e.g. the enclosure in red) as shown in Figure 6-3.

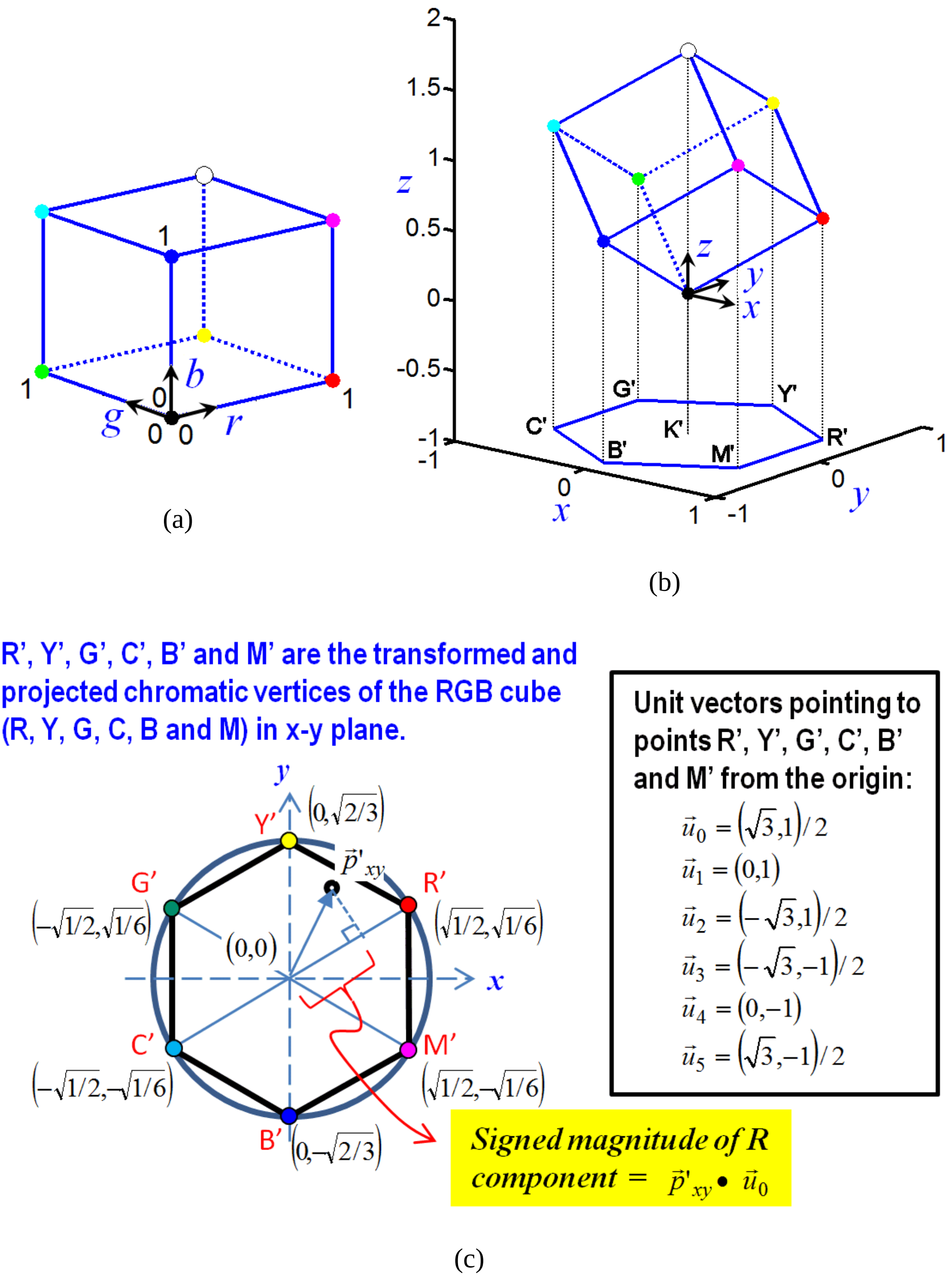


Figure 6-2 Different domains for palette generation: (a) RGB color cube in *rgb* space, (b) transformed RGB color cube in *xyz* space and (c) the x-y plane used to derive a palette color

Figure 6-4 shows the consequences of using two palettes of different properties to reconstruct a color image. As shown in Figure 6-4(b), even though error diffusion is exploited, there can be serious color shift in the recovered color image if the used palette does not have the aforementioned properties.

In general, we can design a palette that uses *d* consecutive palette colors to form an enclosure to enclose image pixel colors whose luminance levels are in $\left[max\left(0,(k-\frac{d}{2}+1)/255\right),min\left(1,(k+\frac{d}{2}+1)/255\right)\right]$. It can be achieved by using *d* different unit direction vectors $\vec{u}_i$ alternately when determining the k^{th} palette color with eqn.(6.3). For comparison, we developed three palettes for Figure 6-4(a) under three different settings (*d*=3, 6 and 9). Figure 6-5 shows the enclosures formed with the three palettes when *k*=100. One can see that a triangle can leave many pixel colors outside while using a dodecagon instead of a hexagon helps little to enclose more pixel colors. In view of this, we select *d*=6 in our realization.

Though the palette generation process seems to be complicated, it can be realized easily as the involved projection and transformation are simple and the chromatic vertices of the RGB cube only contain component values 0 and 1. After simplification, eqn. (6.3) can be rewritten as

$$\vec{c}_k = \arg\max_{\vec{p}\in\Omega_k}\left(\vec{v}_{mod(k,6)}\begin{bmatrix}2 & -1 & -1\\ -1 & 2 & -1\\ -1 & -1 & 2\end{bmatrix}\begin{bmatrix}r_p\\ g_p\\ b_p\end{bmatrix}\right) \quad \text{for } k = 0,1\ldots 255 \qquad (6.4)$$

where
$$\begin{cases}\vec{v}_0 = [1 \quad 0 \quad 0]\\ \vec{v}_1 = [1 \quad 1 \quad 0]\\ \vec{v}_2 = [0 \quad 1 \quad 0]\\ \vec{v}_3 = [0 \quad 1 \quad 1]\\ \vec{v}_4 = [0 \quad 0 \quad 1]\\ \vec{v}_5 = [1 \quad 0 \quad 1]\end{cases} \qquad (6.5)$$

are six unit vectors and $[r_p, g_p, b_p]$ is pixel color $\vec{p}$ in (r,g,b) format.

In case Ω_k is empty for a particular k, palette color $\vec{c}_k$ for that index value k is determined as

$$\vec{c}_k = \arg\max_{\vec{p}\in\Omega'_k}\left(\vec{p}'_{xy}\cdot\vec{u}_{mod(k,6)}\right) \qquad \text{for } k = 0,1\ldots255 \qquad (6.6)$$

where $\Omega'_k = \{\vec{p}|\vec{p} \in$ surface of the RGB color cube and $B_l(k) \leq$ luminance of $\vec{p} < B_u(k)\}$. It is image independent and hence can be precomputed.

Unlike other VQ-based RCGC algorithms [27–31], the proposed palette generation algorithm is not an iterative algorithm. The whole palette can be determined by scanning the image once and hence it is able to support real-time applications.

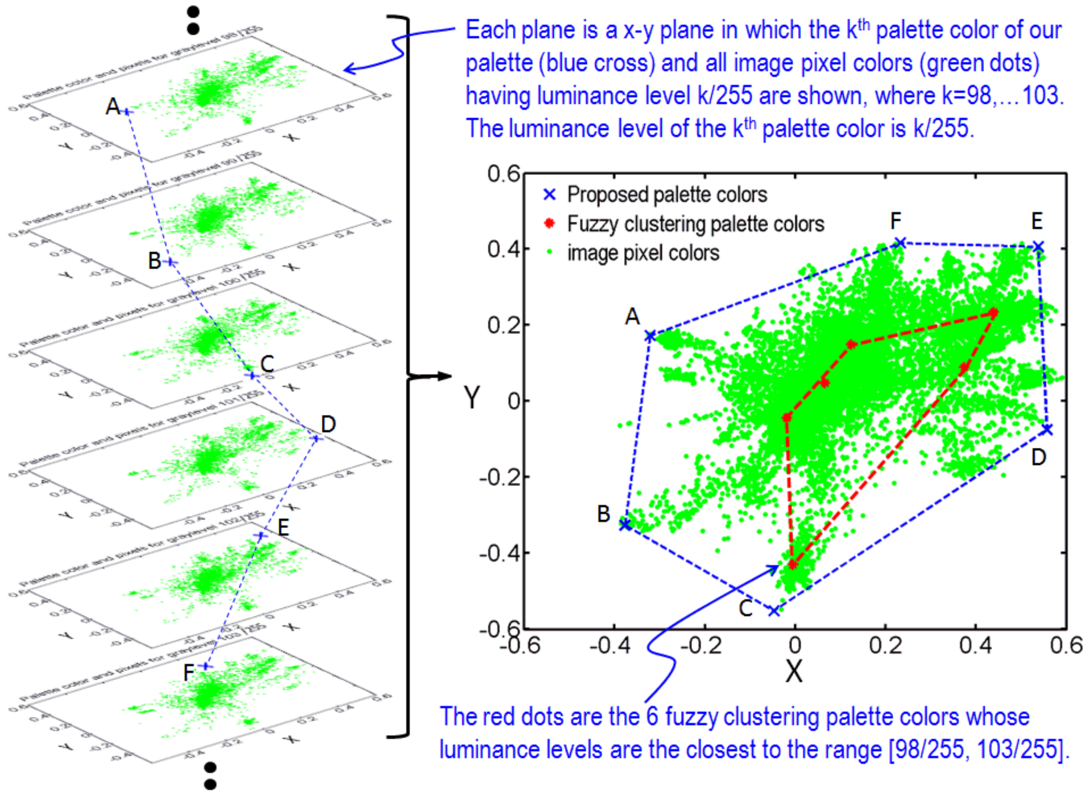
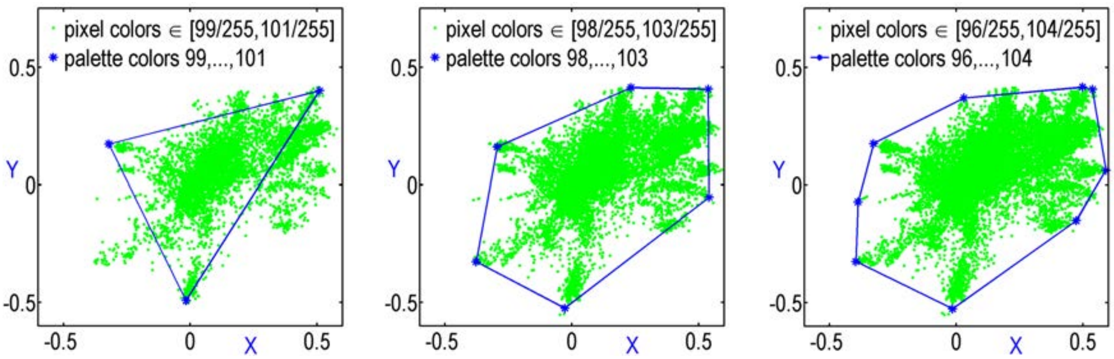


Figure 6-3. An example showing how 6 consecutive palette colors in our palette form an enclosure to cover most image pixel colors having similar luminance levels. The palette is developed based on the image shown in Figure 6-4(a).

(a) (b) (c)

Figure 6-4. Results of using different palettes in the proposed RCGC algorithm to reconstruct a color image: (a) original, (b) a palette obtained with fuzzy clustering [72] and (c) the proposed palette.



(a) (b) (c)

Figure 6-5. Enclosures formed with d palette colors when the proposed palette is developed under different settings: (a) d=3, (b) d=6, and (c) d=12. The data is obtained for k=100 and the palettes are developed for the image shown in Figure 6-4(a).

6.4 Luminance-constrained color quantization and error diffusion

In our realization of RCGC, luminance-constrained color quantization and error diffusion are performed with the color palette generated in Section 6.3. The feasibility of luminance-constrain color quantization relies on the palette property that a palette color's index value is roughly proportional to its luminance value.

For any input color $C^{'}(i,j)$, its luminance value $Y(i,j)$ is computed and then used to confine a subset of the palette colors to one of which we can quantize $C^{'}(i,j)$. In particular, the selected subset of palette colors is given as

$$\Gamma = \{\overrightarrow{c_k}| = 5 \leq k - 255Y(i,j) \leq 6 \; and \; k \in [0,255]\} \tag{6.7}$$

Unlike conventional color quantization algorithms that search for the best color from 256 palette colors, the proposed algorithm only searches at most 12 palette colors. It significantly reduces the realization effort and bounds the maximum luminance distortion of a pixel.

As CIEL*a*b* color space is device independent and perceptually uniform, we select the Euclidean distance in CIEL*a*b* space as the quantization criterion. In formulation, we have

$$\hat{C}(i,j) = \arg\min_{\vec{c}_k \in \Gamma} \|T_{RGB2LAB}(C'(i,j)) - T_{RGB2LAB}(\vec{c}_k(i,j))\| \tag{6.8}$$

where $T_{RGB2LAB}(.)$ is the transform from *rgb* color space to CIEL*a*b* color space and ||•|| denotes the *L2 norm*. The index value of $\hat{C}(i,j)$ is denoted as $I_{\hat{C}}(i,j)$.

After quantizing $C^{'}(i,j)$, its quantization error

$$E_c(i,j) = C^{'}(i,j) - \hat{C}(i,j) \tag{6.9}$$

is diffused to pixel (i,j)'s neighboring pixels. The diffusion affects the original colors of its neighboring pixels. It explains why, when processing pixel (i,j), we color-quantize $C'(i,j)$ instead of $C(i,j)$. Specifically, after taking all the color errors introduced by the processed neighbors into account, pixel (i,j)'s color is modified to be

$$C'(i,j) = C(i,j) + \sum_{(m,n)\in\Phi} E_c(i-m, j-n)h(m,n) \tag{6.10}$$

at the moment we process pixel (i,j), where $\Phi = \{(m,n) \mid m=0,1 \text{ and } n=-1,0,1\}$ is the support of the diffusion filter H and $h(m,n)$ is the $(m,n)^{th}$ coefficient of filter H for $(m,n) \in \Phi$. In our realization, the standard diffusion filter [39] is adopted. This diffusion filter is commonly used in binary halftoning. The diffusion process is performed in channels *r*, *g* and *b* separately.

6.5 Color palette embedding

For applications where color information needed to be protected, the palette can be embedded into the grayscale image (i.e. the index plane $I_{\bar{C}}$) with encryption. It is also helpful to reduce the bit overhead. Conventional VQ-based RCGC algorithms [27–31] adopt the least significant bit substitution (LSB) method [85] to embed the bit sequence of the color palette. The same method is adopted here for the same purpose.

In particular, the whole palette is rearranged to form a bit sequence of length L. The index plane is then partitioned into L non-overlapped blocks. From each block, a pixel is selected to carry a bit from the bit sequence. A pseudo-random number generator (PRNG) with a secret key is applied to select the pixels into which the palette information is embedded.

Embedding the bit sequence into the index image will introduce some noise. The index plane is sensitive to noise because two colors of adjacent indexes can be very different. Therefore, we take the palette-embedding process into account during the color-to-grayscale conversion to minimize its distortion to the conversion output. For pixels that are not selected to carry the palette information, they are processed as mentioned before. For those are selected, they are color-quantized differently as follows:

$$\hat{C}(i,j) = \arg\min_{\vec{c}_k \in \Gamma_s} \| T_{RGB2LAB}(C'(i,j)) - T_{RGB2LAB}(\vec{c}_k(i,j)) \| \qquad (6.11)$$

where s∈{0,1} is the value of the bit embedded and

$$\Gamma_s = \{\vec{c}_k | = mod(k,2) = s \text{ and } \vec{c}_k \in \Gamma\} \quad \text{for } s\text{=0,1} \qquad (6.12)$$

As the error introduced by palette-embedding will be diffused with the quantization error to the neighboring pixels, its spectrum will also be shaped and its impact on the visual quality of the output images can be minimized.

Palette extraction is easy with the secret key. The secret key allows us to locate the pixels that carry the embedded information and their least significant bits form the bit sequence for us to reconstruct the palette.

The basic idea of the method used here stems from the ±1 LSB embedding algorithm [85]. This algorithm is widely applied in image steganography. According to the investigation in [31], this method is simple and efficient enough to protect the color information from being acquired by unauthorized people without a security key. Since we are not doing watermarking, whether the embedded information can stand for attacks is not the issue.

6.6 Halftoning artifacts suppression

Figures 6-6(a) and 6-6(d) show, respectively, $I_{\hat{C}}$, the color-embedded grayscale image produced with the proposed algorithm, and $\hat{C}$, the color image directly reconstructed by using $I_{\hat{C}}$ as the index plane and the palette embedded in $I_{\hat{C}}$. As shown in Figure 6-6(d), there is no artifact such as color shift and false contour in $\hat{C}$, and its visual quality is already good at a reasonable viewing distance. The error diffusion step shifts the quantization noise to high frequency band such that the noise can be removed by the low-pass filtering effect of human eyes. However, the high frequency noise can still be visible when we get too close to view the image. A descreening scheme is hence applied to alleviate this problem.

At first glance, bilateral filtering [86] seems to be good for handling the situation as it is designed to preserve sharp edges and remove noise simultaneously. However, our simulation results show that it does not work properly in this scenario. After error diffusion, the quantized luminance value of a pixel can shift -5 to 6 levels from the original. As the grayscale of a pixel is used as an index to determine the color of the pixel, a minor luminance difference among adjacent pixels can lead to a significant chrominance difference. Pixels on the same side of an edge may not have similar chromatic values and hence bilateral filtering does not work properly.

In our solution, an edge-aware low-pass filter is used to remove the noise as follows:

Step 1: Apply Canny edge detector [87] to the color-embedded grayscale image to get a binary edge map of the image.

Step 2: Label different regions according to the edge map.

Step 3: Derive a spatial-variant low-pass filter $F_{(i,j)}$ for pixel (i,j) as

$$F_{(i,j)}(m,n) = \begin{cases} Ae^{-\left(\frac{m^2+n^2}{\sigma^2}\right)} & \text{if pixels } (i,j) \text{ and } (m,n) \text{ are in the same region} \\ A\mu e^{-\left(\frac{m^2+n^2}{\sigma^2}\right)} & else \end{cases}$$

$$\text{for} \quad |m|,|n| \leq 4 \qquad (6.13)$$

where $F_{(i,j)}(m,n)$ is the $(m,n)^{th}$ filter coefficient of filter $F_{(i,j)}$, σ controls the cut-off frequency of the filter, A is a normalization factor that makes the weights in the kernel sum to one, and μ∈[0,1] is a controlling parameter that controls the contribution of the pixels from different regions. Specifically, μ=1 means that the edge information is ignored and the filtering is linear, and μ=0 means that pixels not in the same region with the pixel being processed are ignored.

Step 4: Apply filter $F_{(i,j)}$ to the chromatic channels of the reconstructed color image as follows.

$$a^*(i,j) = \sum_{|m|,|n|\le 4} F_{(i,j)}(m,n)a^*(i+m,j+n)$$

$$b^*(i,j) = \sum_{|m|,|n|\le 4} F_{(i,j)}(m,n)b^*(i+m,j+n)$$

$$\text{for all } (i,j) \tag{6.14}$$

where $a^*(i,j)$ and $b^*(i,j)$ are, respectively, the $(i,j)^{th}$ pixels of channels a* and b* of color image $\hat{C}$ in CIEL*a*b* color space.

Note that the luminance channel is not filtered so as to preserve the texture information of the image. The block diagram shown in Figure 6-1(c) shows the reversion of our reversible color-to-grayscale conversion when halftone artifacts suppression is exploited. The final reconstructed color image is $\tilde{C}$.

Figure 6-6 illustrates step-by-step how $\tilde{C}$ is reconstructed with $I_{\hat{C}}$. Figure 6-6(b) is the edge map acquired from the index image shown in Figure 6-6(a). As a reference, Figure 6-6(c) shows an edge map obtained with the original image. One can see that most of the edges can be successfully extracted based on $I_{\hat{C}}$. It is hence reliable to utilize the edge information in nonlinear low-pass filtering.

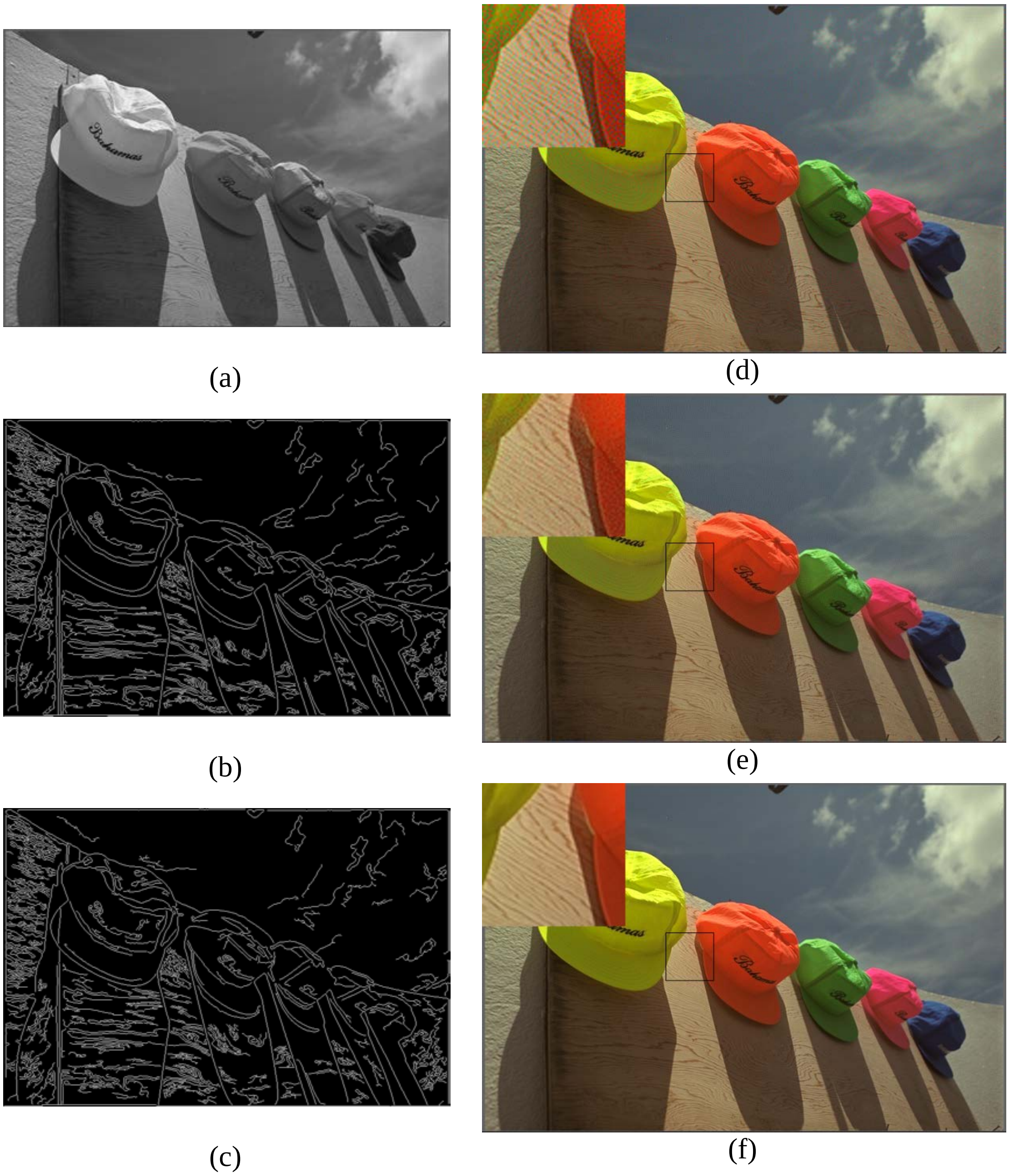


Figure 6-6 Recovering color image from our color-embedded grayscale image: (a) color-embedded grayscale image $I_{\hat{C}}$; (b) edge map obtained with (a); (c) edge map obtained with the original image C; (d) color image obtained before non-linear low-pass filtering; (e) color image obtained after nonlinear low-pass filtering; (f) original color image.

Figures 6-6(d), (e) and (f) show $\hat{C}$, $\tilde{C}$ and the original image C respectively. On their left upper corners, the enlarged versions of the regions enclosed in black boxes are displayed for better inspection. One can see that the high frequency impulse noise in $\hat{C}$ can be efficiently reduced after filtering, while the texture information in the original image is preserved.

6.7 Performance evaluation

This Section provides some simulation results and experimental data for evaluating the performance of the RCGC algorithm presented in this Chapter.

6.7.1 Parameters used in simulation

In Section 6.6, a descreening process is suggested to suppress the artifacts caused by the halftoning process of our RCGC algorithm. There are two parameters for controlling the spatial-variant low-pass filter $F_{(i,j)}$ used in the descreening process. In particular, parameter σ controls the cut-off frequency of low-pass filtering, and parameter μ controls the contribution of the pixels from different regions. These two parameters were experimentally trained based on the 24 true-color images in Kodak set [71] by minimizing objective function:

$$J_t = \left\|C - \tilde{C}\right\|^2 \qquad (6.15)$$

where C and $\tilde{C}$ are, respectively, the original color image and the recovered color image after filtering. Figure 6-7 shows how J_t changes with σ and μ. It is found that J_t reaches its minimum at σ=1.22 and μ=0.7. All subsequent simulation results reported in this Chapter were obtained with these settings. However, as shown in Figire 6-7, J_t is actually insensitive to σ and μ when σ∈[0.8,2.0] and μ∈[0.2,1.0]. Hence, some other settings of σ and μ can also be used to achieve the same performance.

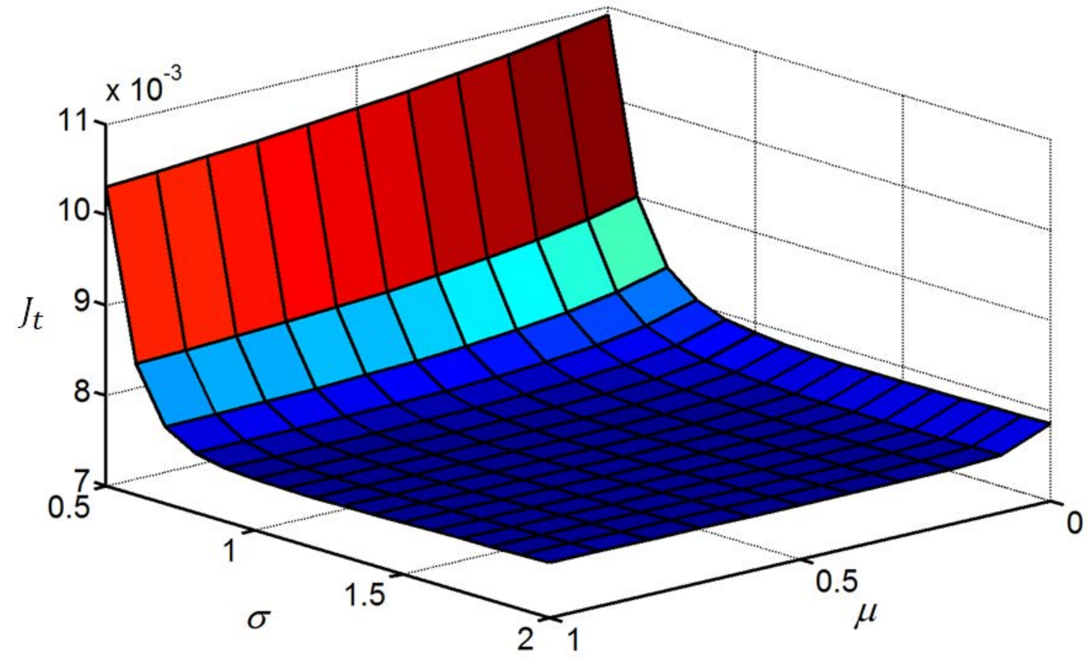


Figure 6-7 The plot of objective function J_t vs. σ and μ

6.7.2 State-of-art RCGC algorithms used in comparison

Several other RCGC algorithms, including Queiroz's [35], Ko et al.'s [34], Horiuchi et al.'s [36], Tanaka et al.'s [30] and Chaumont et al.'s [31], were also realized for comparison. Among them, the first three are SE-based RCGC algorithms while the others are VQ-based RCGC algorithms. When realizing Queiroz's [35], a 4×4 DCT was applied, and the highest 6 frequency bands were used to embed the duplicated color planes in *YUV* color space. When realizing Ko et al.'s [34], a four-level wavelet packet transform was implemented, and the two subbands with minimum amount of information were used to embed color. Accordingly, two chromatic planes were subsampled by four in each direction to fit the size of the high frequency blocks. In the realization of Horiuchi et al.'s [36], one-level discrete wavelet transform was applied and the chroma information was downsampled by 16 in each direction. For all the three SE-based RCGC algorithms, bilinear kernel was applied for upsampling when reconstructing a color image. When realizing Chaumont et al.'s [31], the number of palette colors is determined adaptively as suggested in [31]. As for the other VQ-based RCGC algorithms including [30] and ours, the palette size was fixed to be 256.

6.7.3 Simulation results

Two sets of testing color images are used. One is the Kodak set [71] that includes 24 color images of size 768×512 or 512×768. The other one includes 12 images shown in Figure 6-8. They are available from *'Miscellaneous'* image set of the USC-SIPI Image Database [88]. Each of them is of size 512×512.

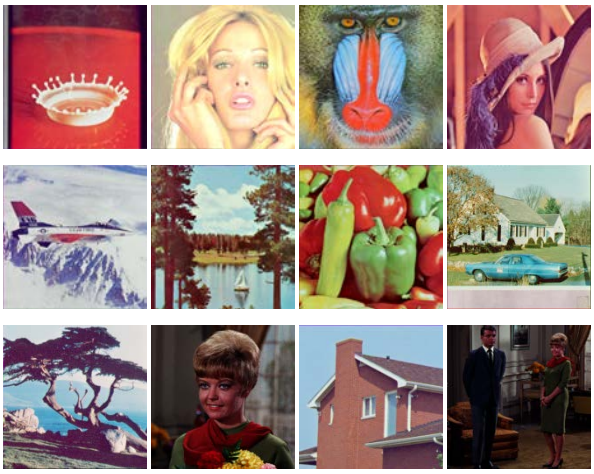


Figure 6-8 Test image set 2

6.7.3.1 Impact on palettes

Figure 6-9 shows the color palettes developed in different VQ-based RCGC algorithms ([30], [31] and ours) for test images '*powder*', '*ape*' and '*woman*'. One can easily find that the palette colors in the palettes of [30] and [31] are more biased in hue, while ours are more scattered in the color space. Moreover, in Chaumont et al.'s palette [31], similar colors are usually assigned similar index values, while the proposed palette crawls colors of different hues in turns.

Our palette generation method picks pixel colors of similar luminance levels and then selects the outermost ones in the *x*-*y* plane as palette colors. As a consequence, for rich color images such as *Powder* and *Ape*, the created palette contains colors near R, Y, G, C, B and M color vertices in turns as shown in Figure 6.9. Figure 6-10 provides another view of the picture by showing the distributions of all our palette colors in the *x*-*y* plane for different testing images.

The advantage of this arrangement is that most pixel colors in the image can be enclosed by some palette colors of similar luminance levels. By exploiting the error diffusion technique, the proposed RCGC algorithm is able to render much more than 256 colors and preserve the spatial texture of an image.

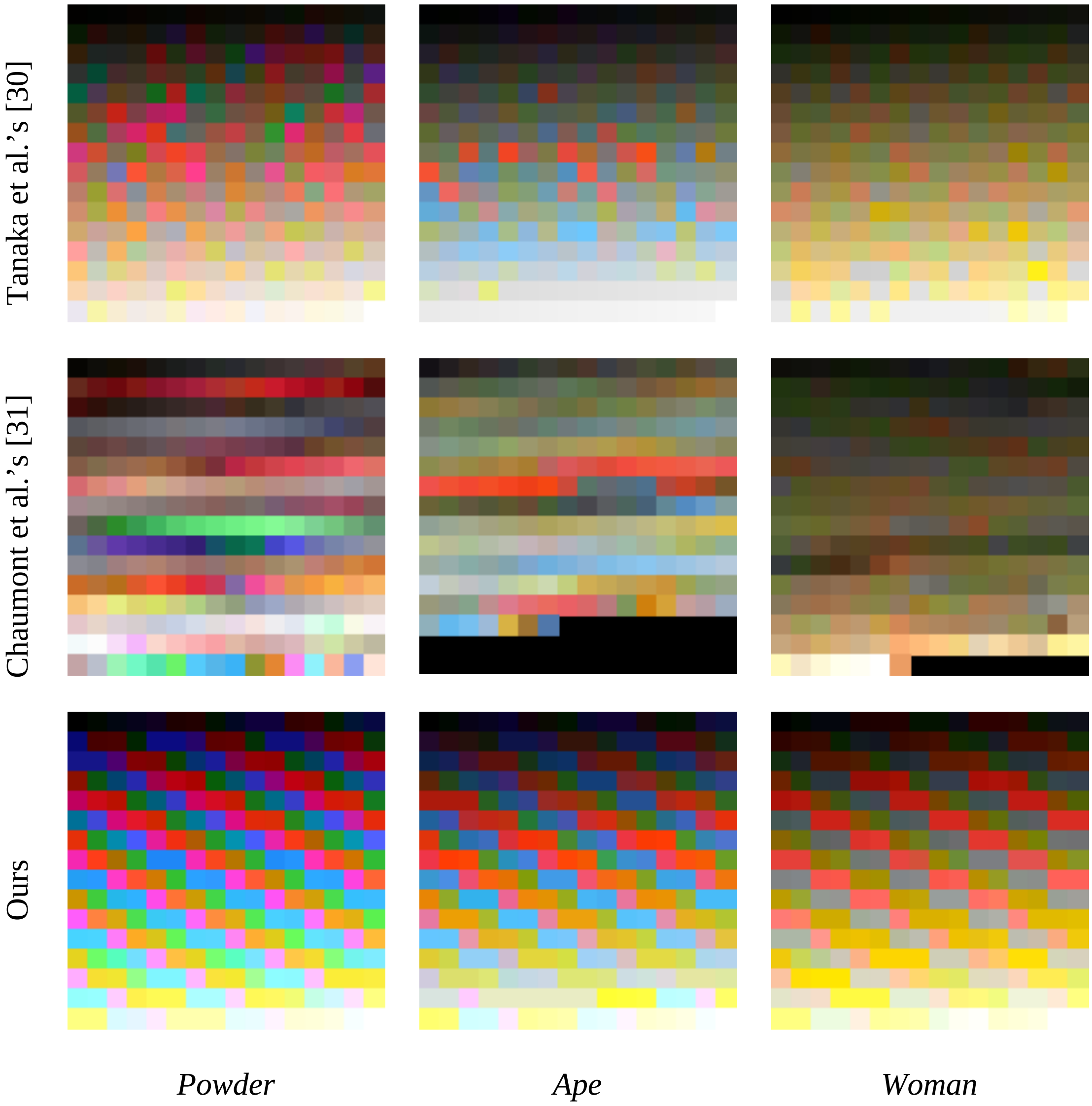


Figure 6-9 Palettes generated with different RCGC algorithms for test images *Powder*, *Ape* and *Woman*.

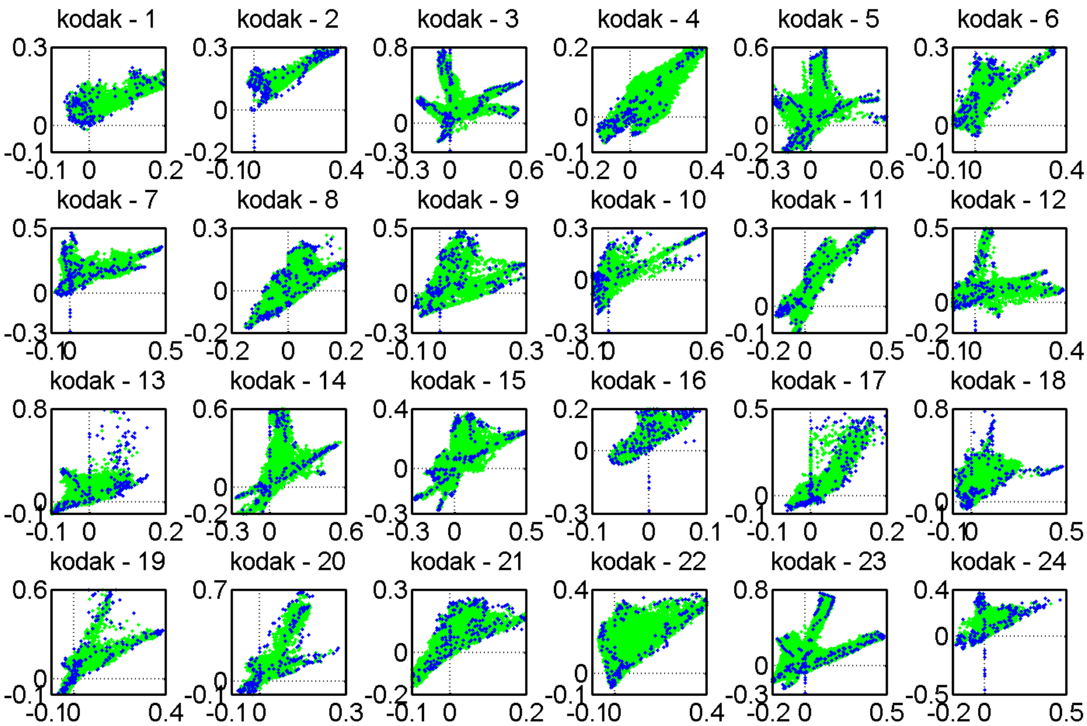


Figure 6-10 Distributions of image pixel colors and our palette colors in *x*-*y* space for different test images in Kodak image set. Green dots and blue dots denote the image pixel colors and the palette colors respectively.

6.7.3.2 Subjective comparison

Figures 6-11 and 6-12 show, respectively, the color-embedded grayscale images and the recovered color images of test image *Woman* obtained with various evaluated RCGC algorithms for visual evaluation.

As shown in the two figures, we may observe the artifacts introduced by the state-or-art RCGC algorithms as described in Section 2.4. In particular, image blurring is a common issue of SE-based algorithms [34–36] because some texture information is discarded to make room for embedding the color information. Moreover, the embedding process gives rise to another issue that serious pattern noise can appear in the color-embedded grayscale images as shown in Figures 6-11(a) and (b). This pattern noise is comparatively less severe in Horiuchi et al.'s output [36] (shown in Figure 6-11(c)) because it preserves some texture information at the cost of chromatic information. However, the pattern noise is still visible in the neck and the forehead.

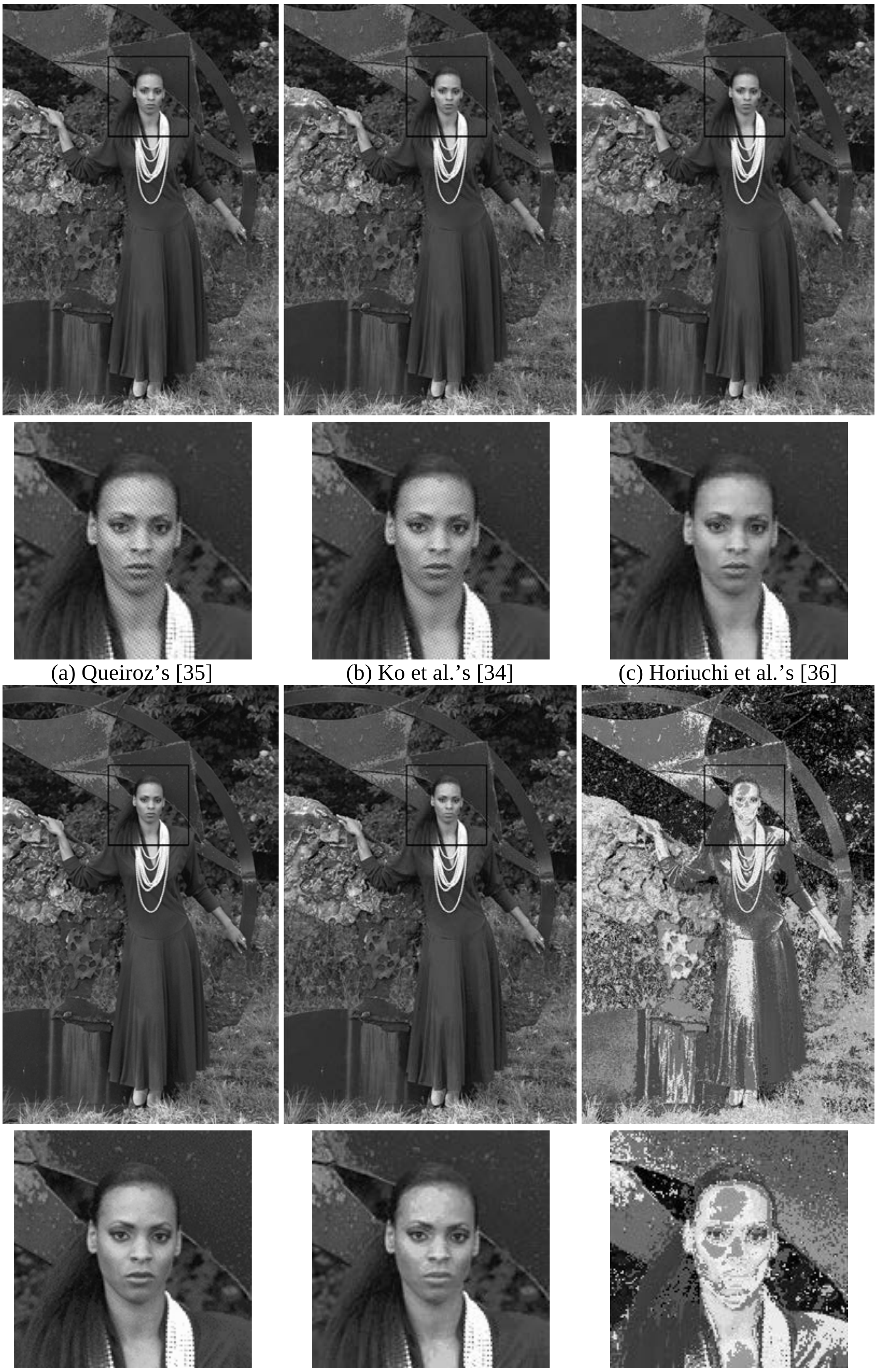
When inspecting the recovered color images of SE-based algorithms [34–36], one can see color shift in some local regions (e.g. pearl necklaces in Figures 6-12(a) and 6-12(b), and the skin and lip in Figure 6-12(c)). Figure 6-12(b) is obviously blurred as compared with the original test image shown in Figure 6-13. Figure 6-12(a) actually does not preserve original texture details properly. One can see in Figure 6-12(a) that the eye and the ear on the left are seriously distorted by local high frequency noise.

In contrast, state-of-art VQ-based RCGC algorithms [30, 31] have some different issues. Theoretically, VQ-based algorithms can achieve good mean square error (MSE) performance since their palettes are optimized by clustering algorithms to minimize the quantization error. However, without the help of noise shaping, the limited number of palette colors generally results in false contour and color shift. As shown in Figures 6-12(e) and 6-12(f), one can easily see the color shift on the lip and the false contours on the face.

Another issue of conventional VQ-based RCGC algorithms is that there is no linear relationship between the index value and the luminance value of a palette color. One can observe that the color-embedded grayscale image in Figure 6-11(f) is very different from the ground truth. Tanaka et al.'s result [30] looks better as shown in Figure 6-11(e) since it introduces a lightness constraint into the clustering process when optimizing its color palette. However, false contour can still be easily observed.

By exploiting the error diffusion technique, the proposed algorithm is able to render much more than 256 colors and preserve the spatial texture of an image. As shown in Figure 6-12(d), our recovered color image is sharp and there is no color shift or saturation loss. The noise carried in the recovered color image is high frequency noise that is invisible from a reasonable distance. One can also see from Figure 6-11(d) that our color-embedded grayscale image is very close to the ground truth, which increases the security of the hidden color information, as a visually normal grayscale image does not alert people.

Figures 6-14 and 6-15 provide the simulation results of another test image for inspection and similar observations can be obtained. One can find severe pattern noise in Figures 6-14(a)-(c), false contours in Figures 6-14(e)-(f), color noise or blurring at the edges in Figures 6-15(a)-(c), and color shift in Figures 6-15(e)-(f). From this set of simulation results, we have an extra observation that the performance of Tanaka et al.'s [30] is poor when handling this test image. The additional lightness constraint adopted in its palette generation scheme makes the palette difficult to handle wide-spread colors and hence the color shift in the recovered color image can be very severe.



(a) Queiroz's [35] (b) Ko et al.'s [34] (c) Horiuchi et al.'s [36]

(d) Proposed (e) Tanaka et al.'s [30] (f) Chaumont et al.'s [31]

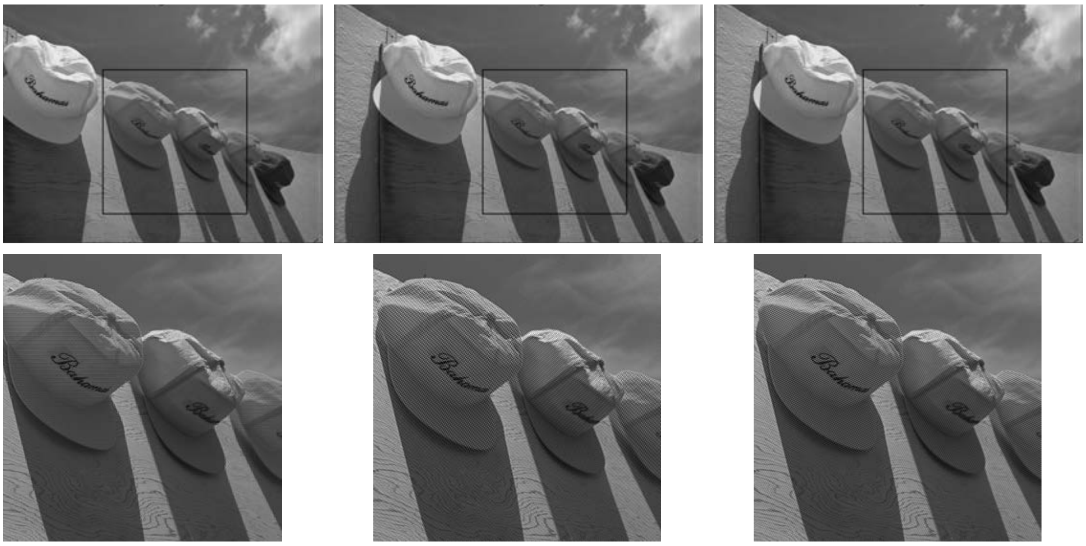
Figure 6-11 Color-embedded grayscale images obtained with various algorithms (Test image: *Woman*)

(a) Queiroz's [35] (b) Ko et al.'s [34] (c) Horiuchi et al.'s [36]

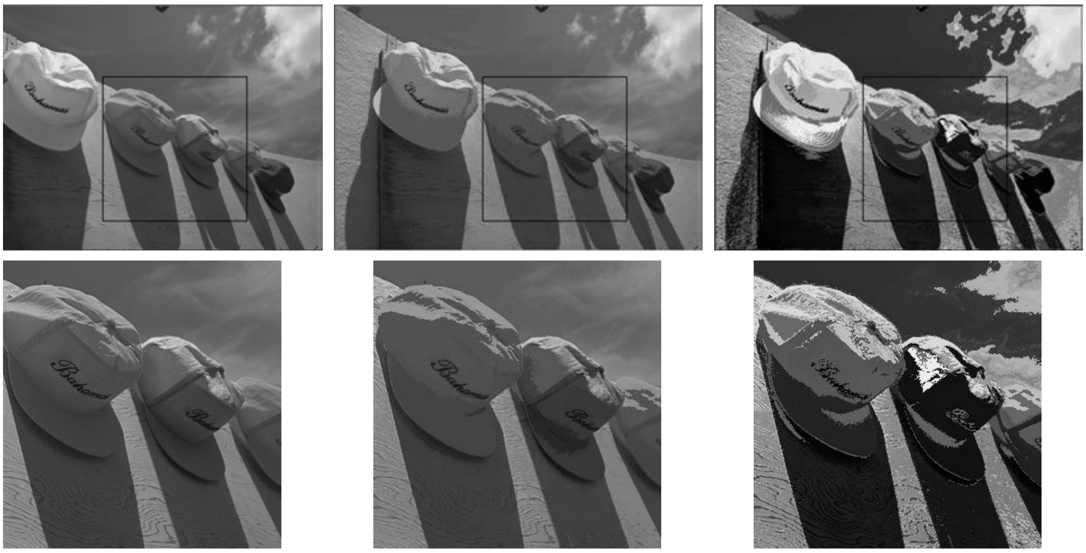
Figure 6-12 Recovered color images obtained with various algorithms (Test image: *Woman*)

(a) Original (b) Luminance plane of original

Figure 6-13. Ground truth for Figures 6-11 and 6-12 (Test image: *Woman*)

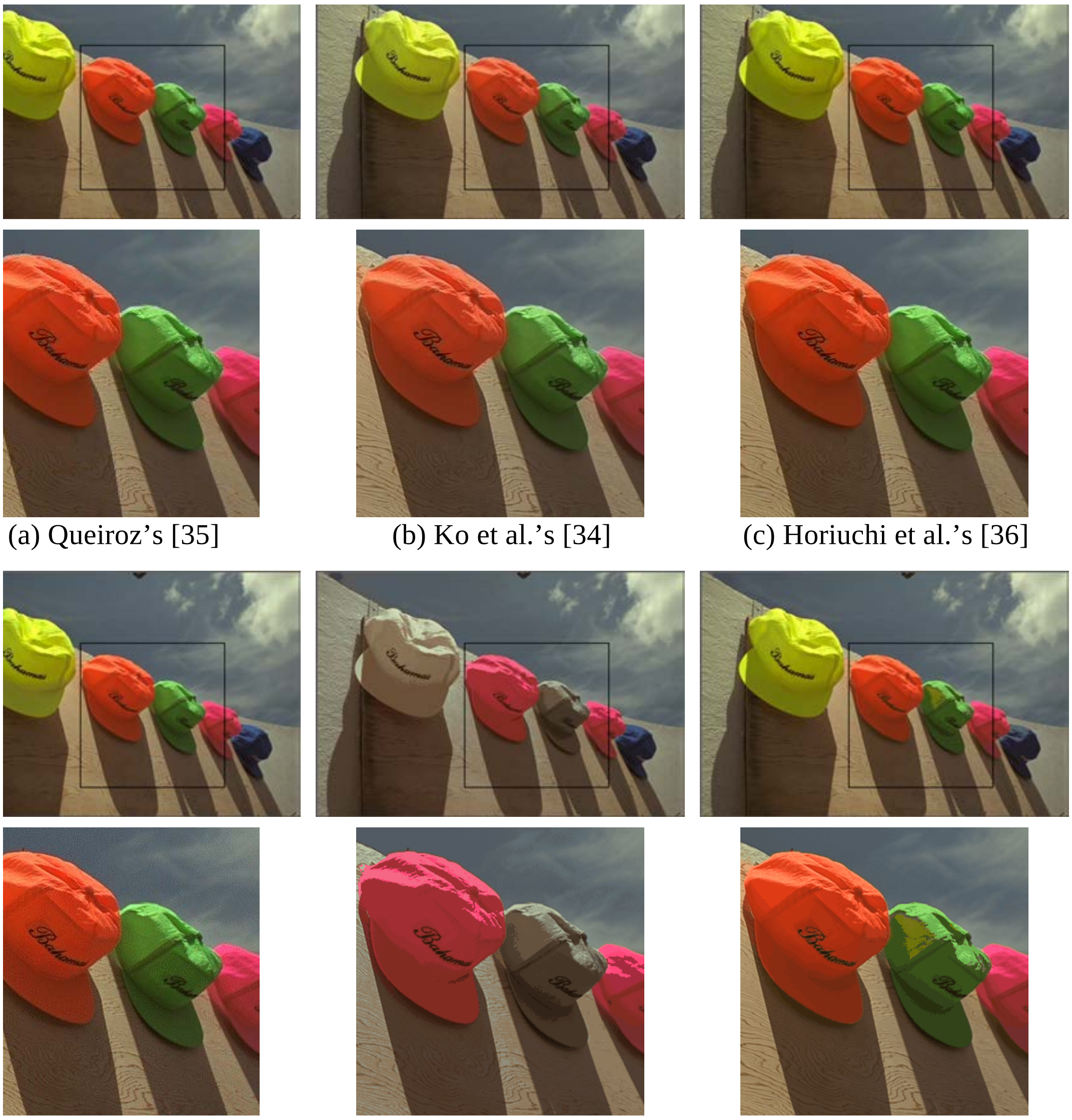


(a) Queiroz's [35] (b) Ko et al.'s [34] (c) Horiuchi et al.'s [36]



(d) Ours (e) Tanaka et al.'s [30] (f) Chaumont et al.'s [31]

Figure 6-14 Color-embedded grayscale images obtained with various algorithms (Test image: *Hats*)



(a) Queiroz's [35] (b) Ko et al.'s [34] (c) Horiuchi et al.'s [36]

(d) Ours (e) Tanaka et al.'s [30] (f) Chaumont et al.'s [31]

Figure 6-15 Recovered color images obtained with various algorithms (Test image: *Hats*)

6.7.3.3 Objective comparison

The objective of RCGC algorithm is to produce high quality color-embedded grayscale image that allows one to reconstruct high quality color image. In view of this, the quality of both the color-embedded grayscale image and the recovered color image should be taken into account when measuring the performance of the evaluated RCGC algorithms.

In particular, we measure the color-embedded grayscale images in terms of *PSNR*, *SSIM* [89] and *HVS-PSNR* [90], and measure the recovered color images in terms of *CPSNR*, *CSSIM* [91], *HVS-PSNR*$_{Color}$, CIEL*a*b* color difference ΔE, and s-CIELAB distance $\Delta E_{s-CIELAB}$ [92, 93].

CPSNR is defined as

$$CPSNR = -10log_{10}\left(\frac{1}{3MN}\left\|C - \tilde{C}\right\|^2\right) \quad (6.16)$$

where $M \times N$ is the image size. CIEL*a*b* color difference ΔE is the Euclidean distance between two images in CIEL*a*b* color space:

$$\Delta E = \sqrt{(L_2^* - L_1^*)^2 + (a_2^* - a_1^*)^2 + (b_2^* - b_1^*)^2} \quad (6.17)$$

where L_1^*, a_1^* and b_1^* are the components of C and L_2^*, a_2^* and b_2^* are the components of $\tilde{C}$ in CIEL*a*b* color space.

HVS-PSNR$_{Color}$ is an extension of *HVS-PSNR* for evaluating the quality of a recovered color image. It is defined as:

$$HVS - PSNR_{Color} = 10log_{10}\left(\frac{3MN}{\|F_G \otimes C - \tilde{C}\|^2}\right) \quad (6.18)$$

where F_G specifies the CSF model that simulates our human visual system [1], and $\otimes$ symbolizes the 2-D convolution operation.

For assessments that involve human visual system model (*HVS-PSNR*, *HVS-PSNR*$_{Color}$, $\Delta E_{s-CIELAB}$ and *CSSIM*), we consider two cases where the resolutions are, respectively, 100 and 150 *dpi* with a fixed viewing distance of 20 inches.

The performances shown in Tables 6-1 and 6-2 are, respectively, based on the simulation results obtained with testing image sets 1 and 2. Note that the parameters μ and σ for the nonlinear low-pass filtering module used in our algorithm is trained with testing image set 1 only. The data shown in Table 6-2 hence reflect whether the parameters can also work properly with images that do not involved in the training.

The findings based on the data reported in both tables are consistent, so we may consider that the obtained parameters μ and σ are universally valid.

As far as the quality of the color-embedded grayscale image is concerned, the proposed and Horiuchi et al.'s algorithms [36] perform more or less equally well in terms of these measures and they are respectively the best in terms of *PSNR* and *SSIM*. Ko et al. [34] performs best in terms of *HVS-PSNR* for 150 dpi because it produces regular high frequency texture which cannot be observed at a far view distance. It matches our findings in Figure 6-11 and 6-14.

As for the recovered color images, the proposed algorithm gets leading scores in most assessments except *CPSNR*. As we have indicated before, VQ-based algorithms [30, 31] tend to achieve good *CPSNR* performance because their palette is designed to minimize mean squared error. In terms of the scores, among all the algorithms, our algorithm and Chaumont et al.'s algorithm [31] can both reconstruct color images of higher quality. However, by considering that there are false contours and color shift in its recovered image as shown in Figure 6-12(f) and Figure 6-15(f) and the quality of its color embedded grayscale image is very poor as shown in Figure 6-11(f) and Figure 6-14(f), Chaumont et al.'s algorithm is actually inferior to our algorithm.

Method	Color-embedded Grayscale Image				Recovered Color Image							
			dpi=100	dpi=150			dpi = 100			dpi = 150		
	PSNR (dB)	SSIM	HVS-PSNR (dB)	HVS-PSNR (dB)	CPSNR (dB)	ΔE	HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$	HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$
Queiroz's [35]	31.64	0.976	40.92	43.85	31.84	2.80	36.78	0.877	1.75	38.09	0.889	1.63
Ko et al.'s [34]	32.82	0.989	42.88	46.67	32.13	4.46	36.05	0.838	3.43	37.01	0.848	3.27
[illegible] et al.'s [36]	33.20	0.997	39.19	43.66	29.53	3.12	33.69	0.927	1.49	36.02	0.952	1.21
[illegible]	28.07	0.986	29.79	31.19	33.07	3.45	34.78	0.874	2.55	35.51	0.879	2.44
[illegible]	16.72	0.643	17.92	18.47	37.79	1.90	39.94	0.962	1.02	40.99	0.968	0.92
[illegible]	35.87	0.988	41.54	44.01	32.71	2.97	38.04	0.963	0.79	41.24	0.968	0.58

Method	Color-embedded Grayscale Image				Recovered Color Image							
			dpi=100	dpi=150			dpi=100			dpi=150		
	PSNR (dB)	SSIM	HVS-PSNR (dB)	HVS-PSNR (dB)	CPSNR (dB)	ΔE	HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$	HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$
Queiroz's [35]	29.89	0.962	39.59	43.12	28.68	8.52	30.32	0.716	3.11	31.22	0.747	2.85
Ko et al.'s [34]	30.26	0.980	41.18	46.06	29.88	7.69	31.42	0.823	2.51	32.18	0.856	2.19
Horiuchi et al.'s [36]	35.95	0.999	40.87	43.24	29.06	7.83	30.97	0.841	2.32	31.99	0.877	1.96
Tanaka et al.'s [30]	31.31	0.974	32.75	33.56	33.24	5.17	34.85	0.874	1.92	37.22	0.886	1.78
Chaumont et al.'s [31]	16.03	0.627	16.96	17.35	34.41	5.19	36.85	0.919	1.48	38.32	0.929	1.34
Proposed	35.95	0.986	41.33	43.51	31.47	5.43	36.45	0.956	0.97	39.52	0.964	0.72

Table 6-1 Performance of RCGC algorithms for Kodak image set [71]

Table 6-2 Performance of RCGC algorithms for *'Miscellaneous'* image set [88]

As a final remark, we note that the proposed algorithm takes 0.0156 bit per pixel (*bpp)* to carry the embedded palette that carries the chromatic information in Figure 6-

12(d). In fact, this figure is inversely proportional to the image size in terms of number of pixels. When one uses the naive approach that compresses the chromatic information directly and then embeds it as binary data into the grayscale image with a general data hiding algorithm, the chromatic information has to be compressed at a compression ratio of 1026.

6.8 Summary

A novel VQ-based RCGC algorithm that emphasizes the quality of both the color-embedded grayscale image and the recovered color image is presented in this Chapter. The improved quality of the images is mainly achieved by developing a color palette that fits into the application and exploiting error diffusion to shape the quantization noise to high frequency band. Simulation results show that the performance of the proposed algorithm is superior to conventional RCGC algorithms subjectively and objectively in terms of different metrics. This work has been published as a journal paper in [38].

Chapter 7.

Optimizing color palettes for the proposed reversible color-to-grayscale conversion framework

7.1 Introduction

In Chapter 6 we proposed a framework of VQ-based RCGC algorithms in which a halftoning technique is exploited to improve the quality of both the recovered color image and the color-embedded grayscale image. To support the operation of RCGC, a tailor-made color palette is generated based on the given image to bear two critical properties. First, there is a linear relationship between the index value and the luminance value of a palette color. Second, every six consecutive palette colors form an enclosure in the color space to cover as many image pixel colors as possible. The first property allows one to represent the luminance plane of an image with an index plane while the second property allows one to render any colors in the image with palette colors with the help of halftoning techniques.

The simulation results shown in Section 6.7 are encouraging, but we found that for some rich color test images there can be some halftoning artifacts in their recovered color images. That inspires us to think about whether it is possible to develop an even better palette to replace the one developed in Chapter 6. In this Chapter, a new color palette generation algorithm is proposed. In contrast to the color palette generation algorithm proposed in Chapter 6, it considers whether a pixel color in the image can be rendered in practical situations instead of ideal situations. When it optimizes the palette, the contribution of a specific pixel color is weighted according to the extent that it can be effectively rendered with the palette colors by halftoning.

Chapter 7 is organized as follows. In Section 7.2, we present a method to prepare an initial color palette for further optimization. In Section 7.3, we formulate the palette generation problem as an optimization problem. Section 7.4 shows how the optimization is realized with simulated annealing while Section 7.5 shows how the new color palette is incorporated into the framework to realize RCGC. Simulation results for performance evaluation are provided in Section 7.6. A brief summary is provided in Section 7.7.

7.2 Initial color palette generation

In contrast to other VQ-based RCGC algorithms [27–31], the proposed RCGC algorithm adopts a color palette that is developed in the CIELAB space. It is because the components of a color in CIELAB space closely match human perception of lightness and color difference. When a color image is represented in RGB format, a conversion between the RGB space and the CIELAB space (with D65 white point) [94] is required such that the palette can be defined and the color-to-grayscale conversion can be carried out in the CLELAB domain. Hereafter, all colors referred in this Chapter are represented in CIELAB format and the luminance of a color is defined to be the value of its luminance component in CIELAB space.

Without loss of generality, we assume that the palette size is 256. Accordingly, the CIELAB color space is divided into 51 layers along the luminance direction in CIELAB space equally as

$$\mathcal{L}_k = \left\{ c \,\middle|\, \frac{500}{255}k \le \text{luminance of color } c < \frac{500}{255}(k+1) \right\} \quad \text{for } k\text{=0:50} \qquad (7.1)$$

Here we note that the luminance of a color in CIELAB space is bounded in [0,100]. From each layer $\mathcal{L}_k$, five colors are selected to be palette colors. By including the absolute white color (the one whose luminance component is 100) as one of the palette colors, there are totally 51×5+1=256 palette colors. For the sake of reference, we define layer $\mathcal{L}_{51}$ as the extra layer that only contains the absolute white color.

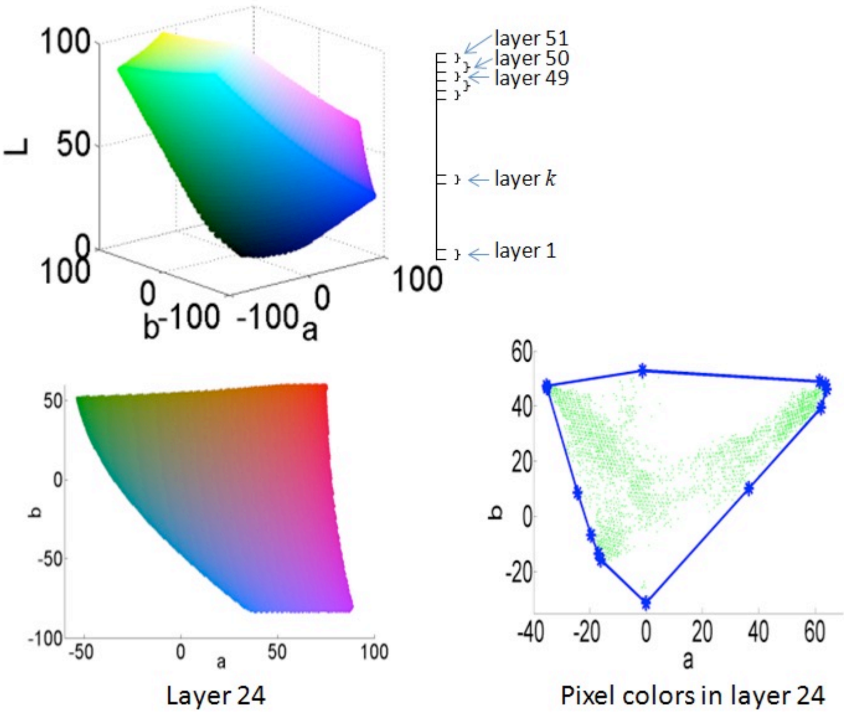


Figure 7-1 (a) The CIELAB color space; (b) collapsed layer $\mathcal{L}_{24}$ (i.e. $\mathcal{L}'_{24}$); (c) the distribution of all colors of testing image *Parrot* in $\mathcal{L}'_{24}$ and their convex hull obtained with [95].

Let I be the color image to be processed and $I(i,j)$ be the color of its pixel (i,j). $I(i,j)$ is a vector in a form of $\{L^*(i,j), a^*(i,j), b^*(i,j)\}$, where $L^*(i,j)$, $a^*(i,j)$ and $b^*(i,j)$ denote, respectively, the luminance and chrominance values of pixel (i,j) in CIELAB format. All pixels of image I can be classified into 52 sets according to their luminance as follows.

$$\Lambda_k = \{I(i,j) | I(i,j) \in \mathcal{L}_k\} \quad \text{for } k=0:51 \tag{7.2}$$

Except set Λ_{51}, the maximum luminance difference among the pixels in each set is bounded to be 500/255≈1.96 (<2% of the full scale), which is negligible to a certain extent. The luminance difference among the pixels in Λ_{51} is 0.

One of the desired goals of VQ-based RCGC is to replace a pixel's color with a palette color whose index value is close to the luminance value of the pixel. It introduces an incentive to set up a constraint that the five palette colors in $\mathcal{L}_k$ are trained based on the colors of the pixels in Λ_k only because their luminance values are similar. This approach also provides a side benefit of reducing the complexity of palette generation as it turns the original palette generation task into 51 separate small-scale palette generation tasks.

Consider the case that we are going to train five palette colors for layer $\mathcal{L}_k$ based on the colors of the pixels in Λ_k for a specific *k*. By ignoring the luminance difference of the pixels in Λ_k, one can collapse the space of layer $\mathcal{L}_k$ into a two-dimensional space with mapping

$$f\big((L^*, a^*, b^*)\big) = (a^*, b^*) \tag{7.3}$$

where L^*, a^* and b^* are the three CIELAB components of a color in $\mathcal{L}_k$. This further simplifies the palette generation task by reducing the space dimension from 3 to 2. We note that the mapping result is a color with its luminance removed. The collapsed space of layer $\mathcal{L}_k$ is referred to as $\mathcal{L}'_k$ hereafter.

After the mapping, the color of each pixel in Λ_k can be represented as a point of coordinates (a^*, b^*) in $\mathcal{L}'_k$. As an example, Figure 7-1(b) shows the collapsed space of layer $\mathcal{L}_{24}$ (i.e. $\mathcal{L}'_{24}$) and Figure 7-1(c) shows the distribution of $f(I(i,j))$ in $\mathcal{L}'_{24}$ for all $I(i,j)\in\Lambda_k$ when I is the testing image *Parrot*. Based on the distribution of $f(I(i,j))$ in $\mathcal{L}'_{24}$, a convex hull can be obtained with the convex hull generation algorithm proposed in [95]. The convex hull forms an enclosure as shown in Figure 7-1(c). In theory, any color inside the enclosure can be rendered by mixing dots of the colors corresponding to the vertices of the convex hull [2]. Hence, if there is no limit on the palette size, we can make all colors corresponding to the vertices as the palette colors for layer $\mathcal{L}_{24}$ as halftoning allows us to render an image with dot density modulation [2].

In our approach, the number of palette colors in $\mathcal{L}'_k$ is limited to be 5, so a convex hull of 5 vertices is expected. It is rare to happen that the convex hull obtained at this stage has less than 5 vertices. However, when it happens, we cluster $f(I(i,j))$ in $\mathcal{L}'_k$ according to their distances to the vertices, pick the vertex associated with the cluster having the most number of $f(I(i,j))$ and split it into two vertices by introducing some small random amount of perturbation. The process repeats until we have a convex hull of 5 vertices.

When the convex hull has more than 5 vertices, its vertices are reduced with an iterative algorithm as follows. In each iteration cycle, we search along the boundary of the convex hull in $\mathcal{L}'_k$ to find out a pair of neighboring vertices based on a measure to be defined and then replace them with a new vertex. The new vertex is the intersection of the two straight lines extended from the sides each of which has one of the replaced vertices as its end point. The measure is referred to as selectability index and it is defined as $S_i = w^2 + h^2$, where w is the distance between the two vertices and h is the distance of the intersection point from the side connecting the two vertices.

We test all neighboring vertices and the pair having the lowest selectability index value are selected to merge as mentioned above. Figure 7-2 shows how the merging is carried out to reduce the vertices of a convex hull in an iteration cycle.

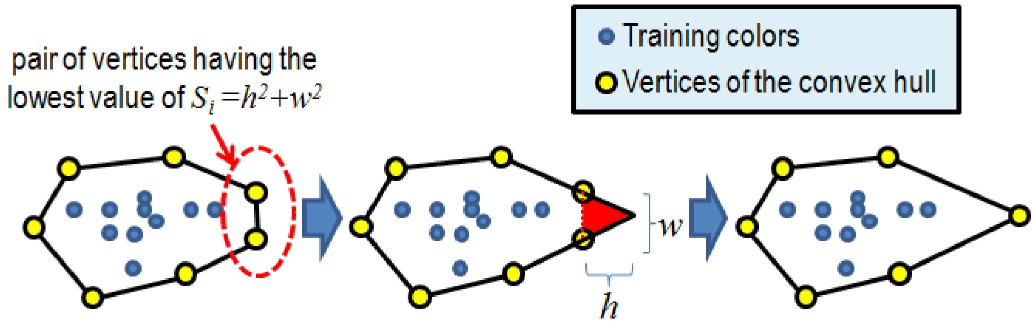


Figure 7-2 Reducing the vertices of a convex hull by merging 2 vertices to 1

The merging results in a larger convex hull that fully covers the original convex hull and hence it guarantees that all colors in the image are still inside the updated convex hull after the merging. The merging step repeats until a 5-point convex hull is obtained.

The vertices of the resultant 5-point convex hull can be selected to form five palette colors for layer $\mathcal{L}_k$ after assigning a luminance component $L_* = 500(k+0.5)/255$ to them. However, we found that the performance of the resultant palette is suboptimal. It is because this approach concerns too much on whether all colors in the image can be rendered by halftoning. In order to make the resultant large convex hull covers all colors in the image, its vertices are located far away from the center region of the convex hull. Accordingly, colors in the center region cannot be directly represented with the palette colors defined by the vertices. In the following section, we will show how to release the cover-all-colors constraint to improve the color palette.

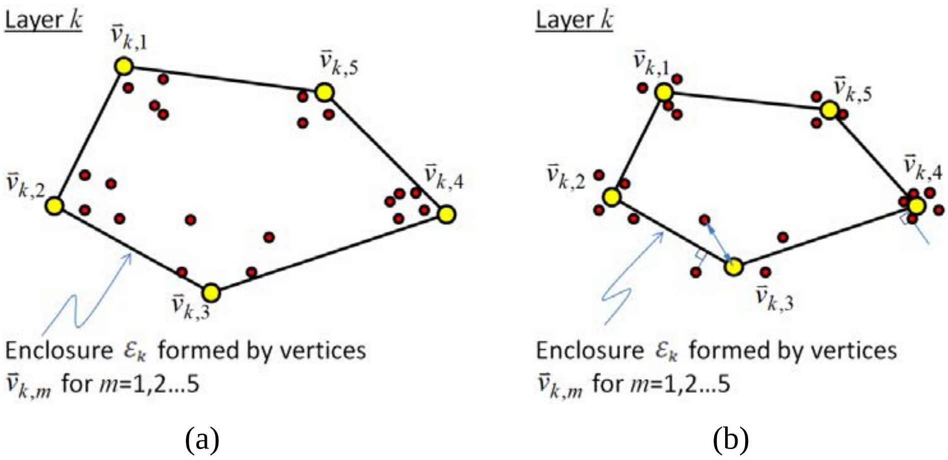


Figure 7-3 5-point convex hulls that have different properties: (a) covering all training colors and (b) not covering all training colors

7.3 Formulation of palette optimization

Let $\vec{v}_{k.m}$ for m=1,2…5 be the luminance-removed palette colors for layer $\mathcal{L}_k$ and $\mathcal{E}_k$ be the convex hull of $\{\vec{v}_{k.m}|m = 1,2 \ldots 5\}$ in $\mathcal{L}'_k$. For the sake of presentation, we refer to the members in Λ'_k={$f(I(i,j))| I(i,j)$∈Λ_k} as the training colors as they are used to train $\vec{v}_{k.m}$'s. Figures 7-3(a) and 7-3(b) show, respectively, the case when the convex hull of $\vec{v}_{k.m}$ for m=1,2…5 covers all training colors and the opposite case.

On one hand, we want to enclose all training colors in $\mathcal{E}_k$ as shown in Figure 7-3(a) such that they can be rendered with a halftoning technique in RCGC. On the other hand, we want to reduce the mean square error between the training colors and their closest palette colors such that, if possible, we can use palette colors to represent them directly. To achieve this goal, $\vec{v}_{k.m}$ should be the center of a cluster of training colors and then $\mathcal{E}_k$ does not enclose all training colors as shown in Figure 7-3(b). The two objectives are mutually contradictory. In our approach, we proposed two separate objective functions accordingly and then jointly optimize them to make a balance between these two factors.

When halftoning is applied, the color in a spatial region of an image is rendered by mixing pixels of the palette colors properly in the region. Though in theory any color inside $\mathcal{E}_k$ can be rendered correctly with a halftoning technique, the size of the spatial region casts a physical constraint on its realization. As an example, if there is a single pixel with outstanding color in a smooth region, it is impossible to mix pixels of different colors to render its color and we can only assign the closest palette color to that pixel. In general, the smaller the size of a homogeneous color region, the more difficult its color can be rendered with a halftoning technique.

In view of this factor, we introduce a measure to evaluate to what extent a pixel shares its color with its neighboring pixels in a local region of 5×5 pixels as follows.

$$H(i,j) = \frac{1}{25}\sum\nolimits_{|i'-i|,|j'-j|\le 2} P_{(i,j)}(i',j') \tag{7.4}$$

where

$$P_{(i,j)}(i',j') = \begin{cases} 1 & if\ \|I(i,j) - I(i',j')\| < \rho \\ 0 & else \end{cases} \tag{7.5}$$

In our realization, threshold parameter ρ is selected to be 2.3 based on the well accepted rule that two colors are visually indistinguishable when their difference in CIELAB domain is less than 2.3 [96].

$H(i,j) \in \{d/25 | d = 1,2 \ldots 25\}$ is bounded in (0,1]. A smaller value of $H(i,j)$ implies that pixel (*i*,*j*) shares its color with fewer pixels in its local region and hence it is more difficult to render its color with halftoning techniques. In that case, it is more desirable to have a palette color which is close to $I(i,j)$ such that we can reduce the visual color shift of pixel (*i*,*j*) even though halftoning does not work properly. On the contrary, a color shared in a large region can be easily rendered with a halftoning technique and hence it is less critical to make a palette color close to it. Since $H(i,j)$

reflects the effectiveness of using halftoning to render a specific color, it can be used to weight the contribution of individual training colors when defining the first objective function for optimizing the palette colors. In formulation, we have

$$J_{in} = \sum_{m=1}^{5} \sum_{\substack{f(I(i,j))\in\mathcal{E}_k \\ I(i,j)\in\Omega_{k,m}}} \frac{\|f(I(i,j)) - \vec{v}_{k.m}\|^2}{H(i,j)} \tag{7.6}$$

where $\Omega_{k,m}$={c|c∈$\mathcal{E}_k$ and $\|c - \vec{v}_{k.m}\|$≤$\|c - \vec{v}_{k.n}\|$ for n=1,2…5} for m=1,2…5.

Note that only the training colors inside $\mathcal{E}_k$ are taken into account in objective function J_{in}. It is because, from theoretical point of view, training colors outside $\mathcal{E}_k$ cannot be faithfully rendered by mixing palette colors. To take them into account, another objective function is introduced as follows.

$$J_{out} = \sum_{\substack{f(I(i,j))\notin\mathcal{E}_k \\ I(i,j)\in\Lambda_k}} \varkappa\big(f\big(I(i,j)\big), \mathcal{E}_k\big)^2 \tag{7.7}$$

where $\varkappa(f(I(i,j)), \mathcal{E}_k)$ denotes the shortest distance of $f\big(I(i,j)\big)$ from $\mathcal{E}_k$. $\varkappa(f(I(i,j)), \mathcal{E}_k)$ provides a measure of the visual color error between $I(i,j)$ and the closest color that can be rendered with $\vec{v}_{k.m}$for m=1,2,…5 in ideal situation.

The two objective functions try to minimize the visual color errors of the pixels inside and outside $\mathcal{E}_k$ respectively. A compromise can be made by minimizing a joint objective function defined as

$$J = J_{in} + \mu J_{out} \tag{7.8}$$

where μ is a weighting factor that controls the contribution of each objective. The optimal luminance-removed palette colors for layer $\mathcal{L}_k$ are the elements of the set $\{\vec{v}_{k.m}|m = 1,2 \dots 5\}$ that minimizes objective function *J*.

7.4 Realization of palette optimization

Objective function *J* is a non-linear function since classification is required to judge whether $f(I(i,j))$ is in $\mathcal{E}_k$ and whether $f(I(i,j))$ is in $\Omega_{k,m}$ for all *m*=1,2,…5. In our solution, simulated annealing is exploited to minimize *J*. The vertices of the 5-point convex hull obtained with the approach discussed in Section 7.2 are used as the initial estimates of $\vec{v}_{k.m}$ for *m*=1,2…5. Accordingly, we have J_{out}=0 in the initial state and Figure 7-3(a) shows the situation of such a state.

Simulated annealing is a double-loop iterative algorithm that simulates an annealing process [97]. In the outer loop, temperature *T* is gradually reduced with

$$T_{p+1} = \alpha T_p \qquad (7.9)$$

where T_p is the temperature in the p^{th} loop iteration cycle and α is a constant used to achieve cooling. For each temperature T_p, a fixed number of iterations are carried out to search for a better estimate of $\vec{v}_{k.m}$. Specifically, in each iteration cycle, for each *m*, a new estimate of $v_{k,m}$ is determined as

$$\vec{v}'_{k.m} = \vec{v}_{k.m} + \gamma_{k,m}\vec{r} \qquad (7.10)$$

where $\vec{r}$ is a random 2D perturbation vector whose magnitude is bound in (0,1] and

$$\gamma_{k,m} = \min\left\{\|\vec{v}_{k.m} - \vec{v}_{k.m+}\|, \|\vec{v}_{k.m} - \vec{v}_{k.m-}\|, \left\|\vec{v}_{k.m} - \tfrac{1}{5}\textstyle\sum_{n=1}^{5}\vec{v}_{k.n}\right\| \right\} \times 0.1 \qquad (7.11)$$

where $$\vec{v}_{k.m+} = \begin{cases} \vec{v}_{k.m+1} & if\ m < 5 \\ \vec{v}_{k.1} & if\ m = 5 \end{cases} \tag{7.12}$$

and $$\vec{v}_{k.m-} = \begin{cases} \vec{v}_{k.m-1} & if\ m > 1 \\ \vec{v}_{k.5} & if\ m = 1 \end{cases} \tag{7.13}$$

Once the new set of estimates is determined, its cost is evaluated with objective function (7.8). The new set of estimates will be accepted if the cost can be reduced. Otherwise the new set of estimates can only be accepted when the following test is passed.

$$rand(0,1) \le e^{\frac{-\Delta J}{K_B T_p}} \tag{7.14}$$

where $rand(0,1)$ is a real-time random value generator that outputs a uniformly distributed random number in the interval (0,1), ΔJ is the increment in cost and K_B=0.001. This arrangement allows some uphill moves so that it is possible to escape from a local minimum. The most recently accepted set of $\vec{v}_{k.m}$ estimates will be carried forward to the next iteration cycle.

Simulated annealing terminates when no new estimate is accepted for more than a pre-determined consecutive number of iterations. The best set of $\vec{v}_{k.m}$ estimates that achieve the lowest cost throughout the course are selected to be the five palette colors for layer $\mathcal{L}_k$. The vertices are sorted and re-indexed such that the direction angle of $\vec{v}_{k.m} - \sum_{n=1,2\dots 5} \vec{v}_{k.n}$ is larger for a larger value of index m.

We note that palette colors for different layers can be separately optimized in parallel. Once they are determined, the final 256-color palette is obtained as follows.

$$\begin{cases} \vec{p}_{5k+m} = \left(\frac{100(5k+m-1)}{255}, a(\vec{v}_{k.m}), b(\vec{v}_{k.m})\right) \quad \text{for} \quad 0 \le k \le 50, 1 \le m \le 5 \\ \vec{p}_{256} = (100,0,0) \end{cases} \tag{7.15}$$

where $\vec{p}_n$ is the n^{th} palette color of the 256-color palette, $a(\vec{v}_{k.m})$ and $b(\vec{v}_{k.m})$ denote, respectively, the a^* and b^* compoments of $\vec{v}_{k.m}$.

7.5 Working under our proposed framework

Once the optimal color palettes are acquired, color quantization and error diffusion can work together to realize RCGC. Let C be the color image to be processed. Figure 7-4 shows the operation flows of the RCGC process and the color image reconstruction process. Specifically, the RCGC method scans the color image C with serpentine scanning and processes it pixel by pixel. Assume that the current pixel being processed is (i,j). After extracting the luminance component of color $C'(i,j)$, which is denoted as $L'(i,j)$in Figure 7-4, $C'(i,j)$ is quantized to

$$\hat{C}(i,j) = \vec{p}_{5\kappa+\eta} \tag{7.16}$$

where

$$\kappa = \left\lfloor \frac{255}{500} \cdot L'(i,j) \right\rfloor \tag{7.17}$$

$$\eta = \begin{cases} \arg\min_{\mathrm{n}=1,2\ldots5} \|f(C'(i,j)) - f(\vec{p}_{5\kappa+n})\| & \text{if } \kappa < 51 \\ 1 & \text{if } \kappa = 51 \end{cases} \tag{7.18}$$

The quantization noise

$$E_c(i,j) = C'(i,j) - \hat{C}(i,j) \tag{7.19}$$

is then diffused to its neighbor pixels using Floyd's diffusion filter [39]. The diffusion is performed in channels L^*, a^* and b^* separately.

The index of the palette color assigned to replace $C'(i,j)$ is given by

$$I_{\hat{C}}(i,j) = 5\kappa + \eta \tag{7.20}$$

After the conversion, the index plane also serves as a 256-level grayscale version of the original image C. When necessary, the color palette can also be embedded into $I_{\hat{c}}(i,j)$ with the approach similar to that described in Section 6.5.

To recover the color image, the palette is first extracted from $I_{\hat{c}}$. Though a simple table-lookup process, a preliminary color image $\hat{C}$ can be obtained. The halftoning artifacts suppression process discussed in Section 6.6 can further be applied to $\hat{C}$ to remove the impulse noise introduced by halftoning. The final reconstructed color image $\tilde{C}$ is then obtained.

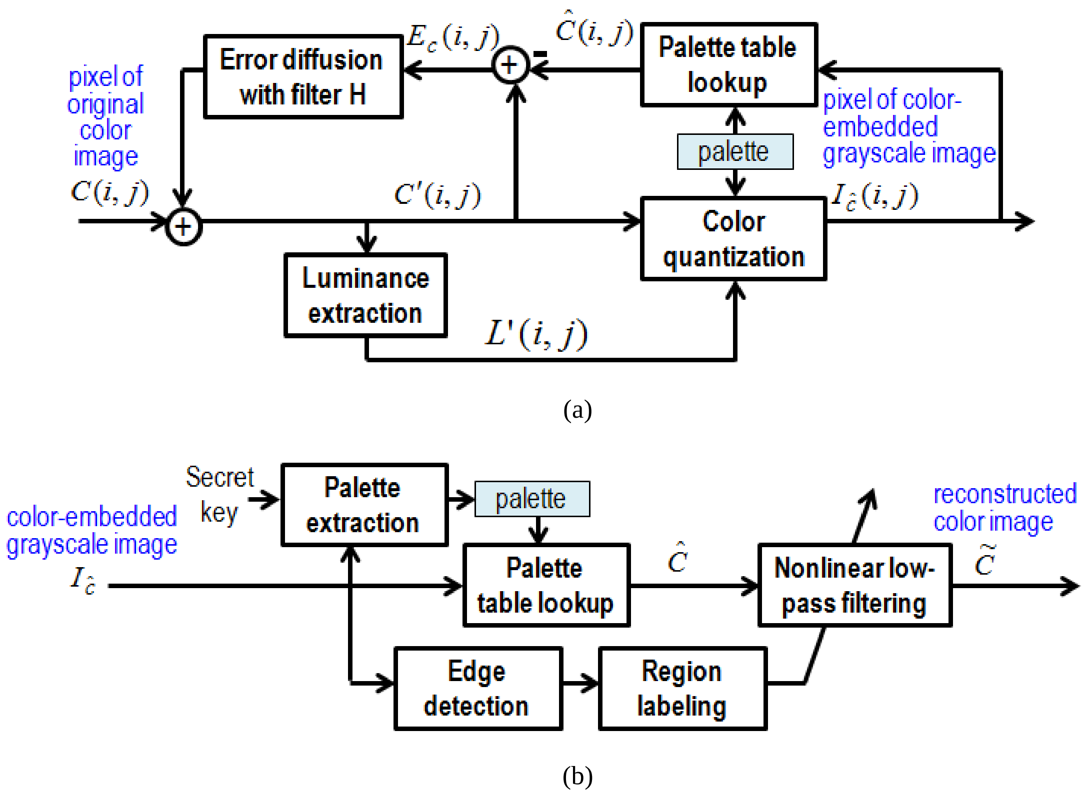


Figure 7-4 Operation flows of (a) color-to-grayscale conversion and (b) color image recovery

7.6 Performance evaluation

This Section provides some simulation results for evaluating the performance of the proposed RCGC algorithm when it works with the palettes produced with the palette generation algorithm proposed in this Chapter.

7.6.1 Parameters used in simulation

Some parameters are required for optimizing a palette with simulated annealing when realizing the color palette generation algorithm proposed in this Chapter. Their values were selected as follows in our simulation. As suggested in [97], the value of α in eqn. (7.9) is set to 0.9 and the number of iterations in each temperature is fixed to 50. The initial temperature T_0 is set to be $T_0 = -0.05J_0/K_b\log_e(0.1)$, where J_0 is the initial cost evaluated with eqn. (7.8). By setting so, the chance of accepting a 5% increase in cost J is 10% at the initial stage. Controlling parameter $\gamma_{k,t}$ is adaptively determined for each $\{k,t\}$ as in eqn. (7.11). The rule is set to keep the order of the vertices of a convex hull unchanged under a perturbation.

7.6.2 Simulation results

Some state-of-art RCGC algorithms including Xu and Chan's algorithm [38] (the one proposed in Chapter 6), Queiroz's [35], Ko et al.'s [34], Horiuchi et al.'s [36], Tanaka et al.'s [30] and Chaumont et al.'s [31] were also evaluated in our simulations for comparison. Two sets of test images were exploited. One is the Kodak set that includes 24 color images [71]. The other one includes 10 color images that are available in [98]. They are shown in Figure 7-5 for reference.

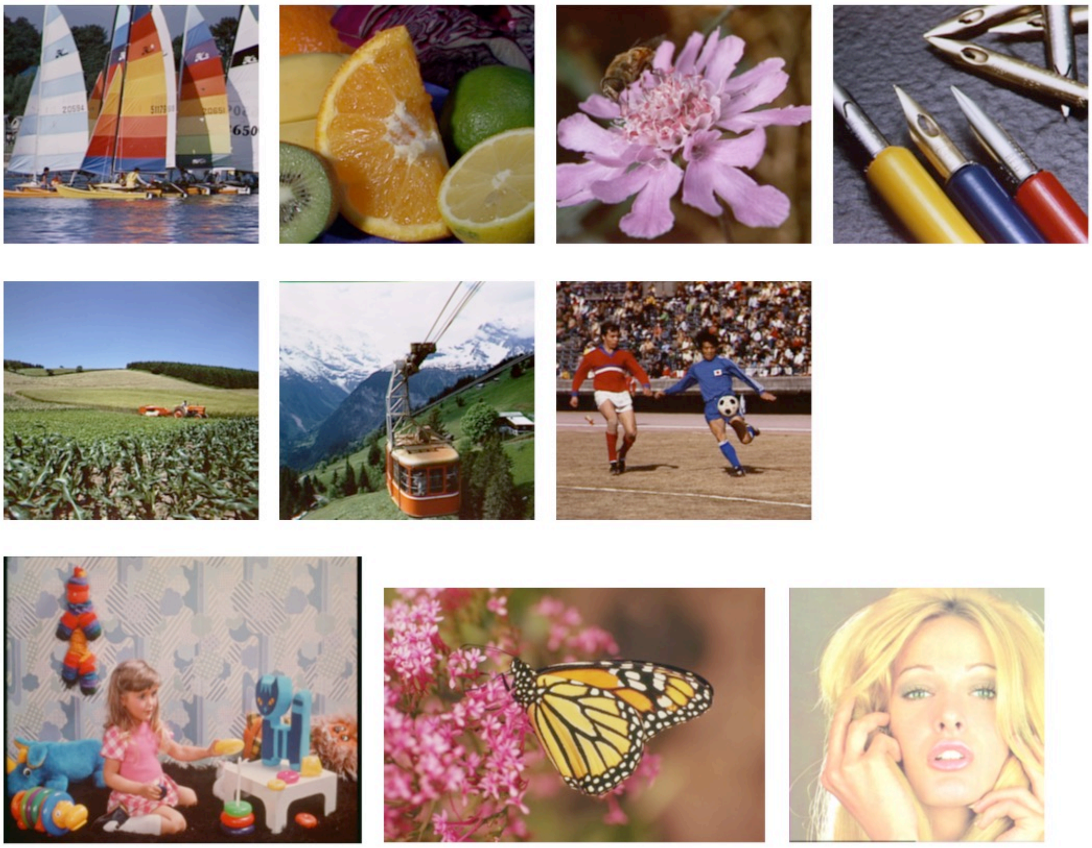


Figure 7-5 Second set of test images

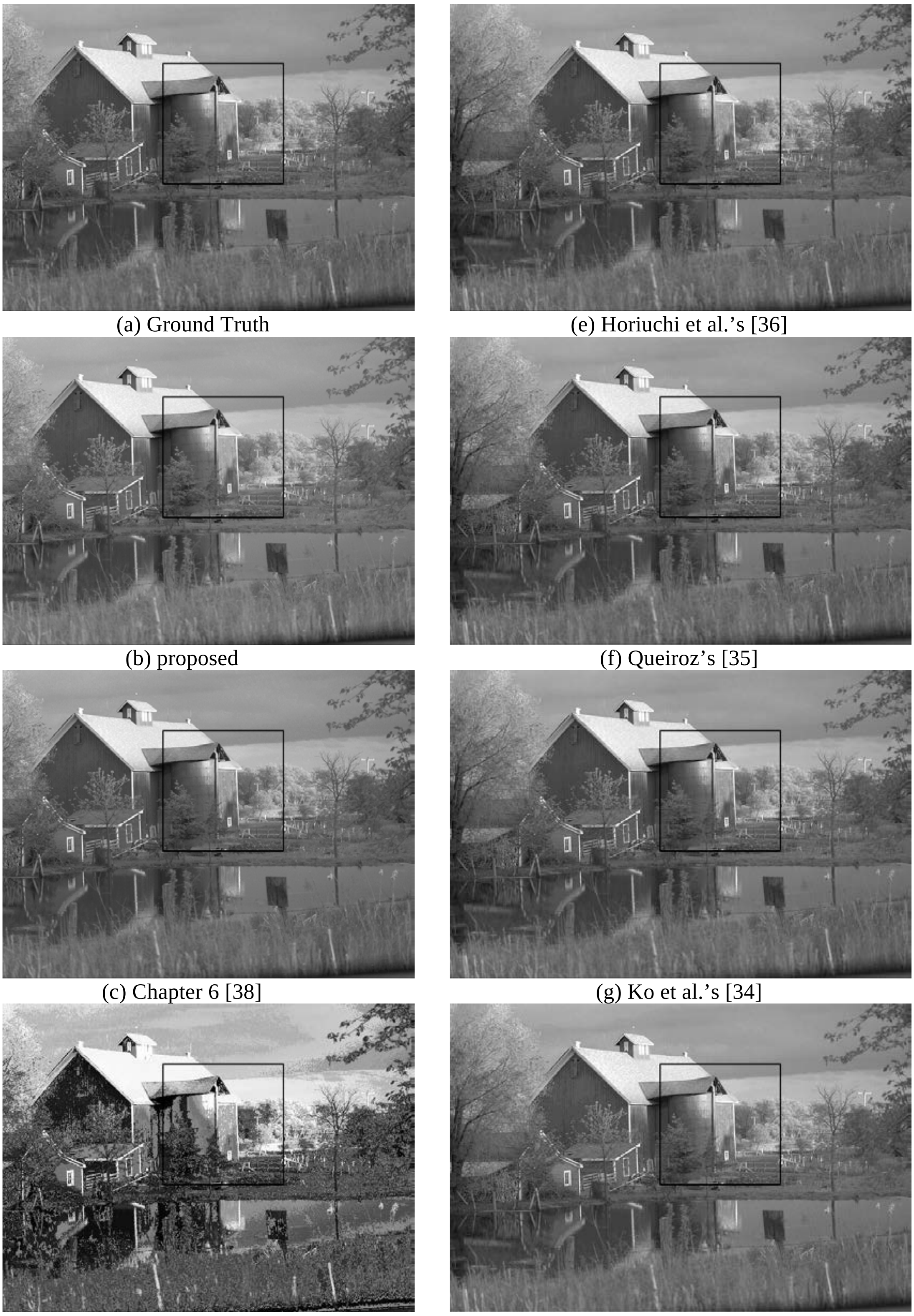
7.6.2.1 Subjective study

Figures 7-6 and 7-7 show, respectively, the color-embedded grayscale images and the recovered color images obtained with various RCGC algorithms when the test image is *House*. For better inspection, a marked region of each image shown in Figures 7-6 and 7-7 is enlarged and displayed in Figures 7-8 or 7-9.

The findings from the simulation results are similar to those obtained in Chapter 6, For examples, one can notice regular pattern noise in the smooth regions of the color-embedded grayscale images produced with SE-based RCGC algorithms (See Figures 7-8(e), (f) and (g)) and luminance distortion in the color-embedded grayscale images produced with traditional VQ-based RCGC algorithms (See Figures 7-8(d) and (h)). In contrast, Figures 7-8(b) and (c) are extremely similar to the original.

In Figures 7-9(d) and (h), we can observe that the recovered color images of Tanaka et al.'s [30] and Chaumont et al.'s [31] have some false contours (see the red wall under sunshine). There is also severe color shift in Figure 7-9(h). For the color images recovered with SE-based algorithms, one can find the artifacts caused by downsampling the chrominance planes and removing the high frequency content of the luminance plane. For example, there are aliasing artifacts along the sharp straight edges (e.g. the pipe) in Figure 7-9(e). Figures 7-9(f) and (g) are somehow blurred (e.g. the tree leaves) as compared with Figure 7-9(a) due to the loss of high frequency content caused by downsampling. Comparatively speaking, Figures 7-9(b) and (c) can preserve more texture.

To allow an easier visual comparison between the algorithms proposed in Chapter 6 [38] and Chapter 7, we provide the recovered color images of another test image in Figure 7-10. Figure 7-11 shows some enlarged regions of Figure 7-10. As shown in Figure 7-11(b) and (c), the impulse noise in Figure 7-11(b) is much less than that in Figure 7-11(c). Unlike the palette developed in Chapter 6, the palette colors of the palette developed in Chapter 7 form a smaller enclosure in the color space. In consequence, a color inside the enclosure can be rendered with closer palette colors when halftoning is exploited. The hue fluctuation introduced by the halftoning process is smaller in a smooth region. With the help of the halftoning artifacts suppression process discussed in Chapter 6.6, the impulse noise can be significantly removed.



(a) Ground Truth

(e) Horiuchi et al.'s [36]

(b) proposed

(f) Queiroz's [35]

(c) Chapter 6 [38]

(g) Ko et al.'s [34]

(d) Chaumont et al. [31]

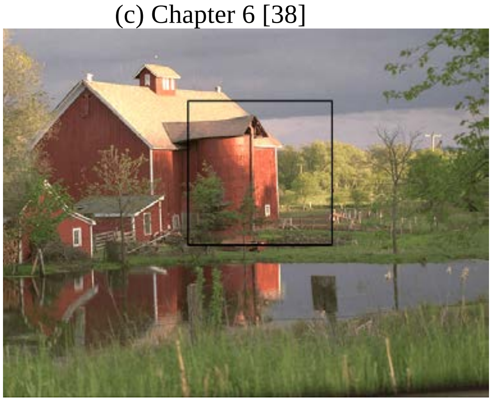
(h) Tanaka et al.'s [30]

Figure 7-6 Color embedded grayscale images obtained with various algorithms. (Test image is *House*.)

(a) Ground Truth

(b) proposed

(c) Chapter 6 [38]

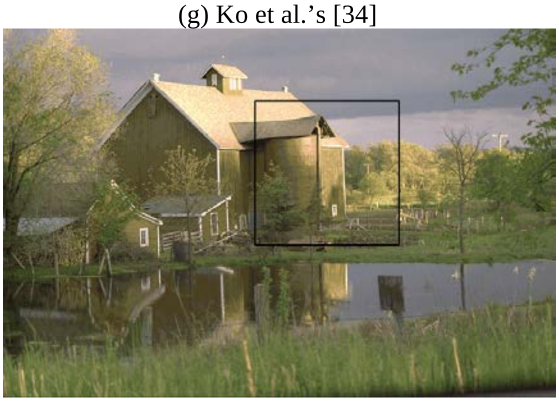


(d) Chaumont et al.[31]

(e) Horiuchi et al.'s [36]

(f) Queiroz's [35]

(g) Ko et al.'s [34]

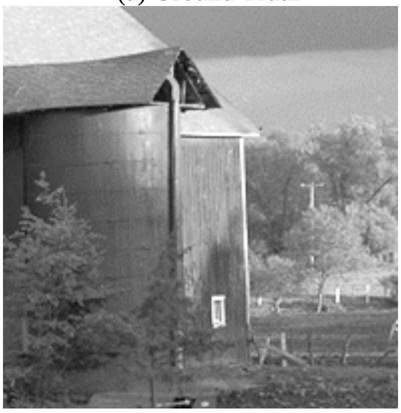


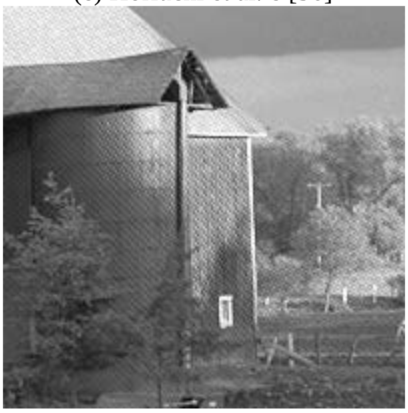
(h) Tanaka et al.'s [30]

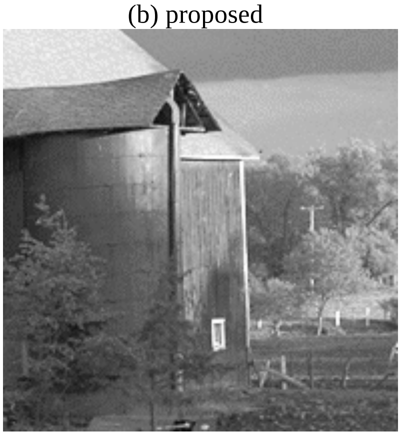
Figure 7-7 Recovered color images obtained with various algorithms (Test image is *House*.)

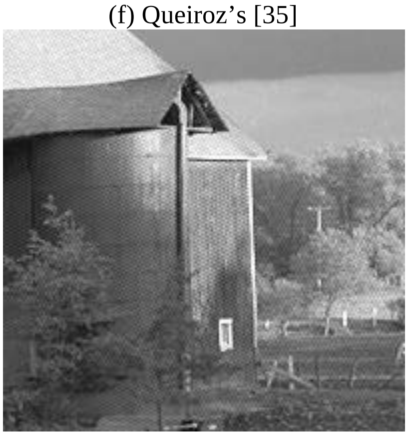

(a) Ground Truth

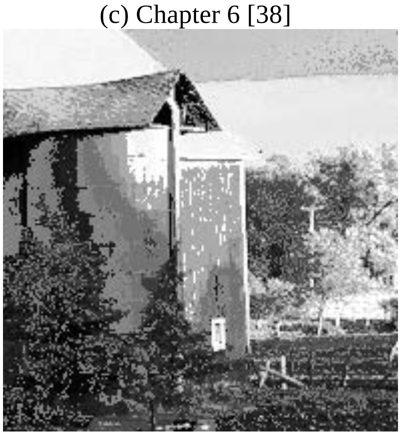

(e) Horiuchi et al.'s [36]


(b) proposed


(f) Queiroz's [35]


(c) Chapter 6 [38]


(g) Ko et al.'s [34]


(d) Chaumont et al. [31]

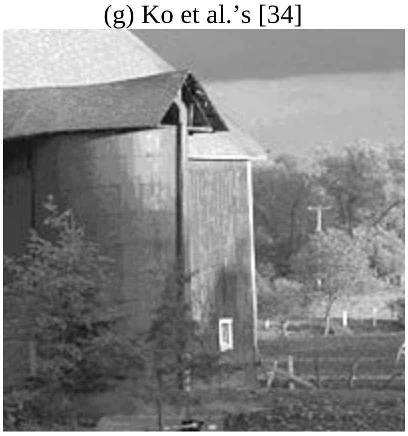

(h) Tanaka et al.'s [30]

Figure 7-8 Enlarged portions of Figure 7-6

(a) Ground Truth

(b) proposed

(c) Chapter 6 [38]

(d) Chaumont et al.[31]

(e) Horiuchi et al.'s [36]

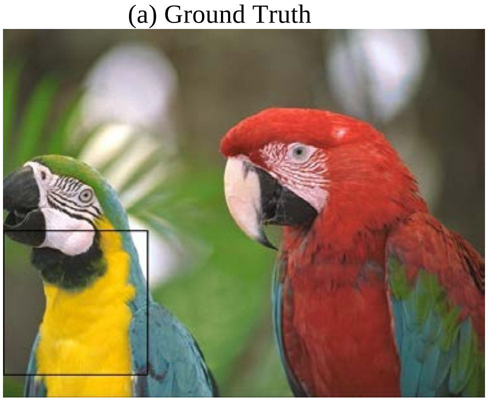
(f) Queiroz's [35]

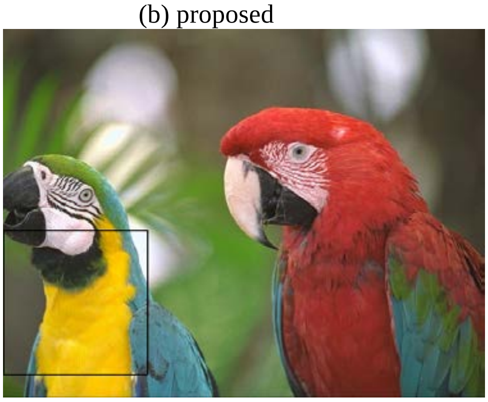
(g) Ko et al.'s [34]

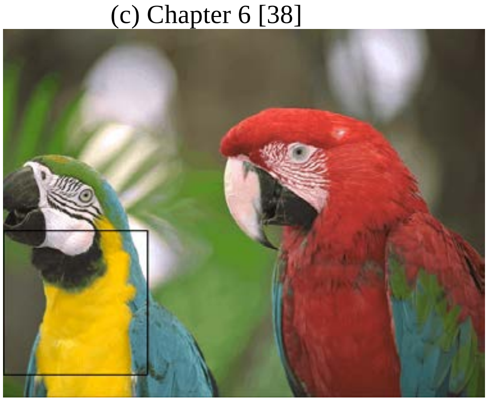
(h) Tanaka et al.'s [30]

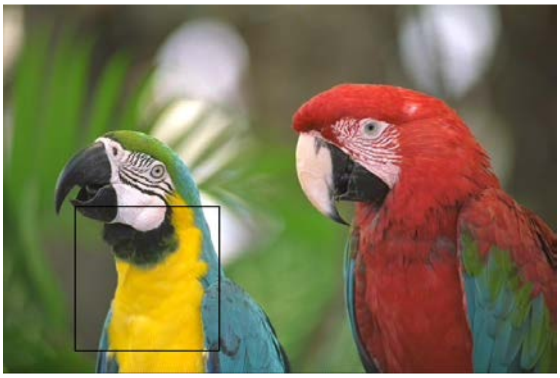
Figure 7-9 Enlarged portions of Figure 7-7

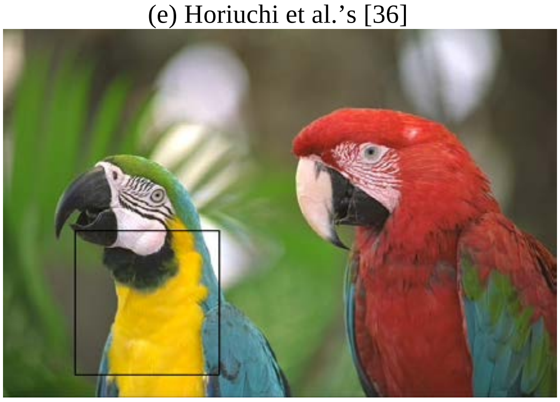

(a) Ground Truth

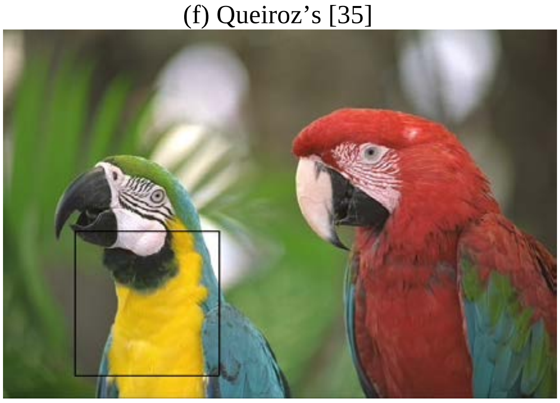

(b) proposed


(c) Chapter 6 [38]


(d) Chaumont et al.[31]


(e) Horiuchi et al.'s [36]


(f) Queiroz's [35]


(g) Ko et al.'s [34]

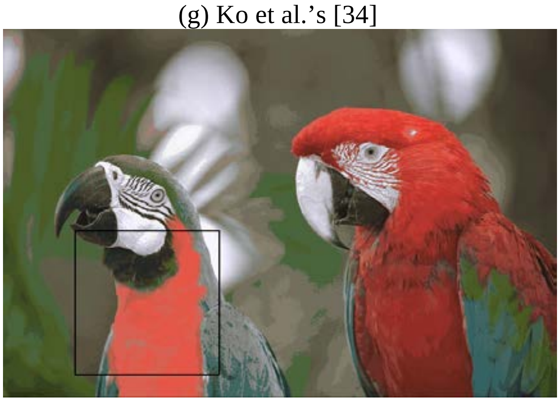

(h) Tanaka et al.'s [30]

Figure 7-10 Recovered color images obtained with various algorithms (Test image is *Parrot*.)

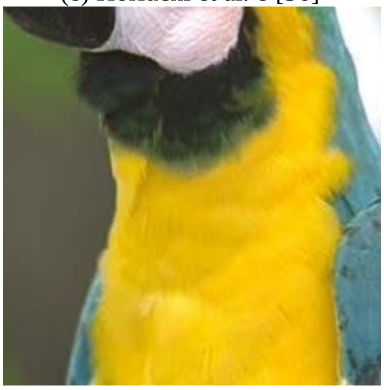
(a) Ground Truth

(b) proposed

(c) Chapter 6 [38]

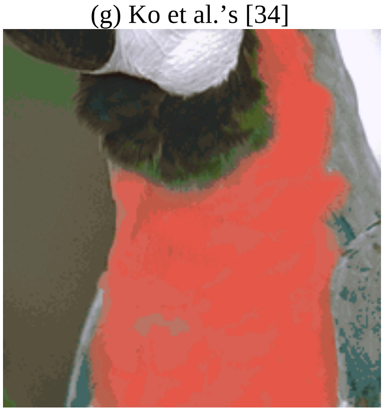
(d) Chaumont et al.[31]

(e) Horiuchi et al.'s [36]



(f) Queiroz's [35]

(g) Ko et al.'s [34]



(h) Tanaka et al.'s [30]

Figure 7-11 Enlarged portions of Figure 7-10

7.6.2.2 Objective study

The performance of various RCGC algorithms was also evaluated in terms of various objective measures. Tables 7-1 and 7-2 show, respectively, the evaluation results obtained based on the Kodak image set [71] and the test image set shown in Figure 7-5 [98]. While the other evaluated RCGC algorithms ([30], [31], [34]–[36] and [38]) consider the Y channel in YUV space as the luminance channel, the RCGC algorithm proposed in Chapter 7 considers the L^* channel in CIELAB space as the luminance channel and uses it as the target to produce color-embedded grayscale image. For a fair comparison, when computing the luminance-oriented metrics of the algorithm proposed in Chapter 7, the computations are based the normalized L^* values while the computations for other algorithms are based on the normalized Y values.

Tables 7-1, 7-2 and 7-3 show the performance of various RCGC algorithms. For each metric, the scores of the best two algorithms are bolded while the score of the best one is underscored in the Tables for easier comparison. The algorithm proposed in Chapter 7 provides the best performance in terms of a lot of metrics. In general, both its produced color-embedded grayscale images and recovered color images have leading-good quality. As compared with the algorithm proposed in Chapter 6 [38], the average *PSNR* improvements for the Kodak image set , 'Miscellaneous' image set [88] and the additional test image set are, respectively, 1.29 dB, 5.88 dB and 1.24 dB. As for C*PSNR*, the average improvements for the three image sets are 4.14 dB, 1.85 dB and 2.46 dB respectively.

d	Color-embedded Grayscale Image				Recovered Color Image							
	PSNR (dB)	SSIM	dpi=100	dpi=150	CPSNR (dB)	ΔE	dpi = 100			dpi = 150		
			HVS-PSNR (dB)	HVS-PSNR (dB)			HVS-PSNR$_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$	HVS-PSNR$_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$
z's	31.64	0.976	40.92	43.85	31.84	2.80	36.78	0.877	1.75	38.09	0.889	1.63
l.'s	32.82	0.989	42.88	46.67	32.13	4.46	36.05	0.838	3.43	37.01	0.848	3.27
i et 6]	33.20	0.997	39.19	43.66	29.53	3.12	33.69	0.927	1.49	36.02	0.952	1.21
et 0]	28.07	0.986	29.79	31.19	33.07	3.45	34.78	0.874	2.55	35.51	0.879	2.44
nt et 1]	16.72	0.643	17.92	18.47	37.79	1.90	39.94	0.962	1.02	40.99	0.968	0.92
d 38]	35.87	0.988	41.54	44.01	32.71	2.97	38.04	0.963	0.79	41.24	0.968	0.58
ed	37.16	0.991	43.53	46.85	36.85	1.78	42.10	0.987	0.52	44.97	0.990	0.38

Table 7-1 Average performance of RCGC algorithms for Kodak image set [71]

d	Color-embedded Grayscale Image				Recovered Color Image							
	PSNR (dB)	SSIM	dpi=100	dpi=150	CPSNR (dB)	ΔE	dpi = 100			dpi = 150		
			HVS-PSNR (dB)	HVS-PSNR (dB)			HVS-PSNR$_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$	HVS-PSNR$_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$
z's	29.89	0.962	39.59	43.12	28.68	8.52	30.32	0.716	3.11	31.22	0.747	2.85
.'s	30.26	0.980	41.18	46.06	29.88	7.69	31.42	0.823	2.51	32.18	0.856	2.19
et 6]	35.95	0.999	40.87	43.24	29.06	7.83	30.97	0.841	2.32	31.99	0.877	1.96
et 0]	31.31	0.974	32.75	33.56	33.24	5.17	34.85	0.874	1.92	37.22	0.886	1.78
nt et 1]	16.03	0.627	16.96	17.35	34.41	5.19	36.85	0.919	1.48	38.32	0.929	1.34
d 38]	35.95	0.986	41.33	43.51	31.47	5.43	36.45	0.956	0.97	39.52	0.964	0.72
ed	41.83	0.997	49.21	54.32	33.32	3.54	36.71	0.945	0.96	39.61	0.964	0.64

Table 7-2 Performance of RCGC algorithms for *'Miscellaneous'* image set [88]

d	Color-embedded Grayscale Image				Recovered Color Image							
	PSNR (dB)	SSIM	dpi=100	dpi=150	CPSNR (dB)	ΔE	dpi = 100			dpi = 150		
			HVS-PSNR (dB)	HVS-PSNR (dB)			HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$	HVS-$PSNR_{Color}$ (dB)	CSSIM	$\Delta E_{S\text{-}CIELAB}$
z's	30.73	0.961	40.49	47.75	30.33	5.40	32.72	0.766	2.82	33.66	0.799	2.51
.'s	30.94	0.999	42.11	49.63	31.69	4.70	34.00	0.850	2.15	34.88	0.887	1.81
i et 6]	35.21	0.999	41.08	45.45	29.75	4.98	32.62	0.843	2.18	34.04	0.884	1.78
et 0]	33.35	0.985	34.74	35.40	30.59	4.55	33.59	0.821	2.71	34.81	0.835	2.52
nt et 1]	18.64	0.688	19.82	20.39	34.50	3.24	36.61	0.904	1.60	37.77	0.921	1.42
d 38]	35.77	0.989	41.59	44.13	31.33	4.15	36.44	0.942	1.08	39.41	0.958	0.81
ed	37.01	0.992	43.35	44.66	33.79	3.57	38.26	0.966	0.88	40.76	0.979	0.64

Table 7-3 Average performance of RCGC algorithms for Additional image set [98]

We examine the computation time for our palette optimization and the traditional fuzzy c-mean clustering method [72] in a notebook computer with Intel i7 CPU and Matlab environment. Note the program is not optimized. The image size is 768×512. The average time for 256 color palette generation is 51.277 seconds for the proposed algorithm, while it is 96.2 seconds for fuzzy c-mean clustering method [72]. Considering that our optimization can be operated in parallel in each segment, our algorithm is capable to achieve faster speed than traditional clustering method.

7.7 Summary

In Chapter 6, palettes are optimized under an assumption that any colors in an image can be rendered with a halftoning technique as long as they are enclosed by the palette colors in the RGB color space. As a matter of fact, this assumption is only valid in ideal situations. In this Chapter, we take the practical constraint of halftoning into account and redefine a new objective function to optimize a color palette. A new VQ-based RCGC algorithm is then proposed under the framework proposed in Chapter 6. Simulation results show that the new RCGC algorithm can further improve the quality of both the color-embedded grayscale image and the recovered color image remarkably in terms of various objective and subjective metrics.

Chapter 8.

Conclusions

8.1 Summary of this work

As a powerful noise shaping technique, digital halftoning [1, 2] can be used in various digital signal processing applications to reduce the impact of quantization noise on the quality of a quantized signal. Though there have already been tremendous studies on its applications in printing, studies on its applications in some other areas are relatively limited. This work is dedicated to studying its applications in 3D profilometry and reversible color to grayscale conversion.

Digital fringe projection technique has been widely used in 3D profilometry and binary defocusing method is one of the most effective and popular methods to realize it. By projecting defocused binary fringe patterns, a measurement system can support real-time measurements and solve the luminance nonlinearity problem of a projector. In Chapter 3, we proposed a framework to measure objects by projecting octa-level fringe patterns. It directly improves the measurement accuracy by increasing the intensity levels of a fringe pattern at no extra cost as compared with a system that projects binary fringe patterns. Three patch-based octa-level fringe pattern generation algorithms are proposed to support the framework. While the one introduced in Chapter 3 is a basic version, the two proposed in Chapters 4 and 5 were developed to address the harmonic distortion issue that comes with the use of patched-based fringe patterns. In particular, the former one gets rid of the harmonics along the direction of sinusoidal variation in a fringe pattern and the latter one aims at removing the harmonics along any directions. We also proactively solve some common weaknesses of conventional patch-based binary fringe pattern generation algorithms (e.g. [6], [10], [18] and [23]). In consequence, the measurement performance achieved with the resultant fringe patterns is robust to the extent of defocusing and there is no length limitation of a fringe period. By exploiting a FMED-based halftoning technique to get initial estimates and improving the strategy to search a better estimate during the optimization of patches, optimized patches can seamlessly and randomly tiled to produce octa-level fringe patterns of ideal noise characteristics, which eventually improve the measurement accuracy significantly.

Conventional reversible color-to-grayscale conversion algorithms [27–32, 34–37] generally concern more about the quality of recovered color images and hence their produced color-embedded grayscale images are usually visually distorted and suspicious. In Chapter 6, we proposed a VQ-based RCGC framework in which halftoning is exploited to shift the quantization noise of both color-embedded grayscale images and recovered color images to their high frequency bands. Consequently, it significantly improves their visual quality simultaneously. To support the operation of the framework, we proposed two color palette generation algorithms in Chapters 6 and 7 respectively. The former one takes the halftoning effect in an ideal situation into account while the latter one further takes the practical constraints of a halftoning process into account when optimizing a color palette. Under the proposed RCGC framework, two RCGC algorithms were developed. Each of them adopts one of the palettes developed with the two proposed palette generation algorithms. Both RCGC algorithms can effectively preserve the spatial features of the original image and eliminate visual artifacts such as pattern noise, false contour and blurring in their output images. By taking the practical constraint of a halftoning process into account, the algorithm proposed in Chapter 7 can remove the impulse noise introduced by the halftoning process in a better way. Simulation results show that the performancne of both algorithms are better than conventional RCGC algorithms in terms of various objective measures.

In short, this work presents several our contributions made to improve the accuracy of 3D shape measurement and the performance of reversible color-to-grayscale conversion. The proposed techniques can accomplish significant improvement over many of the existing techniques.

8.2 Future works

Further extension of the present work is possible in several directions. Traditional fringe projection profilometry methods project fringe patterns on the object to be measured, capture the distorted fringe patterns that are determined by the distance of the surface of the object from the camera, and then construct a phase map based on the captured fringe patterns. A phase unwrapping process is required to turn the reconstructed phase map to the depth map of the object. There can be serious errors in phase unwrapping when the object has an abruptly changing height profile. Recently, proposals [99–101] have been made to encode markers in the projected fringe patterns to resolve the ambiguities in the fringe images in phase unwrapping. It is interesting to explore the feasibility of encoding markers in binary and octa-level fringe patterns and investigate the best encoding method for resolving the ambiguities effectively.

Under the proposed VQ-based reversible color-to-grayscale conversion framework, the quality of color-embedded grayscale images and recovered color images highly relies on the exploited color palette. In the proposals suggested in Chapters 6 and 7, the number of palette colors is fixed and there is a one-to-one correspondence between the index value and the palette color. It anchors the color of a pixel in the recovered color image and the luminance value of the corresponding pixel in the color-embedded grayscale image. This constraint makes it impossible to assign accurate grayscales and colors to the color-embedded grayscale image and the recovered color images respectively at the same time. A possible way to alleviate this constraint is to re-index palette colors adaptively in the course of conversion. By doing so, the correspondence between index value and palette color is local. It should be helpful to further improve the quality of the images simultaneously.

Digital halftoning can definitely be applied to some other possible application areas and it is worthwhile to explore each one of them. As a matter of fact, we also carried out two preliminary studies in this project to apply halftoning to block truncation coding [102] and liquid crystal display blacklight dimming [103]. In the first study, without either decreasing the compression ratio or increasing the realization complexity, we successfully improve the visual quality of an encoded image by removing the blocking artifacts. In the second study, we effectively suppress the false contours caused by the re-quantization process that is carried out to compensate for the backlight dimming. The results are encouraging and they suggest two other possible directions to carry out further studies.

Bibliography

[1] R. Ulichney, *Digital halftoning*. MIT Press, 1987.

[2] D. L. Lau and G. R. Arce, *Modern digital halftoning*, 2rd ed. CRC Press, 2008.

[3] F. Chen, G. M. Brown, and M. Song, “Overview of three-dimensional shape measurement using optical methods,” *Opt. Eng.*, vol. 39, no. 1, pp. 10–22, Jan. 2000.

[4] X.-Y. Su and W.-S. Zhou, “Automated phase-measuring profilometry using defocused projection of a Ronchi grating,” *Opt. Commun.*, vol. 94, no. 6, pp. 561–573, Dec. 1992.

[5] Y. Wang and S. Zhang, “Comparison of the squared binary, sinusoidal pulse width modulation, and optimal pulse width modulation methods for three-dimensional shape measurement with projector defocusing,” *Appl. Opt.*, vol. 51, no. 7, p. 861, Mar. 2012.

[6] C. Zuo, Q. Chen, G. Gu, J. Ren, X. Sui, and Y. Zhang, “Optimized three-step phase-shifting profilometry using the third harmonic injection,” *Opt. Appl.*, vol. 43, no. 2, 2013.

[7] C. Zuo *et al.*, “High-speed three-dimensional shape measurement for dynamic scenes using bi-frequency tripolar pulse-width-modulation fringe projection,” *Opt. Lasers Eng.*, vol. 51, no. 8, pp. 953–960, Aug. 2013.

[8] C. Zuo, Q. Chen, S. Feng, F. Feng, G. Gu, and X. Sui, “Optimized pulse width modulation pattern strategy for three-dimensional profilometry with projector defocusing,” *Appl. Opt.*, vol. 51, no. 19, p. 4477, Jul. 2012.

[9] Y. Xu, L. Ekstrand, J. Dai, and S. Zhang, "Phase error compensation for three-dimensional shape measurement with projector defocusing," *Appl. Opt.*, vol. 50, no. 17, p. 2572, Jun. 2011.

[10] J. Dai, B. Li, and S. Zhang, "High-quality fringe pattern generation using binary pattern optimization through symmetry and periodicity," *Opt. Lasers Eng.*, vol. 52, pp. 195–200, 2014.

[11] J. Dai, B. Li, and S. Zhang, "Intensity-optimized dithering technique for three-dimensional shape measurement with projector defocusing," *Opt. Lasers Eng.*, vol. 53, pp. 79–85, Feb. 2014.

[12] J. Dai and S. Zhang, "Phase-optimized dithering technique for high-quality 3D shape measurement," *Opt. Lasers Eng.*, vol. 51, no. 6, pp. 790–795, 2013.

[13] Y. Wang and S. Zhang, "Three-dimensional shape measurement with binary dithered patterns," *Appl. Opt.*, vol. 51, no. 27, p. 6631, Sep. 2012.

[14] G. A. Ayubi and J. A. Ferrari, "Optimal pulse-width modulation for sinusoidal fringe generation with projector defocusing: comment," *Opt. Lett.*, vol. 36, no. 6, p. 808, Mar. 2011.

[15] Y. Wang and S. Zhang, "Optimal pulse width modulation for sinusoidal fringe generation with projector defocusing," *Opt. Lett.*, vol. 35, no. 24, p. 4121, Dec. 2010.

[16] S. Lei and S. Zhang, "Flexible 3-D shape measurement using projector defocusing," *Opt. Lett.*, vol. 34, no. 20, p. 3080, Oct. 2009.

[17] G. a Ayubi, J. a Ayubi, J. M. Di Martino, and J. a Ferrari, "Pulse-width modulation in defocused three-dimensional fringe projection," *Opt. Lett.*, vol. 35, no. 21, p. 3682, Nov. 2010.

[18] X. Li, Z. Zhang, and C. Yang, "High-quality fringe pattern generation using binary pattern optimization based on a novel objective function," *Opt. - Int. J. Light Electron Opt.*, vol. 127, no. 13, pp. 5322–5327, 2016.

[19] Y. Zhang, S. Member, Z. Xiong, Z. Yang, and S. Member, "Real-Time Scalable Depth Sensing With Hybrid Structured Light Illumination," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 97–109, 2014.

[20] Y. Wang, K. Liu, Q. Hao, D. L. Lau, and L. G. Hassebrook, "Period coded phase shifting strategy for real-time 3-D structured light illumination," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3001–3013, 2011.

[21] W. Lohry and S. Zhang, "Genetic method to optimize binary dithering technique for high-quality fringe generation," *Opt. Lett.*, vol. 38, no. 4, p. 540, 2013.

[22] P. Cong, Z. Xiong, Y. Zhang, S. Zhao, and F. Wu, "Accurate Dynamic 3D Sensing With Fourier-Assisted Phase Shifting," *IEEE J. Sel. Top. Signal Process.*, vol. 9, no. 3, pp. 396–408, 2015.

[23] J. Sun, C. Zuo, S. Feng, S. Yu, Y. Zhang, and Q. Chen, "Improved intensity-optimized dithering technique for 3D shape measurement," *Opt. Lasers Eng.*, vol. 66, pp. 158–164, Mar. 2015.

[24] M. Arevalillo-Herraez, F. R. Villatoro, and M. A. Gdeisat, "A Robust and Simple Measure for Quality-Guided 2D Phase Unwrapping Algorithms," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2601–2609, Jun. 2016.

[25] Z.-X. Xu and Y.-H. Chan, "Optimized multilevel fringe patterns for real-time 3D shape measurement with defocused projector," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 2730–2734.

[26] Z.-X. Xu and Y.-H. Chan, "Removing harmonic distortion of measurements of a defocusing three-step phase-shifting digital fringe projection system," *Opt. Lasers Eng.*, vol. 90, pp. 139–145, 2017.

[27] M. Chaumont and W. Puech, "A fast and efficient method to protect color images," in *SPIE 6058, Visual Communications and Image Processing*, 2007, p. 65081T.

[28] M. Chaumont and W. Puech, "A Grey-Level Image Embedding its Color Palette," in *2007 IEEE International Conference on Image Processing*, 2007, vol. 1, p. I-389-I-392.

[29] M. Chaumont and W. Puech, "Protecting the Color Information by Hiding it," in *Recent Advances in Signal Processing*, InTech, 2009.

[30] G. Tanaka, N. Suetake, and E. Uchino, "Invertible color-to-monochrome transformation based on clustering with lightness constraint," in *IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 2151–2154.

[31] M. Chaumont, W. Puech, and C. Lahanier, "Securing color information of an image by concealing the color palette," *J. Syst. Softw.*, vol. 86, no. 3, pp. 809–825, 2013.

[32] R. L. de Queiroz and K. M. Braun, "Color to gray and back: color embedding into textured gray images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1464–1470, Jun. 2006.

[33] M. A. Ruzon, S. Cohen, G. Wilensky, and M. Newell, "Fast and adaptive color-to-grayscale conversion," in *ACM SIGGRAPH 2006 Sketches on - SIGGRAPH '06*, 2006, p. 124.

[34] K.-W. Ko, O.-S. Kwon, C.-H. Son, and Y.-H. Ha, “Color Embedding and Recovery Based on Wavelet Packet Transform,” *J. Imaging Sci. Technol.*, vol. 52, no. 1, p. 10501, 2008.

[35] R. L. de Queiroz, “Reversible color-to-gray mapping using subband domain texturization,” *Pattern Recognit. Lett.*, vol. 31, no. 4, pp. 269–276, 2010.

[36] T. Horiuchi, F. Nohara, and S. Tominaga, “Accurate reversible color-to-gray mapping algorithm without distortion conditions,” *Pattern Recognit. Lett.*, vol. 31, no. 15, pp. 2405–2414, 2010.

[37] K.-W. Ko, D.-C. Kim, W.-J. Kyung, and Y.-H. Ha, “Color Embedding and Recovery Using Wavelet Packet Transform with Pseudorandomized Saturation Code,” *J. Imaging Sci. Technol.*, vol. 55, no. 3, p. 30501, 2011.

[38] Z.-X. Xu and Y.-H. Chan, “Improving reversible color-to-grayscale conversion with halftoning,” *Signal Process. Image Commun.*, vol. 52, pp. 111–123, 2017.

[39] R. W. Floyd and L. Steinberg, “Adaptive algorithm for spatial greyscale,” vol. 17, no. 2. pp. 75–77, 1976.

[40] J. F. Jarvis, C. N. Judice, and W. H. Ninke, “A survey of techniques for the display of continuous tone pictures on bilevel displays,” *Comput. Graph. Image Process.*, vol. 5, no. 1, pp. 13–40, Mar. 1976.

[41] P. Stucki, *MECCA – a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction*. Zurich: IBM Research Laboratory, 1981.

[42] B. E. Bayer, “An optimum method for two-level rendition of continuous-tone pictures,” in *IEEE Ineternational Conference on Communication*, 1976, pp. 69–77.

[43] D. E. Knuth and D. E., “Digital halftones by dot diffusion,” *ACM Trans. Graph.*, vol. 6, no. 4, pp. 245–273, Oct. 1987.

[44] J. Allebach and Q. Lin, “FM screen design using DBS algorithm,” in *3rd IEEE International Conference on Image Processing*, 1996, vol. 1, pp. 549–552.

[45] I. Katsavounidis and C.-C. Jay Kuo, “A multiscale error diffusion technique for digital halftoning,” *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 483–490, Mar. 1997.

[46] Y.-H. Chan, “A modified multiscale error diffusion technique for digital halftoning,” *IEEE Signal Process. Lett.*, vol. 5, no. 11, pp. 277–280, Nov. 1998.

[47] Y.-H. Chan, “Feature-preserving multiscale error diffusion for digital halftoning,” *J. Electron. Imaging*, vol. 13, no. 3, p. 639, Jul. 2004.

[48] Y.-H. Fung and Y.-H. Chan, “Optimizing the Error Diffusion Filter for Blue Noise Halftoning With Multiscale Error Diffusion,” *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 413–417, Jan. 2013.

[49] Y.-H. Fung and Y.-H. Chan, “A multiscale error diffusion algorithm for green noise digital halftoning,” in *17th Signal Processing Conference*, 2009.

[50] B. W. Kolpatzik and C. A. Bouman, “Optimized error diffusion for image display,” *J. Electron. Imaging*, vol. 1, no. 3, p. 277, Jul. 1992.

[51] P.-W. Wong, “Adaptive error diffusion and its application in multiresolution rendering,” *IEEE Trans. Image Process.*, vol. 5, no. 7, pp. 1184–1196, Jul. 1996.

[52] X. Li, “Edge-Directed Error Diffusion Halftoning,” *IEEE Signal Process. Lett.*, vol. 13, no. 11, pp. 688–690, Nov. 2006.

[53] P. Li and J. P. Allebach, “Tone-Dependent Error Diffusion,” *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 201–215, Feb. 2004.

[54] Y.-H. Fung and Y.-H. Chan, “Tone-dependent noise model for high-quality halftones,” *J. Electron. Imaging*, vol. 22, no. 2, p. 23004, Apr. 2013.

[55] Y.-H. Fung and Y.-H. Chan, “Tone-dependent error diffusion based on an updated blue-noise model,” *J. Electron. Imaging*, vol. 25, no. 1, p. 13013, Jan. 2016.

[56] N. Damera-Venkata and B. L. Evans, “Design and analysis of vector color error diffusion halftoning systems,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1552–1565, 2001.

[57] D. L. Lau, G. R. Arce, and N. C. Gallagher, “Digital color halftoning with generalized error diffusion and multichannel green-noise masks,” *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 923–935, May 2000.

[58] N. Damera-Venkata, B. L. Evans, and V. Monga, “Color error-diffusion halftoning what differentiates it from grayscale error diffusion?,” *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 51–58, Jul. 2003.

[59] M. T. Orchard and C. A. Bouman, “Color quantization of images,” *IEEE Trans. Signal Process.*, vol. 39, no. 12, pp. 2677–2690, 1991.

[60] Y.-H. Fung and Y.-H. Chan, “A Technique for Producing Scalable Color-Quantized Images With Error Diffusion,” *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3218–3224, Oct. 2006.

[61] Y.-H. Fung and Y.-H. Chan, “Edge preserving multiscale error diffusion algorithm for green noise digital halftoning,” in *2010 IEEE International Conference on Image Processing*, 2010, pp. 557–560.

[62] M. Takeda and K. Mutoh, “Fourier transform profilometry for the automatic measurement of 3-D object shapes,” *Appl. Opt.*, vol. 22, no. 24, p. 3977, Dec. 1983.

[63] J.-F. Lin and X. Su, “Two-dimensional Fourier transform profilometry for the automatic measurement of three-dimensional object shapes,” *Opt. Eng.*, vol. 34, no. 11, pp. 3297–3302, Nov. 1995.

[64] E. B. Li, X. Peng, J. Xi, J. F. Chicharo, J. Q. Yao, and D. W. Zhang, “Multi-frequency and multiple phase-shift sinusoidal fringe projection for 3D profilometry,” *Opt. Express*, vol. 13, no. 5, pp. 1561–1569, 2005.

[65] D. C. Ghiglia and M. D. Pritt, *Two-dimensional phase unwrapping : theory, algorithms, and software*. Wiley, 1998.

[66] S. Zhang, “Active versus passive projector nonlinear gamma compensation method for high-quality fringe pattern generation,” in *Dimensional Optical Metrology and Inspection for Practical Applications*, 2014, p. 911002.

[67] W. Zhang, X. Hu, X. Li, and N. Yu, “Recursive Histogram Modification: Establishing Equivalency Between Reversible Data Hiding and Lossless Data Compression,” *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2775–2785, Jul. 2013.

[68] B. Ou, X. Li, Y. Zhao, R. Ni, and Y.-Q. Shi, “Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding,” *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5010–5021, Dec. 2013.

[69] I.-C. Dragoi and D. Coltuc, “Adaptive Pairing Reversible Watermarking,” *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2420–2422, May 2016.

[70] W. Wharton and D. Howorth, *Principles of television reception*. Pitman, 1971.

[71] “Kodak true color image suite.” [Online]. Available: http://r0k.us/graphics/kodak/.

[72] J. C. Bezdek, R. Ehrlich, and W. Full, “FCM: The fuzzy c-means clustering algorithm,” *Comput. Geosci.*, vol. 10, no. 2–3, pp. 191–203, Jan. 1984.

[73] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Appl. Stat.*, vol. 28, no. 1, p. 100, 1979.

[74] D. Ozdenir and L. Akarun, "Fuzzy error diffusion," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 683–690, Apr. 2000.

[75] D. Ozdemir and L. Akarun, "Fuzzy algorithms for combined quantization and dithering," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 923–931, Jun. 2001.

[76] H.-Z. Huang, K. Xu, R. R. Martin, F.-Y. Huang, and S.-M. Hu, "Efficient, Edge-Aware, Combined Color Quantization and Dithering," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1152–1162, Mar. 2016.

[77] D. Caspi, N. Kiryati, and J. Shamir, "Range imaging with adaptive color structured light," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 470–480, May 1998.

[78] F. A. Baqai and J. P. Allebach, "Printer models and the direct binary search algorithm," in *1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 1998, vol. 5, pp. 2949–2952.

[79] N. Karpinsky, M. Hoke, V. Chen, and S. Zhang, "High-resolution, real-time three-dimensional shape measurement on graphics processing unit," *Opt. Eng.*, vol. 53, no. 2, p. 24105, Feb. 2014.

[80] B. Li, Y. Wang, J. Dai, W. Lohry, and S. Zhang, "Some recent advances on superfast 3D shape measurement with digital binary defocusing techniques," *Opt. Lasers Eng.*, vol. 54, pp. 236–246, 2014.

[81] D. Malacara, *Optical Shop Testing, 3rd Edition*, 3rd ed. New York: Wiley, 2007.

[82] P. Heckbert, “Color image quantization for frame buffer display,” in *9th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’82*, 1982, vol. 16, no. 3, pp. 297–307.

[83] M. Gervautz and W. Purgathofer, “A simple method for color quantization: octree quantization,” in *Graphics gems*, Academic Press Professional, Inc., 1990, pp. 287–293.

[84] J. Puzicha, M. Held, J. Ketterer, J. M. Buhmann, and D. W. Fellner, “On spatial quantization of color images,” *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 666–682, Apr. 2000.

[85] N. Nikolaidis and I. Pitas, “Robust image watermarking in the spatial domain,” *Signal Processing*, vol. 66, no. 3, pp. 385–403, 1998.

[86] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 1998, pp. 839–846.

[87] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[88] “The USC-SIPI image database.” [Online]. Available: http://sipi.usc.edu/database/.

[89] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[90] J.-M. Guo and Y.-F. Liu, “Joint Compression/Watermarking Scheme Using Majority-Parity Guidance and Halftoning-Based Block Truncation Coding,” *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2056–2069, Aug. 2010.

[91] M. Hassan and C. Bhagvati, “Structural Similarity Measure for Color Images,” *Int. J. Comput. Appl.*, vol. 43, no. 14, pp. 7–12, Apr. 2012.

[92] X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, “Color image quality metric S-CIELAB and its application on halftone texture visibility,” in *IEEE COMPCON 97*, 1997, pp. 44–48.

[93] X. Zhang and B. Wandell, “A spatial extension of CIELAB for digital color-image reproduction,” *J. Soc. Inf. Disp.*, vol. 5, no. 1, p. 61, 1998.

[94] J. Schanda, “CIE Colorimetry,” in *Colorimetry*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007, pp. 25–78.

[95] S. G. Akl and G. T. Toussaint, “A fast convex hull algorithm,” *Inf. Process. Lett.*, vol. 7, no. 5, pp. 219–222, Aug. 1978.

[96] G. Sharma, *Digital color imaging handbook*. CRC Press, 2003.

[97] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by Simulated Annealing,” *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[98] “Additional Test Images.” [Online]. Available: http://www.hlevkin.com/default.html#testimages.

[99] S. Gai and F. Da, “A novel phase-shifting method based on strip marker,” *Opt. Lasers Eng.*, vol. 48, no. 2, pp. 205–211, Feb. 2010.

[100] H. Cui, W. Liao, N. Dai, and X. Cheng, “A flexible phase-shifting method with absolute phase marker retrieval,” *Measurement*, vol. 45, no. 1, pp. 101–108, Jan. 2012.

[101] B. Budianto, P. K. D. Lun, and T.-C. Hsung, “Marker encoded fringe projection profilometry for efficient 3D model acquisition,” *Appl. Opt.*, vol. 53, no. 31, p. 7442, Nov. 2014.

[102] Z.-X. Xu and Y.-H. Chan, "Eliminating blocking artifacts in halftoning-based block truncation coding," in *European Signal Processing Conference*, 2016, p. 913 - 917.

[103] Z.-X. Xu and Y.-H. Chan, "A constrained error diffusion method to shape quantization noise in backlight dimming," in *19th International Conference on Digital Signal Processing*, 2014, pp. 909–913.