國立交通大學

電機與控制工程學系

博士論文

影像感測器之色彩濾波陣列補插與
輪式機器人之視覺追蹤控制設計

Color Filter Array Interpolation for an Image Sensor and Visual Tracking Control Design of a Wheeled Mobile Robot

研 究 生：蔡奇謚

指導教授：宋開泰 博士

中 華 民 國 九十七 年 七 月

影像感測器之色彩濾波陣列補插與輪式機器人之視覺追蹤控制設計

Color Filter Array Interpolation for an Image Sensor and Visual Tracking Control Design of a Wheeled Mobile Robot

研究生：蔡奇諡　　Student: Chi-Yi Tsai

指導教授：宋開泰 博士　　Advisor: Dr. Kai-Tai Song

國立交通大學

電機與控制工程學系

博士論文

A Dissertation

Submitted to Department of Electrical and Control Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electrical and Control Engineering

July 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年七月

影像感測器之色彩濾波陣列補插與
輪式機器人之視覺追蹤控制設計

學生:蔡奇謚　　　　　　　　　　指導教授:宋開泰　博士

國立交通大學電機與控制工程學系

摘要

本論文的目標為發展一新穎的色彩濾波陣列(Color Filter Array, CFA)補插演算法用以解決單一影像感測器之色彩重現問題，以及一具有強健性的視覺追蹤控制系統用以解決輪式移動型機器人之以視覺為基礎的移動控制問題。目前大部分的數位相機均採用一貝爾模式(Bayer Pattern)之 CFA 覆蓋於單一影像感測器上，用以擷取一貝爾模式之馬賽克影像(Bayer Mosaic Image)。一般所見的全彩影像則是將擷取到的貝爾模式之馬賽克影像，經由一色彩重現處理後，重建出所有像素上的色彩資訊。此色彩重現處理一般稱為 CFA 補插或 CFA 解馬賽克。為了重建出具有高逼真度的全彩影像，本論文提出一新穎的異次投影硬性決定(Heterogeneity-Projection Hard-Decision, HPHD)演算法並結合一新的以色差為基礎之邊緣適應性(Color-Difference Based Edge-Adaptive, CDEA)CFA 補插方法來解決貝爾模式之馬賽克影像的色彩重現問題。在此所提出的 HPHD 演算法主要目的為估測出最佳的補插方向並且執行硬性決定捕差法，此方法的主要特點在於補插的方向決定於執行補插動作之前，如此可提高色彩補插的效率。另一方面，所提出的 CDEA CFA 補插方法致力於將綠色色彩平面之高頻資訊加入其他色彩平面中，用以重現出較少色彩遺物的色彩值。

在視覺追蹤控制的設計中，本論文提出一應用於配有一傾斜式相機之輪式移動型機器人的強健型視覺追蹤控制系統，此系統包含一視覺追蹤控制器(Visual Tracking Controller, VTC)及一視覺狀態估測器(Visual State Estimator, VSE）。首先，一新穎的雙重Jacobian視覺互動模型被推導出來，用以幫助VTC及VSE的設計。VSE主要目的為在影像平面中直接估測出最佳系統狀態以及目標物之移動，VTC則利用VSE所估測出的結果，接著計算出機器人的控制速度。為了掌握實際視覺追蹤系統中所遭受到的不確定性，VSE可以克服由影像雜訊及短暫遮蔽不確定性所造成之干擾。另一方面，VTC不但對於系統參數之不確定性具有某種程度的強健性，並且也能克服速度命令中無法模型化的量化影響。因此，結合所提出的VTC及所提出的VSE，此視覺追蹤控制系統對於影像雜訊、系統參數、速度量化及短暫遮蔽所造成之不確定性影響皆具有強健性。電腦模擬及實驗結果驗證所提出的視覺追蹤控制系統之效果，包含追蹤性能、系統收斂性及系統強健性。

Color Filter Array Interpolation for an Image Sensor and Visual Tracking Control Design of a Wheeled Mobile Robot

Student: Chi-Yi Tsai Advisor: Dr. Kai-Tai Song

Department of Electrical and Control Engineering

National Chiao Tung University

ABSTRACT

The objective of this thesis is to develop a novel color filter array (CFA) interpolation algorithm for color reproduction of a single image sensor and a robust visual tracking control system for vision-based motion control of a wheeled mobile robot. Most digital cameras employ a single image sensor covered with a Bayer CFA to capture a Bayer mosaic image. A full-color image is then reconstructed from the captured Bayer mosaic image through a color reproduction process, commonly known as *CFA interpolation* or *CFA demosaicing*. To reconstruct the full-color images with high fidelity, a novel heterogeneity-projection hard-decision (HPHD) algorithm combined with a new color-difference based edge-adaptive (CDEA) CFA interpolation method is proposed for color reproduction of Bayer mosaic images. The proposed HPHD algorithm aims to estimate the optimal interpolation direction and perform hard-decision interpolation, in which the direction of interpolation is decided before performing the interpolation. On the other hand, the proposed CDEA CFA interpolation method devotes to reproduce color values with fewer color artifacts by adding the high-frequency information of green channel to other color channels. Compared with three recently reported CFA interpolation techniques, the proposed HPHD-CDEA method outperforms all of them in both quantitative and visual comparisons by utilizing twenty-five natural images from Kodak PhotoCD.

In the design of visual tracking control, a robust visual tracking control system, which consists of a visual tracking controller (VTC) and a visual state estimator (VSE), is proposed for a wheeled mobile robot equipped with a tilt camera. A novel dual-Jacobian visual interaction model is first derived to help the design of VTC and VSE. The VSE aims to estimate the optimal system state and target motion in the image plane directly, and the VTC then calculates the robot's control velocities by using the estimation results from VSE. To handle the uncertainties encountered in practical visual tracking control system, the VSE can overcome the disturbances caused by both image noise and temporary occlusion uncertainties. On the other hand, the VTC not only possesses some degree of robustness against the system model uncertainties, but also overcomes the unmodelled quantization effect in the velocity commands. Therefore, by combining the proposed VTC with the proposed VSE, the visual tracking control system is robust to the uncertainties of image noise, system model, velocity quantization and temporary occlusion. Computer simulations and experimental results validate the effectiveness of the proposed visual tracking control system, in terms of tracking performance, system convergence, and robustness.

誌謝

博士班的訓練是一條辛苦又漫長的研究之路。衷心感謝我的指導教授宋開泰博士，感謝他多年來在專業上及論文寫作上的指導，真摯誠意且不厭其煩的給我意見及修正，使我受益良多，也讓本論文得以順利完成。

接著，必須感謝論文口試委員－傅立成教授、王文俊教授、李迪章教授、陳建祥教授、王聖智教授、楊谷洋教授，對於本論文的建議與指引，強化本論文的嚴整性與可讀性。再來，要感謝交通大學電機與控制工程研究所的教授們－李祖添教授、吳炳飛教授、林昇甫教授、林清安教授、以及林心宇教授對於學業上的教導以及觀念上的啟發。

感謝 ISCI 實驗室的學弟們富聖、振揚、俊瑋、濬尉及志昇所提供的協助，也感謝學長戴任詔博士和博士班學弟嘉豪及孟儒對本論文的建議與討論，以及感謝已畢業的學弟們松峙、維祥、鎮源、崇民及柏秋的相互鼓勵及在生活上帶來的樂趣。

另外，特別感謝我的父母，由於他們辛苦栽培，在生活上給予我細心地關愛與照料，使得我才得以順利完成此論文；也感謝我女友在我的身邊全力的支持我，在我最無助及失意的時候給予我意見及鼓勵。

最後，感謝全能的上帝，將一切榮耀歸給上帝。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

An intelligent robot uses its on-board sensors to collect information from the surroundings and react to the changes of its immediate environment. In recent years, vision systems become one of the major on-board sensors for autonomous robots. Most vision systems utilize a monocular digital camera to capture full-color images of the environment for several purposes. A full-color image usually is composed of three color planes and required three separate image sensors to measure each color plane. In order to reduce the cost, many cameras employ a single image sensor covered with a *color filter array* (CFA), which consists of a set of spectrally selective filters that are arranged in an interleaved pattern so that each sensor pixel samples one of three primary color components, to capture the mosaic images. To render a full-color image from a mosaic image, a color reproduction process, commonly known as *CFA interpolation* or *CFA demosaicing*, is required to estimate for each pixel its two missing color values. Therefore, the researches on CFA interpolation for color reproduction have become an important topic in digital camera pipeline process [1-11].

The rendered full-color images of the environment can be used in several robotic applications, for example, recognition, navigation, localization, tracking control, etc. In recent years, computer vision techniques act a key role in robotic systems for making robot motion control and object tracking efficiently. Thus the study of *visual tracking control* (which means the vision-based robot motion control to track a target of interest) has gained increasing attention in recent years [12-36]. Based on the motion constraints of the robot, the research on visual tracking control can be classified into visual servoing for holonomic manipulators and visual tracking for nonholonomic mobile robots. Although visual servoing of holonomic manipulators has been discussed extensively and many results can be found in the literature [12-14], mobile robots are commonly nonholonomic and the visual servoing results for holonomic manipulators are unsuitable for the mobile platform [15]. Hence, the researches on mobile robot visual tracking control have been an active area in robotic researches [15-36]. According to the reasons above, this thesis aims to investigate the methodologies of CFA interpolation for a single image sensor and visual tracking control for a wheeled mobile robot.

1.2 Literature Survey

1.2.1 CFA Interpolation for a Single Image Sensor

Digital color images from single-chip digital cameras are obtained by interpolating the output from a CFA. The simplest CFA interpolation methodologies apply well-known image interpolation techniques, such as nearest-neighbor replication, bilinear interpolation, and cubic spline interpolation, to each color channel separately. However, these single-channel algorithms usually introduce severe color artifacts and blurs around sharp edges [1]. These drawbacks motivate the need of more advanced algorithms for improving demosaicing performance. An excellent review on advanced CFA interpolation algorithms can be found in [2].

In recent years, there have been investigations on more sophisticated CFA interpolation algorithms. In [3], Lu and Tan presented an improved hybrid CFA interpolation method that consists of two successive steps: an interpolation step to render full-color images and a post-processing step to suppress visible demosaicing artifacts. Muresan and Parks proposed an improved edge-directed CFA interpolation algorithm based on optimal recovery interpolation of grayscale images [4]. They first utilized a grayscale image interpolation algorithm based on optimal recovery estimation theory to interpolate the green plane. The red/blue channels were interpolated using inter-channel color difference adaptive filtering. These two CFA interpolation algorithms in general produce high quality visual results, especially in reconstructing sharp or well-defined edges of the image. However, in fine details or texture regions, where edges tend to be short and in different directions, these algorithms introduce undesirable errors and give degraded performance.

Meanwhile, two iterative CFA interpolation techniques were proposed by Gunturk *et al.* [5] and Li [6], respectively. In [5], a projection-onto-convex-set (POCS) technique was presented to estimate the missing color values in red and blue channels using alternating projection scheme based on high inter-channel correlation. In [6], Li formulated the CFA interpolation as a problem of reconstructing correlated signals from decimated versions and proposed a successive approximation strategy by adopting color difference interpolation iteratively. Although these iterative CFA interpolation algorithms perform well in texture regions and reveal low computational complexity, they cannot produce satisfactorily high quality visual results in well-defined edges of the image.

Another recent CFA interpolation approach divides the demosaicing procedure into interpolation stage and decision stage [7-10]. In the interpolation stage, horizontally and vertically interpolated images are produced respectively. In the latter decision stage, a soft-decision method, in which the interpolation must be performed before the decision procedure, was employed for choosing the pixels interpolated in the direction with fewer artifacts. Because the decision stage is essential for these CFA interpolation approaches, we refer them as *decision-based CFA interpolation algorithm*. For the decision stage, Hirakawa *et al.* proposed a homogeneity metric to measure the misguidance level of color artifacts presented in interpolated images [7]. Based on this measurement, the interpolation decision is made by choosing the region with larger homogeneity values. In [8], Wu *et al.* adopted the Fisher's linear discriminant technique to determine the optimal interpolation direction in a local window. In [9], Grossmann and Eldar utilized the YIQ color space as a tool to select the reconstructed regions with a smoother chrominance component. Recently, Omer and Werman proposed an enhanced decision-based CFA interpolation algorithm that combines the decision process with the standard CFA interpolation algorithm such as edge-directed scheme [11] to improve its performance in places the standard algorithm tends to fail [10]. The decision-based CFA interpolation algorithm performs well not only in texture regions, but also in well-defined edges of the image. However, the main drawback of these CFA interpolation algorithms is that they are not efficient in the interpolation stage because each pixel needs to be interpolated at least twice, one in horizontal direction and the other in vertical direction, for the next soft-decision procedure. This drawback also greatly increases the computing efforts in the latter decision stage. Therefore, it is still a challenge in CFA interpolation design to develop an efficient CFA interpolation method with high performance in both texture and edge regions.

1.2.2 Visual Tracking Control for a Wheeled Mobile Robot

The visual tracking control problem addressed in this thesis focuses on the visual tracking control of a unicycle-modeled (usually termed as wheeled) mobile robot equipped with an *on-board* monocular vision system. Due to the high number of different mobile robot visual tracking control methods, we classify the reported methods into four groups based on the type of the target to be tracked. Many efforts focus on the first group which aims to track a static target, such as a ground line, landmark, or reference image, for the purpose of mobile robot navigation or regulation (so-called homing) [15-28]. To track the ground line, Ma *et al.* formulated the visual tracking control problem as controlling the shape of a ground curve in the image plane and proposed a closed-loop vision-guided control system for a nonholonomic mobile robot [16]. Coulaud *et al.* proposed a simple and stable feedback controller design, which avoids sophisticated image processing and control algorithms, for a mobile robot equipped with a fixed camera to track a line on the ground [17]. In the case of tracking the landmark, the reported controllers usually modify the visual servoing technique to satisfy the nonholonomic constraint for the motion control of the mobile robot [18-21]. In [22], Zhang and Ostrowski utilized an optimal control method to solve the visual motion-planning problem by generating a virtual trajectory in the image plane and the corresponding optimal control signals for the robot to follow. Nierobisch *et al.* proposed a visual tracking control method for a mobile robot with a pan-tilt camera to track visual reference landmarks in the acquired views during autonomous navigation [23]. Recently, the homography-based [24, 25] and epipolar-based [15, 26-28] visual tracking control approaches were proposed for a mobile robot equipped with a pinhole or an omni-directional (so-called central catadioptric) camera to track a reference image toward a desired configuration. These two approaches consider the mobile robot visual tracking control problem as a visual servoing regulation or visual homing problem. In [24], Chen *et al*. developed a visual tracking controller based on the Euclidean homography to track a desired time-varying trajectory defined by a prerecorded image sequence of a stationary target viewed by the on-board camera as the mobile robot moves. However, the stability of their result is restricted by the non-zero reference velocity condition of the desired trajectory. To overcome this drawback, Fang *et al.* exploited Lyapunov-based techniques to construct a homography-based visual servoing regulation controller for proving asymptotic regulation of the mobile robot [25]. In [26], Mariottini *et al.* exploited the epipolar geometry defined by the current and desired camera views to develop a two-step visual servoing regulation controller. They also extend this design to the visual servoing regulation control of a mobile robot with a central catadioptric camera [27]. In [28], Goedemé *et al.* developed a vision-only navigation and homing system for mobile robots with an omni-directional camera. Their method divides the visual homing operation into two phases and computes visual homing vector based on epipolar geometry estimation. Although these approaches of the first group provide appropriate solutions for static target visual tracking control problem, they cannot guarantee to solve the moving (non-static) target visual tracking control problem.

The second group aims to track other robot teammates in a robot group for the formation control purpose [29, 30]. The proposed approaches in this group usually are designed based on the central catadioptric camera model in order to detect all robot teammates at the same instant. The subject of the third group is to track a predictable moving target, such as a projectile or straight moving ball, for mobile robot interception purpose [31, 32]. In [31], Borgstadt *et al.* utilized a human vision-based strategy to guide a mobile robot to intercept a projectile ball. Similarly, Capparella *et al.* extended the concept of human-like strategy to develop a vision-based two-level interception approach, which contains a lower level controller to control the on-board pan-tilt camera and a higher lever controller to operate the mobile robot platform, for intercepting a straight moving ball [32]. A common point of the second and third group is that the motion of the target of interest is known and predictable. However, in some robotic applications, a mobile robot requires to track a dynamic and unpredictable motion target, such as a human's face, for the purpose of pursuit or interaction. Thus, the existent methods of the aforementioned two groups are not suitable to solve the dynamic moving target visual tracking control problem.

The purpose of the fourth group aims to solve the problem of tracking a dynamic moving target [33-37]. In [33], Wang *et al.* proposed an adaptive backstepping control law based on an image-based camera-target visual interaction model to track a dynamic moving target with unknown height parameter. Although the approach in [33] guarantees the asymptotic stability of closed-loop visual tracking control system in tracking a dynamic moving target, the case of tracking a static target cannot be guaranteed due to the non-zero restrictions on the reference velocity of the mobile robot. In [34], Song *et al.* combined a face detection algorithm with a PID controller to track a moving person in a home setting. The main disadvantage of their method is that it cannot guarantee the stability of the closed-loop visual tracking system based on a stability criterion. In [35], Malis *et al.* integrated template-based visual tracking algorithms and model-free vision-based control techniques to build a flexible and robust visual tracking control system for various robotic applications. Because their visual tracking result is based on the homography estimation, which requires two images of the target pattern to estimate the optimal homography, the reported system only overcomes the partial occlusion problem but fails in the fully occlusion problem. In [36], Han *et al*. proposed an image-based visual tracking control scheme for a mobile robot to estimate the position of the target in the next image and track the target to the central area of the image. Since their method utilized the differential approximation method to estimate the velocity of target in the image plane, the estimation result is very sensitive to the image noise. Recently, a visual interaction controller had been proposed for a unicycle-modeled mobile robot to track a dynamic moving target such as human's face [37]. The drawback of this method is that the controller requires the target's 3D motion velocity, which is difficult to estimate when only a monocular camera is used.

Therefore, from the literature survey, one of the most important challenges in mobile robot visual tracking control design is to develop a visual tracking control system to estimate the motion of the dynamic moving target and track it based on a stability criterion. Further, in realization of the control schemes, it has been noted that the disturbances of image noise, velocity quantization error and temporary (partial/full) occlusion degrade the performance of the controller and might make the system unstable. These problems have not yet been clarified in many existent related works and hence motivated us to investigate the robustness of the visual tracking control system against the uncertainties of image noise, system model, velocity quantization and temporary occlusion.

1.3 Research Objectives

The objective of this thesis is to develop a novel CFA interpolation algorithm for color reproduction of a single image sensor and a robust visual tracking control system for vision-based motion control of a wheeled mobile robot. For color reproduction, a novel heterogeneity-projection hard-decision (HPHD) CFA interpolation algorithm is proposed to reconstruct the full-color images from Bayer mosaic images [38]. The proposed HPHD CFA interpolation algorithm can combine with many existent image interpolation methods to reconstruct each color plane. However, in CFA interpolation process, it should be noticed that color artifacts existing around edges and fine textures of the demosaiced image are a factor limiting performance. Color artifacts are caused primarily by aliasing error in high-frequency regions such as edges or fine textures, and existing algorithms are unable to resolve color artifacts in these regions effectively to obtain demosaiced results with high visual quality. To effectively reduce color artifacts in demosaiced images, a novel color-difference based edge-adaptive (CDEA) CFA interpolation algorithm is then proposed and combined with the HPHD CFA interpolation algorithm to reproduce color values by exploiting the green plane information for making high-frequency components of red and blue planes similar to the green plane, in which the aliasing errors are usually much smaller than those in red and blue planes.

For visual tracking control, a novel design of a robust visual tracking control system, which consists of a visual tracking controller (VTC) and a visual state estimator (VSE), is proposed for a wheeled mobile robot equipped with a tilt camera. In order to resolve the problem of visual tracking a dynamic moving target and guarantee the stability of the closed-loop visual tracking system, a novel dual-Jacobian visual interaction model is derived to help the design of VTC and VSE. The VSE is constructed by a real-time self-tuning Kalman filter and aims to estimate the optimal system state and target motion in the image plane directly for later use by the VTC. The VTC then calculates the robot's control velocities in the image plane directly. To handle the uncertainties encountered in practical visual tracking control system, the VSE can estimate the optimal target state from the observed image with disturbances to overcome both image noise and temporary occlusion uncertainties.

On the other hand, the VTC not only possesses some degree of robustness against the system model uncertainties, but also overcomes the unmodelled quantization effect in the velocity commands. Therefore, the proposed system is robust to the uncertainties of image noise, system model, velocity quantization and temporary occlusion. This advantage enhances the reliability of the proposed visual tracking control system in practical applications.

1.4 Organization of the Thesis

The remainder of this thesis is organized as follows: In Chapter 2, a novel HPHD CFA interpolation algorithm is first developed based on the concept of spectral-spatial correlation. A new CDEA CFA interpolation algorithm is then proposed to combine with HPHD CFA interpolation algorithm for color reproduction of Bayer mosaic images. In Chapter 3, a novel dual-Jacobian visual interaction model is derived in order to help the design of mobile robot visual tracking control for tracking a dynamic moving target. Accordingly, the results of VTC design with robustness analysis are developed by using the proposed visual interaction model. Furthermore, the robust control law to overcome the velocity quantization error encountered in practical systems will also be presented. Chapter 4 develops the VSE using Kalman filter with self-tuning algorithm to estimate the optimal system state in the image plane for handling the uncertainties caused by image noise and temporary occlusion. Simulation and experimental results of the proposed CFA interpolation algorithm and the proposed mobile robot visual tracking control system are reported and discussed in Chapter 5. Chapter 6 concludes the contributions of this work and provides the recommendations for future research. In Appendix A, an experiment of tweaking parameters is presented to find the local optimal parameters for the proposed HPHD-CDEA CFA interpolation algorithm described in Chapter 2. Appendixes B and C present more discussions and visual comparisons of the proposed CFA interpolation algorithm. An extended discussion on the proposed VTC scheme is presented in Appendix D. Figure 1-1 shows the simplified overview of the thesis.

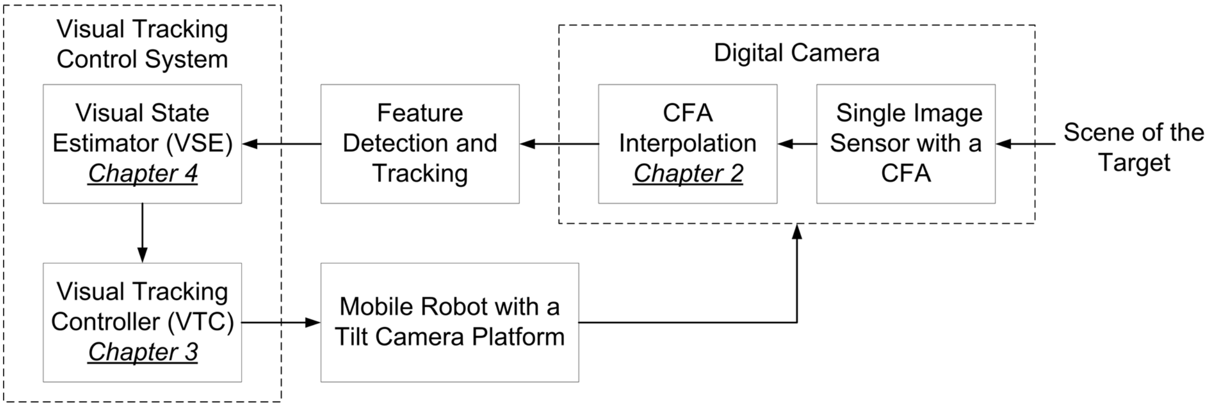

Visual Tracking Control System
Visual State Estimator (VSE) Chapter 4
Visual Tracking Controller (VTC) Chapter 3
Feature Detection and Tracking
Mobile Robot with a Tilt Camera Platform
Digital Camera
CFA Interpolation Chapter 2
Single Image Sensor with a CFA
Scene of the Target

Fig. 1-1: Structure of the thesis.

Chapter 2

CFA Interpolation for Color Reproduction of Bayer Mosaic Images

2.1 Introduction

In this chapter, the design of the proposed HPHD-CDEA CFA interpolation algorithm is presented for color reproduction from Bayer mosaic images. The proposed algorithm consists of HPHD algorithm and CDEA CFA interpolation. The proposed HPHD algorithm aims to estimate the optimal interpolation direction before performing CFA interpolation. Because the decision stage is performed before the interpolation stage (termed as hard-decision interpolation), each pixel only needs to be interpolated once. To do so, a new heterogeneity-projection scheme based on a novel spectral-spatial correlation concept is proposed to estimate the best interpolation direction directly from the original Bayer mosaic image. Using the proposed heterogeneity-projection scheme, a hard-decision rule can be decided before performing CFA interpolation.

In order to effectively reduce color artifacts in CFA interpolation process, the proposed CDEA CFA interpolation algorithm is then combined with the HPHD algorithm to reconstruct the red and blue color planes by exploiting the green plane information for making high-frequency components of red and blue planes similar to the green plane, in which the aliasing errors are usually much smaller than those in red and blue planes. To do so, the red and blue channels are first reconstructed using bilinear interpolation and then edge-adaptive filtered in color-difference space. To reconstruct the green plane, any existent image interpolation methods can be employed to combine with the CDEA CFA interpolation algorithm. Moreover, we also present a new edge-adaptive interpolation method to reconstruct the green channel from CFA samples. The advantages of the proposed HPHD-CDEA CFA interpolation algorithm are summarized as follows:

1) Many existent CFA interpolation methods can be combined with the proposed HPHD algorithm to reconstruct each color plane. For example, the proposed heterogeneity-projection scheme can combine with existent decision-based CFA interpolation algorithms. More specifically, the proposed heterogeneity-projection scheme can adopt into the decision step of existent decision-based CFA interpolation algorithms.

2) Each pixel only has to be interpolated once. Therefore, the proposed algorithm is much more efficient than other decision-based schemes.

3) Any existing image interpolation methods can be combined with the proposed CDEA CFA interpolation algorithm to reconstruct the green plane.

4) The proposed HPHD-CDEA CFA interpolation algorithm performs well not only in texture regions, but also in well-defined edges of the image.

The rest of this chapter is organized as follows: In Section 2.2, the proposed HPHD CFA interpolation algorithm is designed by using a novel concept of spectral-spatial correlation. Section 2.3 presents the proposed CDEA CFA interpolation algorithm based on color-difference model. A new edge-adaptive interpolation method to reconstruct the green plane from CFA samples is also presented in this section. Section 2.4 describes the complete HPHD-CDEA CFA interpolation algorithm and gives an example study. Section 2.5 summarizes the contributions of this work. An experiment of tweaking parameters to find the local optimal parameters for the proposed CFA interpolation algorithm is presented in Appendix A. More discussions and visual comparisons are presented in Appendixes B and C.

2.2 Heterogeneity-Projection Hard-Decision (HPHD) CFA interpolation

Figure 2-1 shows the most used CFA pattern, the Bayer pattern [38], where R, G and B denote, respectively, the pixels having only red, green and blue color values. We limit our discussion to the Bayer pattern in this chapter because it is so popular. In the following, image spectral and spatial correlations are first introduced. A novel spectral-spatial correlation is then derived based on these two correlations.

2.2.1 Spectral and Spatial Correlations

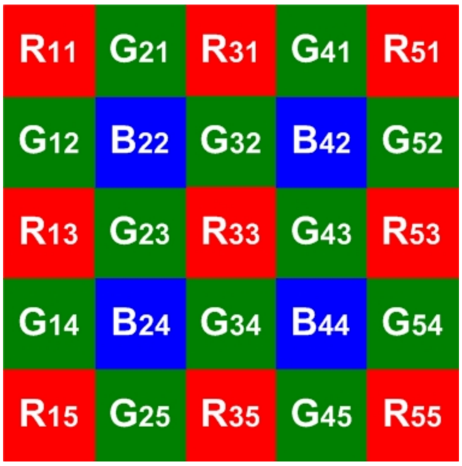
Many existing demosaicing methods are developed using image spectral and/or spatial correlation. The concept of spectral correlation is based on the assumption that the color difference signals are locally constant in chrominance smooth areas [39]. Let $[R \quad G \quad B]$ denote three color planes of a nature color image, the concept of spectral correlation leads to the following assumption.

A1) The color differences between green and red/blue channels satisfy the following conditions

$$R(x,y) = G(x,y) + A_{rg}(x,y) \text{ and } B(x,y) = G(x,y) + A_{bg}(x,y),$$

where $A_{rg}(x,y)$ and $A_{bg}(x,y)$ are piecewise constant within the boundary of a given object.

The spatial correlation reflects the fact that within a homogeneous image region, neighboring pixels share similar color values [40]. In other words, the difference between neighboring pixel values along an edge direction in spatial domain is a constant. Thus we have the following assumption based on the concept of spatial correlation [3].



R11	G21	R31	G41	R51
G12	B22	G32	B42	G52
R13	G23	R33	G43	R53
G14	B24	G34	B44	G54
R15	G25	R35	G45	R55

Fig. 2-1: The Bayer pattern.

A2) The rate of change of neighboring pixel values along an edge direction is a constant.

To illustrate this, let us consider the interpolation of R_{33} in Fig. 2-1. Suppose that the pixel R_{33} is located on a horizontal edge. Based on **A1)**, the neighboring pixels of R_{33} along the horizontal direction have the following relationship between green and red/blue pixel values

$$A_{rg}(x-1,y)=A_{rg}(x,y)=A_{rg}(x+1,y) \text{ and}$$

$$A_{bg}(x-1,y)=A_{bg}(x,y)=A_{bg}(x+1,y). \qquad (2.1)$$

So we have

$$\overline{R}_{23}-G_{23}=R_{33}-\overline{G}_{33}=\overline{R}_{43}-G_{43} \text{ and } \overline{B}_{23}-G_{23}=\overline{B}_{33}-\overline{G}_{33}=\overline{B}_{43}-G_{43}. \qquad (2.2)$$

The assumption **A2)** gives the following relationship on horizontal edges

$$\overline{R}_{23}-R_{33}=R_{33}-\overline{R}_{43}\equiv dR_h,$$

$$G_{23}-\overline{G}_{33}=\overline{G}_{33}-G_{43}\equiv dG_h, \text{ and} \qquad (2.3)$$

$$\overline{B}_{23}-\overline{B}_{33}=\overline{B}_{33}-\overline{B}_{43}\equiv dB_h,$$

where $\overline{G}_{33}$, $\overline{R}_{23}$, $\overline{R}_{43}$, $\overline{B}_{23}$, $\overline{B}_{33}$, and $\overline{B}_{43}$ denote the missing color values at the respective pixel locations. dG_h, dR_h and dB_h are constants.

2.2.2 Spectral-Spatial Correlation (SSC)

A significant characteristic of Bayer pattern is that for each pixel, the surrounding pixels are one of the primary components in different channels. It is then interesting to investigate the relationship between neighboring pixels in different color channels. Consider the following situation: on a horizontal edge, two green pixels surround a red pixel on horizontal direction. Take the difference between the center red pixel and right green pixel, we then have

$$R(x,y)-G(x+1,y)=\left[R(x,y)-\overline{G}(x,y)\right]+\left[\overline{G}(x,y)-G(x+1,y)\right], \qquad (2.4)$$

where $\overline{G}(x,y)$ denotes the missing green value at center red pixel location. Recall assumptions **A1)** and **A2)**, expression (2.4) becomes such that

$$S_{rg}^{h(x,x+1)} \equiv R(x,y)-G(x+1,y)=A_{rg}(x,y)+dG_h\,. \qquad (2.5)$$

Similarly, the difference between a blue pixel and its right green pixel is given by

$$S_{bg}^{h(x,x+1)} \equiv B(x,y)-G(x+1,y)=A_{bg}(x,y)+dG_h\,. \qquad (2.6)$$

The same results also can be obtained along vertical direction on a vertical edge such that

$$S_{rg}^{v(y,y+1)} \equiv R(x,y)-G(x,y+1)=A_{rg}(x,y)+dG_v\text{, and}$$

$$S_{bg}^{v(y,y+1)} \equiv B(x,y)-G(x,y+1)=A_{bg}(x,y)+dG_v\,. \qquad (2.7)$$

Expressions (2.5)-(2.7) show that the difference between surrounding pixels in different color channels is equal to the summation of spectral and spatial correlations. We refer these relationships (2.5)-(2.7) as *spectral-spatial correlation* (SSC). SSC has two important characteristics. First, SSC can be easily and directly calculated from the original Bayer mosaic image. Second, SSC inherits the characteristics of spectral and spatial correlations. In other words, SSC is also piecewise constant within the boundary of a given object or along an edge direction. Therefore, we have the following assumption based on these observations:

A3) The SSC defined in (2.5)-(2.7) within the boundary of a given object or along an edge direction is also piecewise constant.

Assumption **A3)** is a significant clue for us to find the directional smooth regions in Bayer mosaic images directly before performing the interpolation. In the following section, we will present the method of heterogeneity-projection based on **A3)**.

2.2.3 Heterogeneity-Projection for Bayer Mosaic Images

The proposed heterogeneity-projection scheme transfers the original Bayer mosaic image directly into horizontal and vertical heterogeneity maps, respectively. Using these two heterogeneity maps, the interpolation direction can be determined easily by choosing the smallest heterogeneity values.

Assumption **A3)** implies that the n-order directional finite derivative of SSC along an edge direction tends toward a small value. For example, consider a red pixel $R(x,y)$ locates on a horizontal edge, the SSC values of $R(x,y)$ and its neighboring pixels along horizontal direction can be found such that

$$S_{rg}^{h(x,x+1)} = A_{rg}(x,y) + dG_h\,,\quad S_{gr}^{h(x+1,x+2)} = -A_{rg}(x+2,y) + dG_h\,, \tag{2.8}$$

where $S_{gr}^{h(x+1,x+2)} \equiv G(x+1,y) - R(x+2,y)$. Based on the basic definition of the first-order derivative of a one-dimensional discrete function, the first-order horizontal derivative of SSC are given by [41]

$$dS_{rg}^{h(x,x+3)} \equiv S_{rg}^{h(x,x+1)} - S_{rg}^{h(x+2,x+3)} = A_{rg}(x,y) - A_{rg}(x+2,y)\,,$$

$$dS_{gr}^{h(x+1,x+4)} \equiv S_{gr}^{h(x+1,x+2)} - S_{gr}^{h(x+3,x+4)} = A_{rg}(x+4,y) - A_{rg}(x+2,y)\,. \tag{2.9}$$

Recall **A1)** and **A3)**, one can see that $dS_{rg}^{h(x,x+3)}$ and $dS_{gr}^{h(x+1,x+4)}$ both will approach to zero along this horizontal edge. Because the higher-order derivative of a discrete function is a linear combination of the first-order ones, it implies the higher-order horizontal derivative of SSC will also approach to zero along the horizontal edge. Thus we have the following assumption.

A4) If pixels locate on a directional edge, then the corresponding nth-order directional derivative of SSC along the edge direction approaches to zero.

Assumption **A4)** poses a question that how the nth-order directional derivative of SSC can be directly calculated from Bayer mosaic image. To resolve this problem, a heterogeneity-projection scheme is developed to transfer the row data of Bayer mosaic image directly into nth-order directional derivative of SSC. Note that the value of nth-order directional derivative of SSC is defined as *heterogeneity measure*, because it leads to a small value within a directional smooth region.

Denote $RG_{1\times N} = [R_1 \quad G_2 \quad R_3 \quad \cdots]_{1\times N}$ as a row data of Bayer mosaic image, N is the presetting window size, and H_h is the corresponding horizontal heterogeneity value. To calculate the horizontal heterogeneity value H_h from $RG_{1\times N}$, we propose the following steps. First, the row data $RG_{1\times N}$ is transferred into a $1\times(N-3)$ vector of first-order horizontal derivative of SSC using a linear transformation such that

$$dS_{1\times(N-3} = [dS_{rg}^{h(1,4)} \quad dS_{gr}^{h(2,5)} \quad dS_{rg}^{h(3,6)} \quad \cdots]_{1\times(N-3)} = RG_{1\times N}T^{1}_{N\times(N-3)}, \qquad (2.10)$$

where $T^{1}_{N\times(N-3)} = [1 \quad -1 \quad -1 \quad 1]^T \otimes eye(N-3)$, $\otimes$ denotes the 2D convolution operator and $eye(M)$ denotes a $M\times M$ identity matrix. Second, because the high-order derivative of a discrete function is derived by the linear combination of its first-order ones, the horizontal heterogeneity value H_h, the $(N-3)^{\text{th}}$-order horizontal derivative of SSC, is obtained such that [41]

$$\begin{aligned} H_h \equiv d^{N-3} S_{rg}^{h(1,N)} &= d^{N-4} S_{rg}^{h(1,N-1)} - d^{N-4} S_{gr}^{h(2,N)} \\ &= d^{N-5} S_{rg}^{h(1,N-2)} - 2d^{N-5} S_{gr}^{h(2,N-1)} + d^{N-5} S_{rg}^{h(3,N)} \\ &\vdots \\ &= w_1 dS_{rg}^{h(1,4)} + w_2 dS_{gr}^{h(2,5)} + w_3 dS_{rg}^{h(3,6)} + \cdots \\ &= dS_{1\times(N-3)} T^2_{(N-3)\times 1} \end{aligned} \tag{2.11}$$

where $T^2_{(N-3)\times 1} = \prod_{i=1}^{N-4} [1 \quad -1]^T \otimes eye(N-3-i)$ is a $(N-3)\times 1$ coefficient vector which transfers vector $dS_{1\times(N-3)}$ into the $(N-3)^{\text{th}}$-order derivative value through Euclidean inner product [42]. Next, substituting (2.10) into (2.11) yields

$$H_h = RG_{1\times N} T^1_{N\times(N-3)} T^2_{(N-3)\times 1} = RG_{1\times N} P_{N\times 1}, \tag{2.12}$$

where $P_{N\times 1} = T^1_{N\times(N-3)} T^2_{(N-3)\times 1}$ is a $N\times 1$ vector and referred as *heterogeneity vector*. Expression (2.12) shows that the horizontal heterogeneity value H_h is the projection of the row data of Bayer mosaic image onto the heterogeneity vector $P_{N\times 1}$. Thus expression (2.12) is termed as *horizontal heterogeneity-projection*. Figure 2-2 illustrates an example of horizontal heterogeneity-projection from a 1×5 row data of Bayer mosaic image. Using (2.12), the heterogeneity vector $P_{N\times 1}$ is obtained as

$$P_{5\times 1} = T^1_{5\times 2} T^2_{2\times 1} = [1 \quad -2 \quad 0 \quad 2 \quad -1]^T.$$

The horizontal heterogeneity value H_h of R_{33} is then given by

$$H_h = d^2 S_{rg}^{h(1,5)} = RG_{1\times 5} P_{5\times 1} = R_{13} - 2G_{23} + 2G_{43} - R_{53}.$$

Similarly, the vertical heterogeneity value H_v is the projection of Bayer mosaic image's column data onto the heterogeneity vector $P_{N\times 1}$ such that

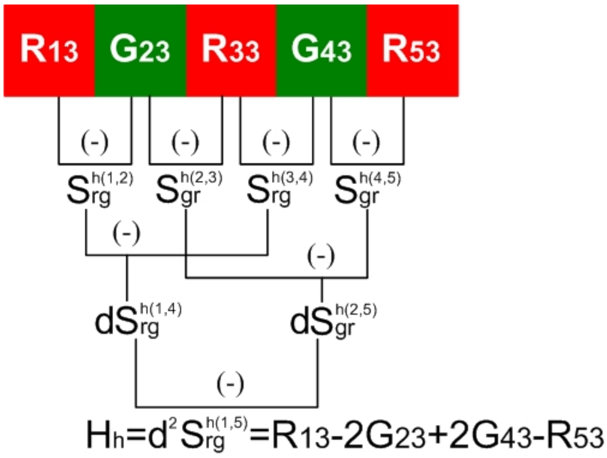


Fig. 2-2: The concept of horizontal heterogeneity-projection from a 1×5 row data of a Bayer mosaic image.

$$H_v = RG_{N\times 1}^{T} P_{N\times 1}, \tag{2.13}$$

where $RG_{N\times 1} = [R_1 \quad G_2 \quad R_3 \quad \cdots]_{N\times 1}^{T}$ is a column data of Bayer mosaic image. Finally, based on (2.12) and (2.13), the horizontal and vertical heterogeneity maps, H_{h_map} and H_{v_map} are obtained, respectively by

$$H_{h\_map} = \left| Bayer \otimes P_{N\times 1}^{T} \right|, \text{ and } H_{v\_map} = \left| Bayer \otimes P_{N\times 1} \right|, \tag{2.14}$$

where *Bayer* denotes the original Bayer mosaic image. One can see from (2.14) that the horizontal and vertical heterogeneity maps are derived directly from the Bayer mosaic image via horizontal and vertical heterogeneity-projection, respectively.

2.2.4 Directional Adaptive Filtering for Error Reduction

Assumption **A4)** states that the directional heterogeneity-projection along an edge direction leads to a small heterogeneity value. However, a small heterogeneity measure does not imply the directional heterogeneity-projection along a right edge direction. This problem will induce the estimation errors in the initial estimated heterogeneity maps. In order to reduce the estimation errors, a directional adaptive filter, whose behavior changes based on the statistical characteristics inside a local window, is proposed to reduce the estimation error and estimate the optimal heterogeneity maps. Moreover, since each heterogeneity measure in the initial heterogeneity maps is static, this estimation problem is equivalent to the static estimation problem, in which the estimation errors are modeled as the zero mean Gaussian noises with non-zero variance. According to [43], the minimum mean square-error (MMSE) solution of the static estimation problem can be estimated using a predictor-corrector filter. Therefore, the design of the proposed directional adaptive filter adopts the structure of predictor-corrector filter to obtain the MMSE estimates.

The proposed directional adaptive filter is divided into horizontal and vertical adaptive filters. For the horizontal heterogeneity map, only the horizontal adaptive filter is applied to it. Figure 2-3(a) illustrates the concept of horizontal adaptive filter. In Fig. 2-3(a), the center pixel H_h is to be adaptively filtered along the horizontal direction based on statistical measures of surrounding pixels H_h^R and H_h^L. The simplest statistical measures of H_h^R and H_h^L are their mean and variance in a local window [41]. For instance, if a 1×3 rectangular window defines the window size, the local mean and variance of H_h^R and H_h^L are, respectively, given by

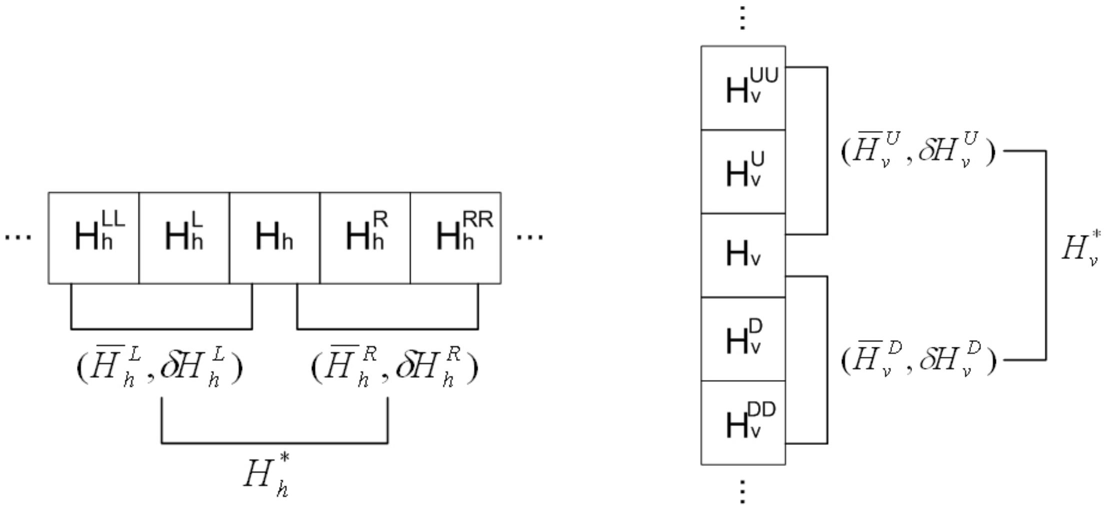
$$\overline{H}_h^R = (H_h + H_h^R + H_h^{RR})/3\,,\quad \delta H_h^R = [(\overline{H}_h^R - H_h)^2 + (\overline{H}_h^R - H_h^R)^2 + (\overline{H}_h^R - H_h^{RR})^2]/3\,. \quad (2.15)$$

$$\overline{H}_h^L = (H_h + H_h^L + H_h^{LL})/3\,,\quad \delta H_h^L = [(\overline{H}_h^L - H_h)^2 + (\overline{H}_h^L - H_h^L)^2 + (\overline{H}_h^L - H_h^{LL})^2]/3\,. \quad (2.16)$$

Using (2.15) and (2.16), the adaptively filtered pixel H_h^* is obtained below

$$H_h^* = \overline{H}_h^L + \frac{\delta H_h^L}{\delta H_h^L + \delta H_h^R}(\overline{H}_h^R - \overline{H}_h^L)\,. \quad (2.17)$$

In (2.17), the local mean $\overline{H}_h^L$ is the predictor term with an associated error variance δH_h^L, and the local mean $\overline{H}_h^R$ is the corresponding corrector term with error variance δH_h^R. Thus,



H_h^{LL}
H_h^{L}
H_h
H_h^{R}
H_h^{RR}
$(\overline{H}_h^L, \delta H_h^L)$
$(\overline{H}_h^R, \delta H_h^R)$
H_h^*
H_v^{UU}
H_v^{U}
H_v
H_v^{D}
H_v^{DD}
$(\overline{H}_v^U, \delta H_v^U)$
$(\overline{H}_v^D, \delta H_v^D)$
H_v^*

(a) (b)

Fig. 2-3: The concept of (a) horizontal and (b) vertical adaptive filtering using a 1×3 rectangular window.

(2.17) provides the MMSE estimate of horizontal heterogeneity measure in a local window. Figure 2-3(b) illustrates an example of vertical adaptive filter for vertical heterogeneity map. Using the same procedure discussed above, the adaptively filtered pixel H_v^* is obtained as follows

$$H_v^* = \overline{H}_v^U + \frac{\delta H_v^U}{\delta H_v^U + \delta H_v^D}(\overline{H}_v^D - \overline{H}_v^U), \tag{2.18}$$

where $(\overline{H}_v^U, \delta H_v^U)$ and $(\overline{H}_v^D, \delta H_v^D)$ are the local mean and variance of H_v^U and H_v^D. Similarly, (2.18) also provides the MMSE estimate of vertical heterogeneity measure in a local window. After adopting the horizontal and vertical adaptive filters presented above into horizontal and vertical heterogeneity maps, respectively, the MMSE estimates of horizontal and vertical heterogeneity maps $H_{h\_map}^*$ and $H_{v\_map}^*$ are obtained.

2.2.5 Hard-Decision CFA interpolation

With the horizontal and vertical heterogeneity maps, a hard-decision rule is applied for CFA interpolation. First, we classify three subsets in the image such that

$$\Omega_h \equiv \left\{(x,y) \mid H^*_{h\_map}(x,y) < \alpha H^*_{v\_map}(x,y)\right\},$$

$$\Omega_v \equiv \left\{(x,y) \mid H^*_{v\_map}(x,y) < \alpha H^*_{h\_map}(x,y)\right\}, \tag{2.19}$$

$$\Omega_s \equiv \left\{(x,y) \mid (x,y) \notin \Omega_h, (x,y) \notin \Omega_v\right\},$$

where Ω_h, Ω_v, and Ω_s denote the horizontal, vertical, and smooth subsets, respectively. α is a positive constant satisfying $0 \le \alpha \le 1$. Second, based on (2.19), the concept of hard-decision rule for CFA interpolation is obtained

```
if $(x,y) \in \Omega_h$
    Perform horizontal interpolation on each missing color channel;
elseif $(x,y) \in \Omega_v$
    Perform vertical interpolation on each missing color channel;          (2.20)
else
    Perform weight averaging of neighboring pixels on each missing color channel.
```

In the following discussion, the CFA interpolation method is developed based on the hard-decision rule (2.20).

Remark 2.1: The parameter α in (2.19) determines the size of smooth subset in the image. A small (large) α leads to a large (small) smooth subset in the image. For example, if $\alpha = 0$, the image only contains smooth subset without horizontal and vertical subsets. Based on (2.20), the interpolation of image only adopts the weight averaging of neighboring pixels on each missing color channel [3, 11, 40]. On the other hand, for $\alpha = 1$, the image only contains horizontal and vertical subsets but without smooth subset and the interpolation of image only adopts horizontal and vertical interpolations on each missing color channel [7-9]. Therefore, for $0 < \alpha < 1$, the hard-decision rule (2.20) is characterized by features of weight averaging and directional interpolating.

2.3 Color-Difference Based Edge-Adaptive (CDEA) CFA interpolation

In this section, a novel CDEA CFA interpolation algorithm is proposed to reproduce the missing color values in Bayer mosaic images. The proposed CFA interpolation algorithm aims to exploit the green channel information for making high-frequency components of red and blue channels similar to the green channel, which is useful to effectively reduce color artifacts in demosaiced images. Any of the existing image interpolation methods can be combined with the proposed algorithm to reconstruct the green channel; but, a new edge-adaptive interpolation method to interpolate the green channel from CFA samples will also be presented in this section. The red and blue channels are first reconstructed using bilinear interpolation and then edge-adaptive low-pass filtered in color-difference space by the assistance of the reconstructed green channel.

2.3.1 Color-Difference Approach to CFA interpolation

In a Bayer pattern, green samples are obtained on a quincunx, while red and blue samples are obtained in rectangular lattices. The density of red and blue samples is one-half that of the green ones, and the aliasing error of high-frequency components in green channel is likely to be less than that in red and blue channels. Thus, a common problem in demosaicing is that the visible color artifacts in high-frequency regions are caused primarily by aliasing in the red and blue channels. Fortunately, there is usually high inter-channel correlation in high-frequency regions among red, green, and blue channels for natural color images [5]. This implies that the red, green, and blue channels are quite similar at fine texture and edge locations with all three colors. Therefore, a valid assumption can be made that object boundaries are the same in all three color channels. More specifically, we have the following assumption.

A5) The high-frequency regions are similar in all three channels and close to the

Fig. 2-4: Test images used in the experiment.

high-frequency regions of the green channel.

In order to validate assumption **A5)**, we utilize twenty-five natural images from the Kodak PhotoCD (see Fig. 2-4), which have been used as test images for several demosaicing studies [3-10].

Figure 2-5 shows the flowchart for demonstrating the assumption of color-difference model mentioned above. The key concept is to replace the high-frequency components of red and blue planes by using those of green plane, and compare then the mean squared error (MSE) between the original and reconstructed color planes. A low-pass filter is utilized for red and blue planes and a high-pass filter for the green plane. We utilize 2-D ideal low-pass and high-pass filters in this procedure. Their transfer functions are given by [41]:

$$H_{lowpass}(u,v)=\begin{cases}1, & if\ D(u,v)\le D_0\\ 0, & if\ D(u,v)>D_0\end{cases} \text{ and } H_{highpass}(u,v)=\begin{cases}0, & if\ D(u,v)\le D_0\\ 1, & if\ D(u,v)>D_0\end{cases},$$

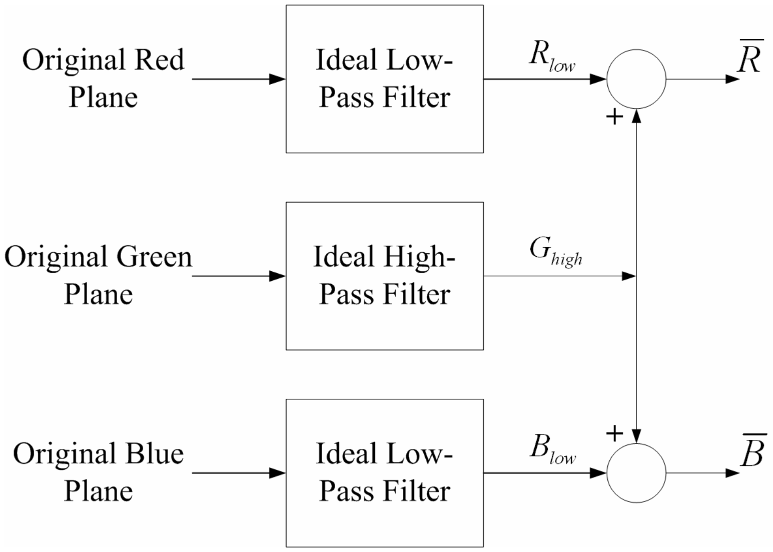


Fig. 2-5: Flowchart for demonstrating the assumption of color-difference model.

where D_0 is a specified nonnegative quantity; and $D(u,v)$ is the distance from point (u,v) to the origin of the frequency plane. We set D_0 equal to 128 in this test. After filtering in each color plane, the new red and blue planes, $\overline{R}$ and $\overline{B}$, are reconstructed respectively by adding the high-frequency components of the green plane to their low-frequency components. Table 2-1 records the MSE comparison results of each step. The first and second columns show the MSE between original and low-pass filtered red (blue) planes R_{low} (B_{low}). The third and fourth columns show the MSE between original and reconstructed red (blue) planes $\overline{R}$ ($\overline{B}$). From the test results, it is clear that the MSE is reduced effectively by adding the high-frequency regions of the green plane G_{high} to the low-pass filtered red (blue) planes. This implies that the high-frequency regions of red and blue planes are similar and close to the high-frequency regions of the green plane. Thus, assumption **A5)** is validated. Based on assumption **A5)**, our motivation in this study is to reduce the color artifacts in high-frequency regions by adding the high-frequency information of green channel to other color channels. As described below, this can be achieved by utilizing the color-difference model.

Let $[R^d \quad G^d \quad B^d]$ denote three color planes of a demosaiced image. The Fourier spectrum of each color plane can be described as follows:

Table 2-1: Comparison of mean squared error at each step in Fig. 2-5.

Image No.	MSE(R, R_{low})	MSE(B, B_{low})	MSE($R, \overline{R}$)	MSE($B, \overline{B}$)
1	147.3784	123.2171	37.0369	8.7712
2	260.2865	263.2530	4.2306	2.9049
3	73.0098	66.0994	9.8449	1.6476
4	59.1224	55.4568	3.1630	4.0863
5	75.0895	71.0032	8.0543	1.6137
6	307.8661	293.3846	6.3342	9.2040
7	178.8668	174.3140	2.1157	4.2843
8	76.3988	76.0140	2.8284	3.3626
9	539.2621	526.6953	8.2720	7.8362
10	81.2787	77.1119	2.6365	3.9927
11	86.7402	89.3761	2.8525	4.2737
12	149.5650	139.4246	4.1582	2.2200
13	70.4837	74.3636	3.1482	2.9343
14	415.0555	421.2574	3.3856	10.3724
15	168.1822	141.2964	9.1458	8.2062
16	112.1141	103.0222	12.2228	3.6253
17	82.9777	80.8168	1.3618	1.8066
18	81.5764	84.8154	2.2567	3.9789
19	187.1993	182.8144	6.1265	10.0651
20	179.4598	163.5454	2.7882	3.8415
21	119.7853	109.3889	2.4227	6.2912
22	155.7798	160.6295	2.6352	4.9741
23	124.3185	129.3350	7.7838	7.3834
24	57.6438	55.0500	4.2972	4.7799
25	232.1675	302.3846	10.3759	19.5268

$$F[R^d]=F[R^d]_l+F[R^d]_h,\quad F[G^d]=F[G^d]_l+F[G^d]_h,\quad F[B^d]=F[B^d]_l+F[B^d]_h, \tag{2.21}$$

where $F[\bullet]$ denotes the 2-D discrete Fourier transform; and the underscores *l* and *h* stand for low-frequency and high-frequency components, respectively. The color-difference models of the demosaiced image are defined such that

$$R_g = R^d - G^d,\quad B_g = B^d - G^d. \tag{2.22}$$

Let $L\{\bullet\}$ denote a linear low-pass filtering process, and $\widetilde{R}_g$ and $\widetilde{B}_g$ denote the low-frequency regions of the color differences corresponding to R_g and B_g. Suppose that the high-frequency components of the color differences R_g and B_g can be removed by the low-pass filtering process, the Fourier spectrum of $\widetilde{R}_g$ and $\widetilde{B}_g$ can be described such that

$$F[\widetilde{R}_g]=L\{F[R_g]\}=F[R^d]_l-F[G^d]_l,\quad F[\widetilde{B}_g]=L\{F[B_g]\}=F[B^d]_l-F[G^d]_l. \tag{2.23}$$

Subsequently, the new red and blue planes of the demosaiced image, $\overline{R}^d$ and $\overline{B}^d$, can be obtained respectively, by adding $\widetilde{R}_g$ and $\widetilde{B}_g$ with G^d. Their Fourier spectra are given by

$$F[\overline{R}^d] = F[\widetilde{R}_g + G^d] = F[R^d]_l + F[G^d]_h, \quad F[\overline{B}^d] = F[\widetilde{B}_g + G^d] = F[B^d]_l + F[G^d]_h. \quad (2.24)$$

It is clear from (2.24) that the high-frequency components of the new red and blue planes of the demosaiced image are replaced by the high-frequency components of the green plane. Because the aliasing error in the green plane is usually much smaller than those in red and blue planes, based on the assumption described above, the aliased errors in red and blue channels can be efficiently reduced by linear low-pass filtering in the color-difference spaces and adding the results with green channel to obtain the new ones. This observation leads to the development of an efficient CFA interpolation algorithm based on color-difference that can reduce the color artifacts in high-frequency regions such as edges or fine textures.

The color-ratio model has been another useful model for the development CFA interpolation algorithms [1, 11]. The main difference between the color-difference model and the color-ratio model is that the latter assumes the ratios between the red and green values are constant within a given object, as are the ratios between the blue and green values. However, the color-ratio model usually fails to work around edge regions and results in some color artifacts because the constant-ratio assumption is not valid in these regions. But, again, the assumption used by color-difference model is that the high-frequency regions are similar in all three channels and close to the high-frequency regions of the green channel. If the high-frequency components of the green channel (such as edges and fine textures) can be recovered within small aliasing errors, then the results can be used to effectively reduce the aliasing errors in red and blue channels. In the following section, we describe the proposed edge-adaptive CFA interpolation algorithm based on the color-difference model.

2.3.2 CDEA Low-pass Filtering

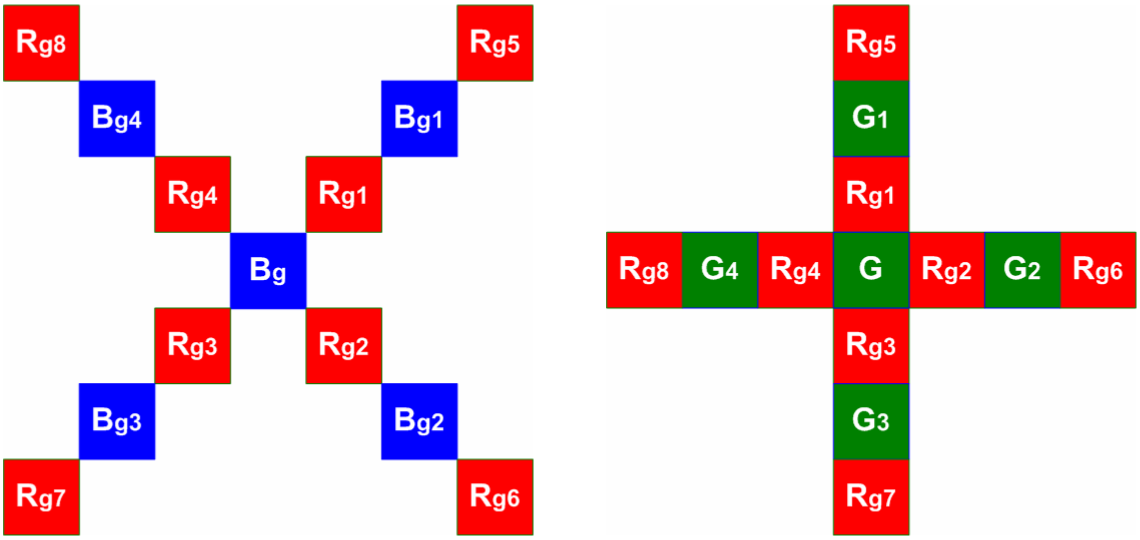
To begin with derivation of the proposed CFA interpolation algorithm, we first assume that the green channel has been fully recovered by using an existing image interpolation method. The initial estimation of red and blue channels can be obtained, for instance, by utilizing some well-known method such as bilinear interpolation. Note that at this stage we do not process the original red (blue) values, but rather keep them the same as the original CFA-sampled color values. When an initial demosaiced image is obtained, we then utilize an adaptive low-pass filter to filter the color-difference value at the missing pixel locations in red (blue) channels.

Let $[R_i^d \quad G_i^d \quad B_i^d]$ denote three color planes of the initial demosaiced image. The red and blue color-difference planes are given by

$$R_g = R_i^d - G_i^d \text{ and } B_g = B_i^d - G_i^d , \tag{2.25}$$

where R_i^d and B_i^d are, respectively, the initial estimated red and blue channels obtained from bilinear interpolation; and G_i^d is the initial estimated green channel obtained by any image interpolation method. The filtering procedure involves two sub-steps: first, edge-adaptive low-pass filtering of the red (blue) values over the original blue (red) pixels, as shown in Fig. 2-6(a); second, edge-adaptive low-pass filtering of the red (blue) values over the original green pixels, as shown in Fig. 2-6(b). Because the same procedure is used for both R_g and B_g color-difference planes, only the procedure of R_g will be described in the following presentation.

We first consider the red color-difference value at blue pixel locations. Referring to Fig. 2-6(a), the red color-difference value at blue pixel position, R_g^B, is to be filtered adaptively by



(a) (b)

Fig. 2-6: (a) The red color-difference value on blue pixel and (b) the red color-difference value on green pixel of a central pixel to be estimated.

$$R_g^B = \frac{e_{a1}\hat{R}_{g1} + e_{a2}\hat{R}_{g2} + e_{a3}\hat{R}_{g3} + e_{a4}\hat{R}_{g4}}{e_{a1} + e_{a2} + e_{a3} + e_{a4}}, \tag{2.26}$$

where $\hat{R}_{g1} \sim \hat{R}_{g4}$ are the red color-difference adjusted values and $e_{a1} \sim e_{a4}$ are the edge indicators corresponding to each color-difference value. These edge indicators are defined as a decreasing function of the directional derivative of the center point and its neighboring points. In the edge-adaptive low-pass filtering stage, we propose to introduce the second-order directional derivatives of neighboring color-difference values for detecting edges more accurately. In the case of Fig. 2-6(a), the edge indicators are given by

$$e_{a1} = \frac{1}{1 + \left|\frac{R_{g1} - R_{g3}}{2\sqrt{2}}\right| + \left|\frac{R_{g5} - 2R_{g1} + R_{g3}}{2\sqrt{2}}\right|}, \quad e_{a2} = \frac{1}{1 + \left|\frac{R_{g4} - R_{g2}}{2\sqrt{2}}\right| + \left|\frac{R_{g4} - 2R_{g2} + R_{g6}}{2\sqrt{2}}\right|},$$

$$e_{a3} = \frac{1}{1 + \left|\frac{R_{g1} - R_{g3}}{2\sqrt{2}}\right| + \left|\frac{R_{g1} - 2R_{g3} + R_{g7}}{2\sqrt{2}}\right|}, \quad e_{a4} = \frac{1}{1 + \left|\frac{R_{g4} - R_{g2}}{2\sqrt{2}}\right| + \left|\frac{R_{g8} - 2R_{g4} + R_{g2}}{2\sqrt{2}}\right|}. \tag{2.27}$$

The red color-difference adjusted values $\hat{R}_{g1} \sim \hat{R}_{g4}$ are derived based on the assumption that the difference of neighboring color-difference values along an interpolation direction is constant. For example, to find the red color-difference adjusted value at R_{g1} location, this assumption gives the following relationships for the neighboring red color-difference values along the right-up direction

$$R_{g1} - R_{g3} = (R_{g1} - \hat{R}_g) + (\hat{R}_g - R_{g3}) \text{ and} \tag{2.28}$$

$$R_{g1} - \hat{R}_g = \hat{R}_g - R_{g3} \tag{2.29}$$

where $\hat{R}_g$ denotes the missing red color-difference value at B_g location. Combining (2.28) and (2.29), we have

$$R_{g1} - R_{g3} = 2(R_{g1} - \hat{R}_g).$$

This implies that $\hat{R}_g = R_{g1} + (R_{g3} - R_{g1})/2$. This value is denoted by $\hat{R}_{g1}$, which is used to estimate R_g in the right-up interpolation direction. In a similar manner, the red color-difference adjusted values along each interpolation direction is given by

$$\begin{aligned} \hat{R}_{g1} &= R_{g1} + \frac{R_{g3} - R_{g1}}{2}, \quad \hat{R}_{g2} = R_{g2} + \frac{R_{g4} - R_{g2}}{2}, \\ \hat{R}_{g3} &= R_{g3} + \frac{R_{g1} - R_{g3}}{2}, \quad \hat{R}_{g4} = R_{g4} + \frac{R_{g2} - R_{g4}}{2}. \end{aligned} \tag{2.30}$$

Subsequently, the rest of the red color-difference value at green pixel locations is adaptively filtered from its four surrounding red color-difference values. In the case of Fig. 2-6(b), the red color-difference value at the central green pixel position, R_g^G, is adaptively filtered according to the hard-decision rule (2.20) such that

$$R_g^G = \begin{cases} \dfrac{e_{b2}\hat{R}_{g2} + e_{b4}\hat{R}_{g4}}{e_{b2} + e_{b4}}, \; if \; (x,y) \in \Omega_h; \\ \dfrac{e_{b1}\hat{R}_{g1} + e_{b3}\hat{R}_{g3}}{e_{b1} + e_{b4}}, \; if \; (x,y) \in \Omega_v; \\ \dfrac{e_{b1}\hat{R}_{g1} + e_{b2}\hat{R}_{g2} + e_{b3}\hat{R}_{g3} + e_{b4}\hat{R}_{g4}}{e_{b1} + e_{b2} + e_{b3} + e_{b4}}, \; if \; (x,y) \in \Omega_s. \end{cases} \tag{2.31}$$

where $\hat{R}_{g1} \sim \hat{R}_{g4}$ denote the red color-difference adjusted values defined in (2.30), and $e_{b1} \sim e_{b4}$ are the corresponding edge indicators given by

$$e_{b1} = \frac{1}{1 + \left| \frac{R_{g3} - R_{g1}}{2} \right| + \left| \frac{R_{g3} - 2R_{g1} + R_{g5}}{2} \right|}, \quad e_{b2} = \frac{1}{1 + \left| \frac{R_{g2} - R_{g4}}{2} \right| + \left| \frac{R_{g6} - 2R_{g2} + R_{g4}}{2} \right|},$$

$$e_{b3} = \frac{1}{1 + \left| \frac{R_{g3} - R_{g1}}{2} \right| + \left| \frac{R_{g7} - 2R_{g3} + R_{g1}}{2} \right|}, \quad e_{b4} = \frac{1}{1 + \left| \frac{R_{g2} - R_{g4}}{2} \right| + \left| \frac{R_{g2} - 2R_{g4} + R_{g8}}{2} \right|}. \tag{2.32}$$

Finally, the full-red plane is obtained by recovering the spatial plane from red color-difference plane such that

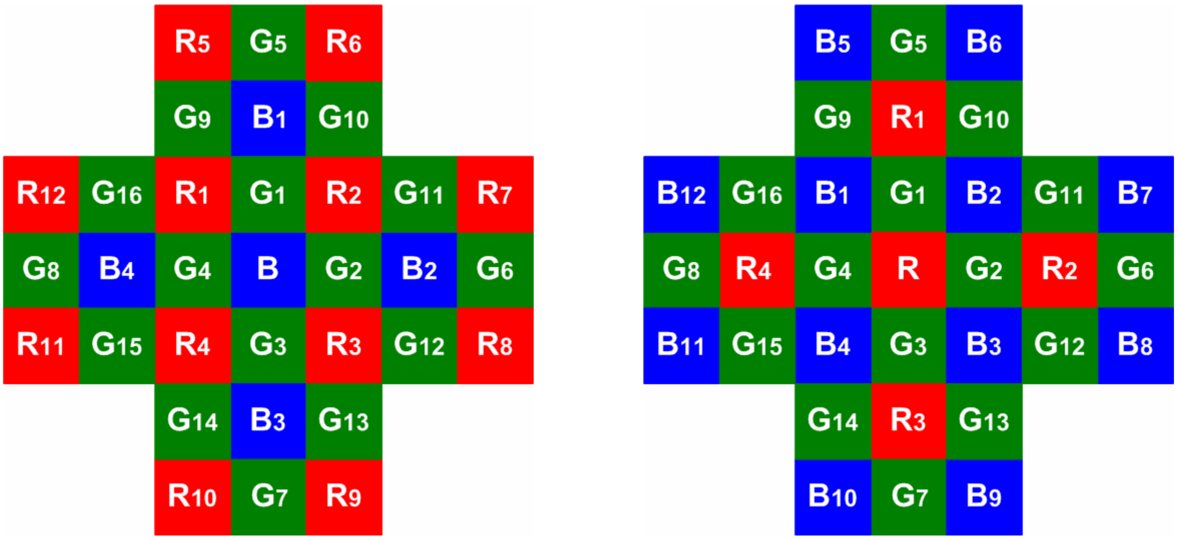
$$\overline{R}_i^d = R_g + G_i^d. \tag{2.33}$$

As the same procedure is utilized for recovering the blue plane, a full-color demosaiced image can be obtained.

2.3.3 Green Channel Edge-Adaptive Interpolation

In this section, we present a novel edge-adaptive interpolation method for green channel reconstruction from CFA samples. The green plane has the most spatial information of the image to be demosaiced and has great influence on the perceptual quality of the image. In order to reconstruct the demosaiced images with satisfactory quality, the hard-decision rule (2.20) is utilized for choosing the direction of interpolation to reconstruct the green channel. Figure 2-7 shows two cases of green samples in Bayer pattern, where the green value of central pixels are to be estimated from its four surrounding green pixels, $G_1 \sim G_4$. The central missing green value G_{miss} is estimated by the following expression

$$G_{miss} = \begin{cases} \dfrac{e_{e2}\hat{G}_2 + e_{e4}\hat{G}_4}{e_{e2} + e_{e4}}, \; if \; (x,y) \in \Omega_h; \\ \dfrac{e_{e1}\hat{G}_1 + e_{e3}\hat{G}_3}{e_{e1} + e_{e3}}, \; if \; (x,y) \in \Omega_v; \\ \dfrac{e_{s1}\hat{G}_1 + e_{s2}\hat{G}_2 + e_{s3}\hat{G}_3 + e_{s4}\hat{G}_4}{e_{s1} + e_{s2} + e_{s3} + e_{s4}}, \; if \; (x,y) \in \Omega_s. \end{cases} \tag{2.34}$$



(a) (b)

Fig. 2-7: Two cases of missing green value on the central pixel. (a) The green value on blue pixel and (b) the green value on red pixel of a central pixel to be estimated.

where $\hat{G}_1 \sim \hat{G}_4$ denote the color-adjusted green values of four surrounding green pixels; $e_{s1} \sim e_{s4}$ are the corresponding edge indicators when the location of central missing pixel (*x*, *y*) locates in the smooth subset Ω_s; and $e_{el} \sim e_{e4}$ are the corresponding edge indicators when the location of central missing pixel (x, y) locates in the horizontal subset Ω_h or vertical subset Ω_v. In other words, if $(x, y) \in \Omega_v$, we regard the central missing pixel as being in a smooth region. Otherwise, we regard the central pixel as being in an edge region. In the smooth region, the edge indicators associated with four surrounding color-adjusted green values are denoted by $e_{s1} \sim e_{s4}$ and the central missing pixel is then estimated by weighted sum of them. In the edge region, the edge indicators associated with four surrounding color-adjusted green values are denoted by $e_{e1} \sim e_{e4}$ and the central missing pixel is then carried out by selecting weighted sum in horizontal and vertical directions.

Based on (2.34), the color-adjusted green values and the corresponding edge indicators need to be determined for estimating the central missing green value G_{miss}. For instance, in the case of Fig. 2-7(a), the color-adjusted values in each interpolation direction are referred to

[3] and given by

$$\hat{G}_1 = G_1 + \frac{B - B_1}{2},\ \ \hat{G}_2 = G_2 + \frac{B - B_2}{2},\ \ \hat{G}_3 = G_3 + \frac{B - B_3}{2},\ \ \hat{G}_4 = G_4 + \frac{B - B_4}{2}. \quad (2.35)$$

Similarly, in Fig. 2-7(b), the color-adjusted values in each interpolation direction are given by

$$\hat{G}_1 = G_1 + \frac{R - R_1}{2},\ \ \hat{G}_2 = G_2 + \frac{R - R_2}{2},\ \ \hat{G}_3 = G_3 + \frac{R - R_3}{2},\ \ \hat{G}_4 = G_4 + \frac{R - R_4}{2}. \quad (2.36)$$

The calculation of edge indicators associated with four surrounding green pixels is divided into two cases. In smooth regions, a valid assumption is that the directional derivatives of each color channel are small. Thus, the edge indicates in smooth regions can be seen as a decreasing function dependent on the sum of local first-order directional derivative such that

$$e_{s1} = \frac{1}{1 + |G_1 - G_3| + |G_5 - G_1| + |B_1 - B| + \left|\frac{G_9 - G_4 + G_{10} - G_2}{2}\right| + \left|\frac{R_5 - R_1 + R_6 - R_2}{2}\right|}$$

$$e_{s2} = \frac{1}{1 + |G_2 - G_4| + |G_6 - G_2| + |B_2 - B| + \left|\frac{G_{11} - G_1 + G_{12} - G_3}{2}\right| + \left|\frac{R_7 - R_2 + R_8 - R_3}{2}\right|}$$

$$e_{s3} = \frac{1}{1 + |G_1 - G_3| + |G_3 - G_7| + |B - B_3| + \left|\frac{G_2 - G_{13} + G_4 - G_{14}}{2}\right| + \left|\frac{R_3 - R_9 + R_4 - R_{10}}{2}\right|}$$

$$e_{s4} = \frac{1}{1 + |G_2 - G_4| + |G_4 - G_8| + |B - B_4| + \left|\frac{G_1 - G_{16} + G_3 - G_{15}}{2}\right| + \left|\frac{R_4 - R_{11} + R_1 - R_{12}}{2}\right|}. \quad (2.37)$$

In edge regions, the assumption is that the directional derivatives of each color channel are consistent along the direction of edges. Thus, the edge indicates in edge regions can be seen as a decreasing function dependent on the consistence of local first-order directional derivative such that

$$e_{e1} = \frac{1}{1 + |G_1 - G_3| + |G_5 - 2G_1 + G_3| + |B_1 - B - G_1 + G_3| + |R_5 - R_1 - G_9 + G_4| + |R_6 - R_2 - G_{10} + G_2|}$$

$$e_{e2} = \frac{1}{1 + |G_2 - G_4| + |G_6 - 2G_2 + G_4| + |B_2 - B - G_2 + G_4| + |R_7 - R_2 - G_{11} + G_1| + |R_8 - R_3 - G_{12} + G_3|}$$

$$e_{e3} = \frac{1}{1 + |G_1 - G_3| + |G_1 - 2G_3 + G_7| + |G_1 - G_3 - B + B_3| + |G_2 - G_{13} - R_3 + R_9| + |G_4 - G_{14} - R_4 + R_{10}|}$$

$$e_{e4} = \frac{1}{1+|G_2 - G_4| + |G_2 - 2G_4 + G_8| + |G_2 - G_4 - B + B_4| + |G_1 - G_{16} - R_1 + R_{12}| + |G_3 - G_{15} - R_4 + R_{11}|}. \tag{2.38}$$

Once the color-adjusted green values and the corresponding edge indicators are obtained from (2.35) to (2.38), the missing green value G can be obtained by using (2.34). Finally, the full green channel can be obtained by adopting the procedure as described above to interpolate all missing green values on red and blue pixel positions. This method for interpolating green channel from CFA samples is combined with the proposed CDEA CFA interpolation algorithm.

Remark 2.2: Although edge-adaptive interpolation can provide more pleasing results, it also increases the computational load and the amount of memory transactions compared with a linear interpolation [39]. In order to reduce the computational cost in CFA interpolation step, we can still use a linear interpolation instead of the edge-adaptive interpolation. More specifically, for linear interpolation, the edge indicators $e_{a1} \sim e_{a4}$ in (2.26) and $e_{s1} \sim e_{s4}$ in (2.34) are fixed such that

$$(e_{a1}, e_{a2}, e_{a3}, e_{a4}) = (1,1,1,1), \ \ (e_{s1}, e_{s2}, e_{s3}, e_{s4}) = (1,1,1,1). \tag{2.39}$$

And the edge indicators $e_{b1} \sim e_{b4}$ in (2.31) and $e_{e1} \sim e_{e4}$ in (2.34) are simplified such that

$$(e_{b1}, e_{b2}, e_{b3}, e_{b4}) = \begin{cases} (0,1,0,1), \ \textit{if} \ \ (x, y) \in \Omega_h \\ (1,0,1,0), \ \textit{if} \ \ (x, y) \in \Omega_v \\ (1,1,1,1), \ \textit{if} \ \ (x, y) \in \Omega_s \end{cases}, \ \ (e_{e1}, e_{e2}, e_{e3}, e_{e4}) = \begin{cases} (0,1,0,1), \ \textit{if} \ \ (x, y) \in \Omega_h \\ (1,0,1,0), \ \textit{if} \ \ (x, y) \in \Omega_v \end{cases} \tag{2.40}$$

The advantage of linear interpolation is that it not only can skip the calculation of edge indicators, but also use bit-shift instead of division to reduce the computation time. Therefore, compared with edge-adaptive interpolation, the computation cost of linear interpolation will be greatly reduced.

2.4 The Complete HPHD-CDEA CFA interpolation Algorithm

We summarize the complete HPHD-CDEA CFA interpolation algorithm as follows.

1) Initialization: Set window size N to calculate the heterogeneity vector $P_{N\times1}$ by $T^{1}_{N\times(N-3)}$ and $T^{2}_{(N-3)\times1}$ defined in (10) and (11), respectively; set parameter α for spatial classification.

2) Decision Stage (HPHD algorithm):

a) Heterogeneity-projection: Calculate the horizontal and vertical heterogeneity maps, H_{h_map} and H_{v_map}, from original Bayer mosaic image by (2.14).

b) Directional adaptive filtering: Filter the horizontal and vertical heterogeneity maps by directional adaptive filters (2.17) and (2.18), respectively.

c) Spatial classification: Use parameter α and the two filtered heterogeneity maps to classify the image into three subsets Ω_h, Ω_v, and Ω_s by (2.10).

3) Interpolation Stage (CDEA CFA interpolation):

a) Interpolate G channel at R and B pixels by interpolation rule (2.34).

b) Interpolate R and B channels by bilinear interpolation; then calculate the red and blue color-difference planes, R_g and B_g, by (2.25).

c) Filter R_g value at B pixels by adaptive filtering rule (2.26) and the B channel similarly.

d) Filter R_g value at B pixels by adaptive filtering rule (2.31) and the B channel similarly.

Fig. 2-8 illustrates the flowchart of the proposed HPHD-CDEA CFA interpolation algorithm. The main difference between the proposed algorithm and the existent decision-based schemes is that the decision stage is performed before the interpolation stage in this design, thanks to the heterogeneity-projection. This advantage contributes not only to improving the quality of demosaicing result, but also to reducing the computational complexity of the decision stage. In Chapter 5, a comparative study of experimental results and analysis of computational complexity will be discussed to demonstrate the performance of the proposed method.

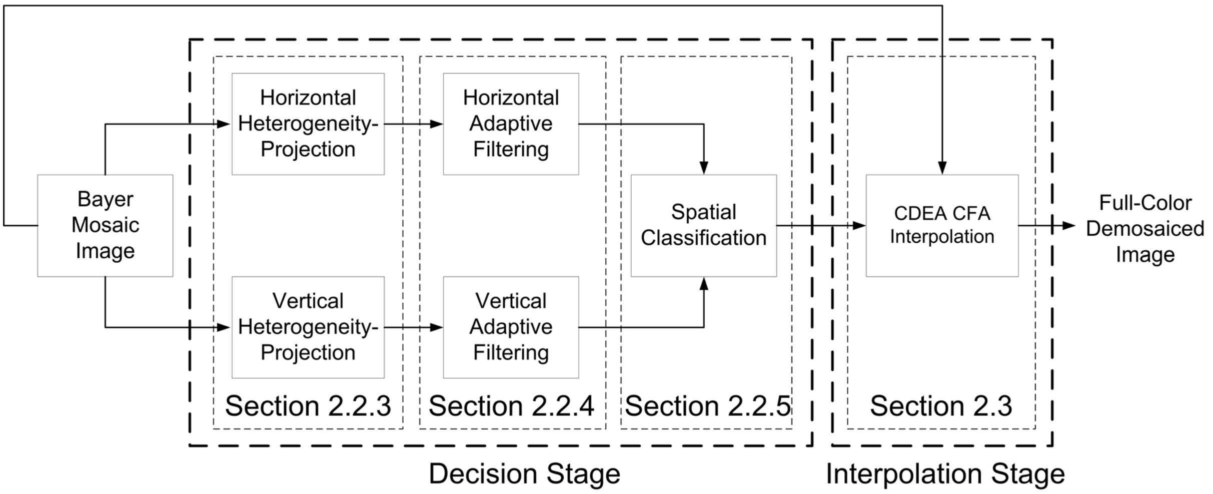

Bayer Mosaic Image
Horizontal Heterogeneity-Projection
Vertical Heterogeneity-Projection
Horizontal Adaptive Filtering
Vertical Adaptive Filtering
Spatial Classification
CDEA CFA Interpolation
Full-Color Demosaiced Image
Section 2.2.3
Section 2.2.4
Section 2.2.5
Section 2.3
Decision Stage
Interpolation Stage

Fig. 2-8: Flowchart of the proposed HPHD-CDEA CFA interpolation algorithm.

2.4.1 Example Study

Figure 2-9 illustrates the execution steps of the proposed algorithm by using an example. The Kodak small Lighthouse image (384x256) is downsampled into a Bayer mosaiced image as shown in Fig. 2-9(a). In this picture, the fence regions usually challenge the performance of a demosaicing procedure. Figures 2-9(b) and 2-9(c), respectively, are the horizontal heterogeneity map H_{h_map} and vertical heterogeneity map H_{v_map} obtained from (2.14) discussed in Section 2.2.3 ($N = 24$ in this example). Through the directional adaptive filtering discussed in Section 2.2.4, the filtered horizontal heterogeneity map $H^*_{h\_map}$ and filtered vertical heterogeneity map $H^*_{v\_map}$ are obtained in Figs. 2-9(d)-(e), respectively. Comparing Figs. 2-9(d)-(e) with Figs. 2-9(b)-(c), one can see that the unwanted noises in two original heterogeneity maps have been removed effectively by using the directional adaptive filters. Employing two filtered heterogeneity maps, the horizontal, vertical, and smooth subsets in the image are obtained directly by (2.19) with $\alpha = 0.8$. Figure 2-9(f) shows three decided subsets in the image, where the gray region is the horizontal subset Ω_h, the white region is the vertical subset Ω_v, and the black region is the smooth subset Ω_s. Note that Fig. 2-9(f) shows the decisions in fence regions are almost all vertical. The interpolations are thus

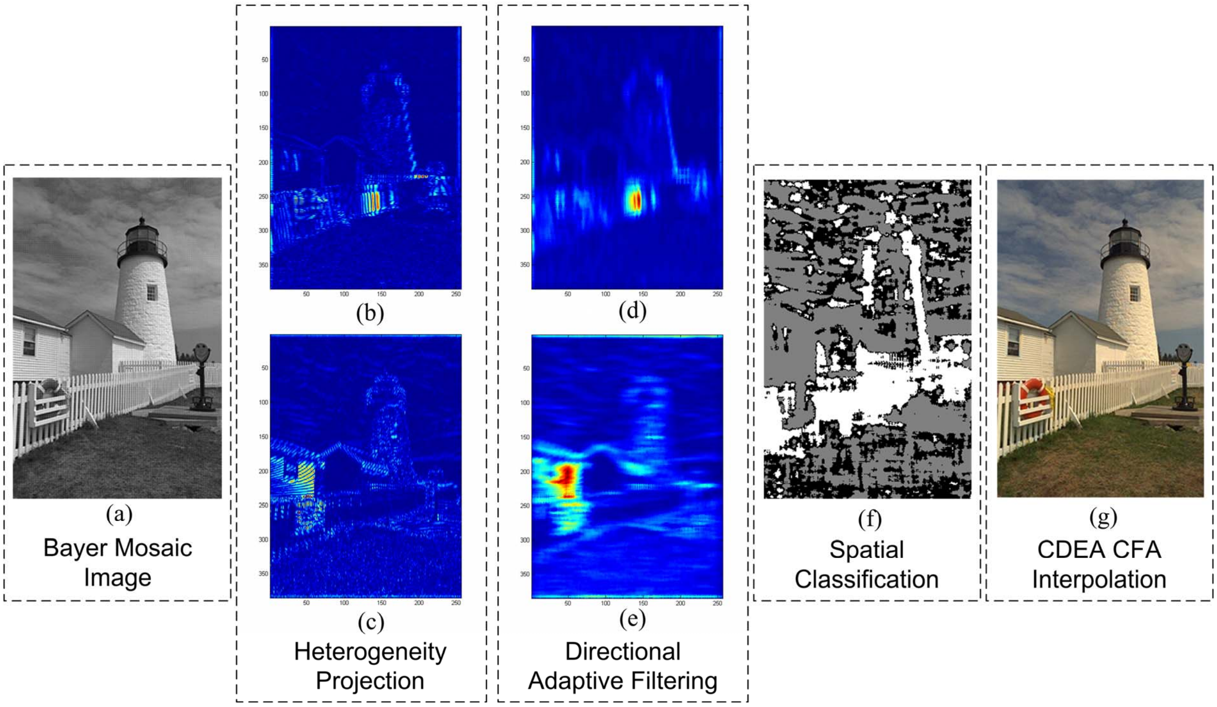


Fig. 2-9: Illustration of execution steps of the proposed HPHD-CDEA CFA interpolation algorithm. (a) Original Bayer mosaic image of small Lighthouse image (384x256). (b) Horizontal heterogeneity map H_{h_map} ($N = 24$). (c) Vertical heterogeneity map H_{v_map}. (d) Filtered horizontal heterogeneity map $H^*_{h\_map}$. (e) Filtered vertical heterogeneity map $H^*_{v\_map}$. (f) Three decided subsets in the image ($\alpha = 0.8$). The gray region is the horizontal subset Ω_h, the white region is the vertical subset Ω_v, and the black region is the smooth subset Ω_s. (g) Interpolation result using the proposed CDEA CFA interpolation presented in Section 2.3.

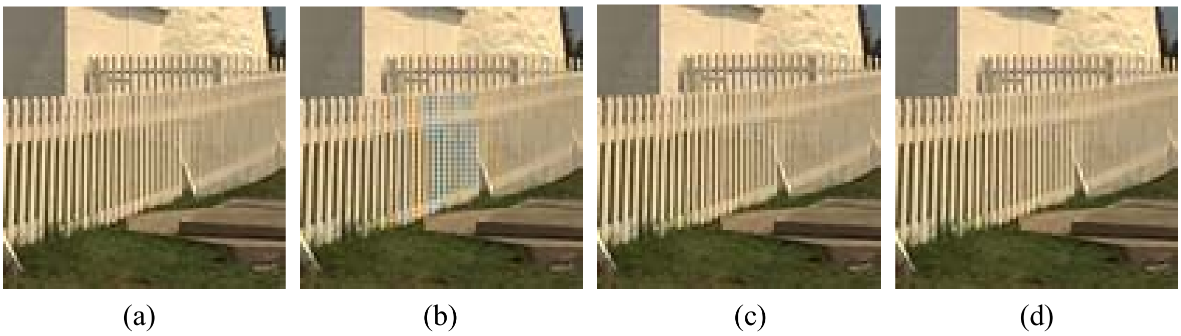


Fig. 2-10: (a) Zoom-in of the original Lighthouse image in the fence region. Zoom-in of the demosaicing results with parameters $N = 24$ and (b) $\alpha = 0$, (c) $\alpha = 0.5$, (d) $\alpha = 0.8$.

along the correct directions. Finally, the proposed CDEA CFA interpolation discussed in Section 2.3 was applied to reconstruct the color image based on these three decided subsets.

Figure 2-9(g) illustrates the interpolation results. In Fig. 2-9(g), one can see that the fine details of interpolation such as the fence and house regions are reconstructed successfully.

To further illustrate the performance, we tweak parameter α and compare the demosaicing results with original image ($N = 24$ is fixed). Figure 2-10(a) is the zoom-in of the original fence regions. Figure 2-10(b) is the zoom-in of demosaicing result with parameter $\alpha = 0$. One can see that the demosaiced image contains many color artifacts due to the inaccurate smooth interpolation. Figures 2-10(c) and 2-10(d) show the demosaicing results with parameter $\alpha = 0.5$ and $\alpha = 0.8$, respectively. It is clear that the proposed HPHD-CDEA CFA interpolation method reduces the color artifacts efficiently. Visually compare Fig. 2-10(d) with Fig. 2-10(a), one can see that most detail features have been reconstructed correctly.

2.5 Summary

A novel hard-decision CFA interpolation procedure has been developed based on the spectral-spatial correlation of a Bayer mosaic image. The proposed HPHD CFA interpolation method effectively reconstructs fine detail features in both edge and texture regions of demosaiced images. One merit of the proposed HPHD algorithm is that it can combine with many existing image interpolation methods such as decision-based algorithm (set $\alpha = 1$), edge-directed interpolation, adaptive interpolation, linear interpolation, etc for improved performance. Moreover, the proposed heterogeneity-projection scheme provides an efficient method for decision-based algorithms to make accurate direction-selection before performing interpolation.

In order to reconstruct the demosaiced images with fewer color artifacts, a novel CDEA CFA interpolation method is then proposed to combine with the HPHD algorithm. The proposed CDEA algorithm effectively reduces color artifacts in both smooth and edge regions of demosaiced images. Furthermore, any existing image interpolation method can be combined with the proposed CDEA algorithm to reconstruct the green channel. A new edge-adaptive interpolation method is also presented by adopting the hard-decision rule from HPHD algorithm to reconstruct the green channel from CFA samples. In Chapter 5, the performance of the proposed HPHD-CDEA method will be compared with three renowned CFA interpolation methods. Experimental results will show that HPHD-CDEA method not only outperforms all of them in PSNR (dB) and S-CIELAB ΔE_{ab}^{*} measures, but also gives superior demosaiced fidelities in visual comparison.

Chapter 3

Robust Mobile Robot Visual Tracking Control Based on a Dual-Jacobian Visual Interaction Model

3.1 Introduction

From the literature discussed in Chapter 1, we have noted that a challenge in mobile robot visual tracking control design is to develop a visual tracking control system to track a dynamic moving target based on a stability criterion and overcome the uncertainties encountered in practical systems. This problem motivates us to derive a new model for designing a robust VTC to solve the visual tracking problem of dynamic moving target and overcome the internal disturbances of practical system (such as system parametric uncertainty and velocity quantization uncertainty). To achieve this, this chapter presents a novel dual-Jacobian visual interaction model to help the design of a robust VTC for a wheeled mobile robot equipped with a tilt camera. The proposed design enhances various image-tracking applications using an on-board monocular camera, such as human-robot interaction and surveillance. Based on Lyapunov theory, the proposed control scheme not only possesses some degree of robustness against parametric uncertainty, but also overcomes the external uncertainty caused by velocity quantization noise. Moreover, the proposed controller fully works in image space; hence the computational complexity and the effects of sensor/camera modeling errors can be greatly reduced. The main differences between the proposed VTC and other existent approaches are summarized as follows:

1) The proposed dual-Jacobian visual interaction model considers not only the effect of

mobile robot motion, but also the effect of target motion. Thus, based on the proposed model, the visual tracking control problem of a unicycle-modeled mobile robot for tracking a dynamic moving target can be solved with exponential convergence using a single controller. Moreover, the proposed model also considers the kinematics of a tilt camera platform mounted on the mobile robot. Therefore, the applicability of the proposed method is greatly increased.

2) The proposed visual tracking control system not only possesses some degree of robustness against the system model uncertainties, but also overcomes the unmodelled quantization effect in the velocity commands and the occlusion effect during visual tracking process. This advantage enhances the reliability of the proposed method in practical applications.

3) The proposed visual tracking control system works fully in image space. Therefore, compared with position-based [23], homography-based [13, 14], and epipole-based [4, 15, 16, 17] visual tracking control approaches, the computational complexity and the sensor/camera modeling errors can be reduced due to the advantages of image-based visual servo control [2].

The basic assumptions of the proposed method are listed as follows:

1) The on-board camera is supposed to be a calibrated pinhole camera. Because the proposed VTC possesses some degree of robustness against parametric uncertainty, a simple linear camera calibration method [44] can be used to estimate the intrinsic parameters of the camera.

2) The width of target is supposed to be *a priori* known constant in order to simplify the depth estimation problem. However, this is not a necessary assumption for the proposed method. Any algorithm or sensor which provides the depth information can be utilized to combine with the proposed method.

The rest of this chapter is organized as follows. Section 3.2 describes the system modeling of visual tracking control problem and the proposed dual-Jacobian visual interaction model accordingly. Section 3.3 presents the results of VTC design. In Section 3.4, the robustness of the control system against the system model uncertainty is analyzed and discussed. Section 3.5 presents the design of the robust control law to overcome the velocity quantization error encountered in practical systems. Section 3.6 summarizes the contributions of this design. Experimental results will be reported in Chapter 5. Several interesting experimental observations will be presented and discussed.

3.2 Camera-Object Visual Interaction Model

This section derives the visual interaction model between a mobile robot and a dynamic moving target. We first introduce the kinematics model of wheeled mobile robot and target used in this design. The mathematic derivations of the proposed model are then presented and explained.

3.2.1 Kinematics Model of Wheeled Mobile Robot and Target

Figure 3-1 shows the model of wheeled mobile robot and target considered in the nonholonomic visual tracking control problem. The wheeled mobile robot equips with a tilt camera to track a dynamic motion target, which is supposed to be a well-recognizable object with appropriate dimensions in the image plane and can only translate with respect to the robot. The tilt camera is mounted on top of the mobile robot and its optical-axis faces the target of interest, for instance, a human face. Figure 3-1(a) illustrates a model of the wheeled mobile robot and the target in the world coordinate frame F_f (see Fig. 3-2), in which the motion of the target is supposed to be holonomic such that

$$\dot{X}_f^t = V_f^t, \tag{3.1}$$

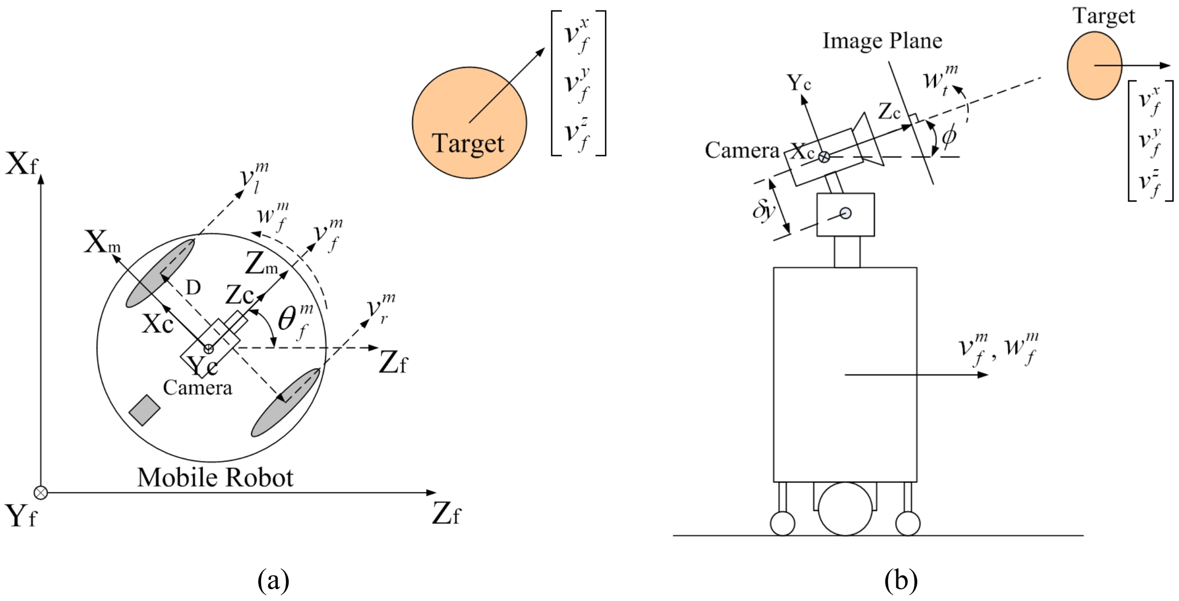


Fig. 3-1: (a) A model of the wheeled mobile robot and the target in the world coordinate frame. (b) Side view of the wheeled mobile robot with a tilt camera mounted on top of it.

where $X_f^t = [x_f^t \quad y_f^t \quad z_f^t]^T$ and $V_f^t = [v_f^x \quad v_f^y \quad v_f^z]^T$ denote, respectively, the position and velocity of target in the world coordinates.

Figure 3-1(b) is the side view of the scenario under consideration, in which the tilt angle ϕ gives the relationship between the camera coordinate frame F_c and the mobile coordinate frame F_m. The kinematics of the wheeled mobile robot is described by [45]

$$\dot{X}_f^m = \begin{bmatrix} v_f^m \sin\theta_f^m \\ 0 \\ v_f^m \cos\theta_f^m \end{bmatrix}, \text{and} \quad \begin{array}{l} \dot{\theta}_f^m = w_f^m \\ \dot{\phi} = w_t^m \end{array}, \tag{3.2}$$

where $X_f^m = [x_f^m \quad y_f^m \quad z_f^m]^T$ is the position of mobile robot in the world coordinates, (θ_f^m, ϕ) are the orientation angle of mobile robot and the tilt angle of onboard camera, w_t^m is the tilt velocity of the camera, and (v_f^m, w_f^m) are the linear and angular velocities of mobile robot. In practice, (v_f^m, w_f^m) can be used to calculate the velocity of each wheel of the mobile robot such that

$$v_l^m = v_f^m - (\mathrm{D}\cdot w_f^m)\big/2 \;\text{ and }\; v_r^m = v_f^m + (\mathrm{D}\cdot w_f^m)\big/2, \tag{3.3}$$

where (v_l^m, v_r^m) are the left- and right-wheel velocities, respectively, and D represents the distance between the two drive wheels. In the rest of this chapter, the target model (3.1) and mobile robot model (3.2) will be utilized to derive the visual interaction model and to design the visual tracking control system.

3.2.2 Coordinate Transformation from World Frame into Camera Frame

Figure 3-2 illustrates the relationship between coordinate systems, namely the world, camera and image coordinate frames. Let $X_f = X_f^t - X_f^m$ denote the related position between mobile robot and the target in the world coordinate frame. In order to describe a mobile robot interacting with the target in the image coordinate frame, a visual interaction model has been derived by transferring the kinematics of X_f from the world coordinate frame into the image coordinate frame. This subsection presents the transformation of the kinematics of X_f from the world coordinate frame into the camera coordinate frame.

As shown in Fig. 3-2, $X_c = [x_c \quad y_c \quad z_c]^T$ denotes the related position in the camera coordinate frame and can be calculated by the coordinate transformation such that

$$X_c = \mathbf{R}(\phi, \theta_f^m) X_f - \delta\mathrm{Y}, \tag{3.4}$$

where

$$\mathbf{R}(\phi,\theta_f^m) = \mathbf{R}(\phi)\mathbf{R}(\theta_f^m) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta_f^m & 0 & -\sin\theta_f^m \\ 0 & 1 & 0 \\ \sin\theta_f^m & 0 & \cos\theta_f^m \end{bmatrix}, \quad \delta\mathrm{Y} = \begin{bmatrix} 0 & \delta y & 0 \end{bmatrix}^T.$$

δy is the distance between the center of robot head and the onboard camera. Because $\delta\mathrm{Y}$ is a constant translation vector, the derivative of (3.4) becomes

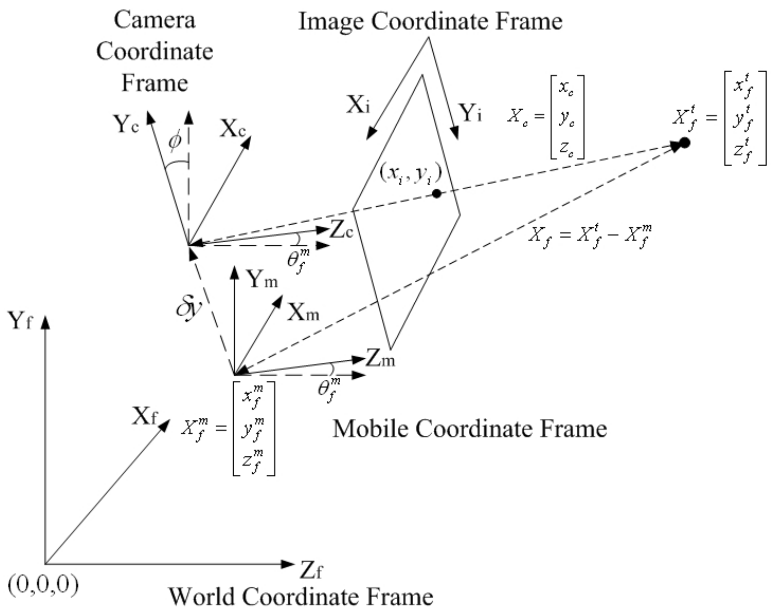


Fig. 3-2: World, camera and image coordinate frames of robotic visual interaction.

$$\dot{X}_c = \left[\frac{\partial \mathbf{R}(\phi, \theta_f^m)}{\partial \phi} \dot{\phi} + \frac{\partial \mathbf{R}(\phi, \theta_f^m)}{\partial \theta_f^m} \dot{\theta}_f^m \right] X_f + \mathbf{R}(\phi, \theta_f^m) \dot{X}_f , \tag{3.5}$$

where $\frac{\partial \mathbf{R}(\phi,\theta_f^m)}{\partial \phi} = \begin{bmatrix} 0 & 0 & 0 \\ -\cos\phi \sin\theta_f^m & -\sin\phi & -\cos\phi\cos\theta_f^m \\ -\sin\phi\sin\theta_f^m & \cos\phi & -\sin\phi\cos\theta_f^m \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{R}(\phi,\theta_f^m) \equiv \mathbf{\Psi}_1 \mathbf{R}(\phi,\theta_f^m)$,

$$\frac{\partial \mathbf{R}(\phi,\theta_f^m)}{\partial \theta_f^m} = \begin{bmatrix} -\sin\theta_f^m & 0 & -\cos\theta_f^m \\ -\sin\phi\cos\theta_f^m & 0 & \sin\phi\sin\theta_f^m \\ \cos\phi\cos\theta_f^m & 0 & -\cos\phi\sin\theta_f^m \end{bmatrix} = \begin{bmatrix} 0 & \sin\phi & -\cos\phi \\ -\sin\phi & 0 & 0 \\ \cos\phi & 0 & 0 \end{bmatrix} \mathbf{R}(\phi,\theta_f^m) \equiv \mathbf{\Psi}_\phi \mathbf{R}(\phi,\theta_f^m) .$$

Substituting (3.1), (3.2) and (3.4) into (3.5), the kinematics of the interaction between robot and target in camera frame can be obtained by taking the derivative of $\boldsymbol{X}_c$ such that

$$\begin{aligned} \dot{\boldsymbol{X}}_c &= (\mathbf{\Psi}_1 w_t^m + \mathbf{\Psi}_\phi w_f^m)\mathbf{R}(\phi,\theta_f^m)(\boldsymbol{X}_f^t - \boldsymbol{X}_f^m) + \mathbf{R}(\phi)\mathbf{R}(\theta_f^m)(\dot{\boldsymbol{X}}_f^t - \dot{\boldsymbol{X}}_f^m) \\ &= (\mathbf{\Psi}_1 w_t^m + \mathbf{\Psi}_\phi w_f^m)(\boldsymbol{X}_c + \delta Y) + \mathbf{R}(\phi)\left[\mathbf{R}(\theta_f^m)V_f^t + \bar{e}_3 v_f^m\right] \\ &= (\mathbf{\Psi}_1 w_t^m + \mathbf{\Psi}_\phi w_f^m)\boldsymbol{X}_c + [\mathbf{R}(\phi)\bar{e}_3 \quad \mathbf{\Psi}_\phi \delta Y \quad \mathbf{\Psi}_1 \delta Y]u + \mathbf{R}(\phi,\theta_f^m)V_f^t \\ &\equiv \mathbf{A}_c \boldsymbol{X}_c + \mathbf{B}_c u + \mathbf{R}(\phi,\theta_f^m)V_f^t , \end{aligned} \tag{3.6}$$

where

$$\mathbf{A}_c = \mathbf{\Psi}_1 w_t^m + \mathbf{\Psi}_\phi w_f^m = \begin{bmatrix} 0 & w_f^m \sin\phi & -w_f^m \cos\phi \\ -w_f^m \sin\phi & 0 & -w_t^m \\ w_f^m \cos\phi & w_t^m & 0 \end{bmatrix}, \quad \bar{e}_3 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \text{ and}$$

$$\mathbf{B}_c = \begin{bmatrix} \mathbf{R}(\phi)\bar{e}_3 & \mathbf{\Psi}_\phi \delta Y & \mathbf{\Psi}_1 \delta Y \end{bmatrix} = \begin{bmatrix} 0 & \delta y \sin\phi & 0 \\ \sin\phi & 0 & 0 \\ -\cos\phi & 0 & \delta y \end{bmatrix}.$$

$u = [v_f^m \quad w_f^m \quad w_t^m]^T$ is the control velocity of the mobile robot and on-board tilt camera. In the following, the kinematics model (3.6) will be used to derive the interaction model in the image coordinate frame.

3.2.3 Coordinate Transformation from Camera Frame into Image Frame

In this subsection, the related position X_c is transformed into the image coordinate frame for deriving the visual interaction model based on (3.6). We first define the system state in the image coordinate frame for the controller design. Fig. 3-3 illustrates the definition of observed system state in the image plane. In Fig. 3-3, x_i and y_i are, respectively, the horizontal and vertical position of the centroid of target in the image plane, and d_x is the width of target in the image plane. Similar to the human's visual tracking behavior, the purpose of the visual tracking control design is to control the centroid position and width of target from an initial state into the desired state in the image plane.

In the following, the visual interaction model is derived by (3.6) and the selected system state. Based on the pinhole camera model, the diffeomorphism (please see [46] for the specification) in the image plane can be defined by the standard projection equations [44] such that:

$$X_i = \begin{bmatrix} x_i & y_i & d_x \end{bmatrix}^T = \begin{bmatrix} -k_x x_c & k_y y_c & k_x k_w z_c \end{bmatrix}^T, \; k_x = \mathrm{f}_x / z_c, \; k_y = \mathrm{f}_y / z_c, \; k_w = \mathrm{W} / z_c, \tag{3.7}$$

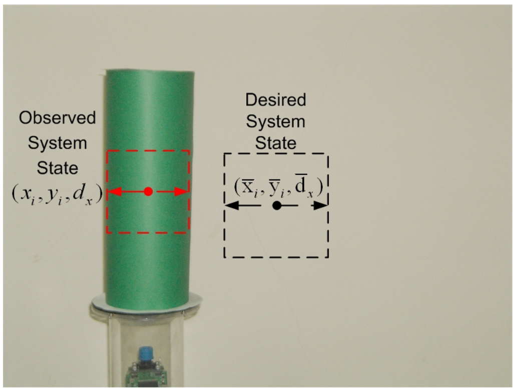


Fig. 3-3: The definition of observed and desired system states in the image plane.

where $(\mathrm{f}_x, \mathrm{f}_y)$ represent fixed focal length along the image *x*-axis and *y*-axis, respectively [47], and W denotes the actual width of the target. By taking the derivative of (3.7), the kinematic relationship between image and camera coordinate frames can be found such that

$$\dot{X}_i = P_i \dot{X}_c,\ \ P_i = \begin{bmatrix} -k_x & 0 & -k_x \mathrm{f}_x^{-1} x_i \\ 0 & k_y & -k_y \mathrm{f}_y^{-1} y_i \\ 0 & 0 & -k_x \mathrm{f}_x^{-1} d_x \end{bmatrix}. \tag{3.8}$$

Substituting (3.6) and (3.7) into (3.8), the kinematic relationship between robot and target in the image coordinate frame can be modeled by quasi-linear parameter-varying (Quasi-LPV) description [48] such that

$$\dot{X}_i = \mathbf{A}_i X_i + \mathbf{B}_i u + C_i \tag{3.9}$$

where $\mathbf{A}_i = diag(A_1, A_2, A_1)$,

$$A_1 = -\frac{k_x}{\mathrm{f}_x}(v_f^x \cos\phi \sin\theta_f^m + v_f^y \sin\phi + v_f^z \cos\phi \cos\theta_f^m),$$

$$A_2 = -\frac{k_y}{\mathrm{f}_y}(v_f^x \cos\phi \sin\theta_f^m + v_f^y \sin\phi + v_f^z \cos\phi \cos\theta_f^m),$$

$$C_i = \begin{bmatrix} k_x(v_f^z \sin\theta_f^m - v_f^x \cos\theta_f^m) \\ k_y(v_f^y \cos\phi - v_f^x \sin\phi \sin\theta_f^m - v_f^z \sin\phi \cos\theta_f^m) \\ 0 \end{bmatrix},$$

$$\mathbf{B}_i = \begin{bmatrix} \frac{k_x}{\mathrm{f}_x} x_i \cos\phi & (\frac{x_i^2 + \mathrm{f}_x^2}{\mathrm{f}_x})\cos\phi - \frac{\mathrm{f}_x}{\mathrm{f}_y}(k_y \delta y + y_i)\sin\phi & -\frac{x_i(k_y \delta y + y_i)}{\mathrm{f}_y} \\ k_y(\sin\phi + \frac{y_i}{\mathrm{f}_y}\cos\phi) & \frac{\mathrm{f}_y}{\mathrm{f}_x} x_i(\sin\phi + \frac{y_i}{\mathrm{f}_y}\cos\phi) & -\frac{y_i^2 + \mathrm{f}_y^2 + k_y y_i \delta y}{\mathrm{f}_y} \\ \frac{k_x}{\mathrm{f}_x} d_i \cos\phi & \frac{x_i d_x}{\mathrm{f}_x}\cos\phi & -\frac{d_x(k_y \delta y + y_i)}{\mathrm{f}_y} \end{bmatrix},$$

where $diag(a,b,c)$ denotes a 3-by-3 diagonal matrix with diagonal element *a*, *b*, and *c*; the elements of system matrix $\mathbf{A}_i$ and vector C_i are time-varying function dependent on the robot's pose and target's velocity; and the elements of control matrix $\mathbf{B}_i$ are time-varying function dependent on the robot's pose and current system state.

3.2.4 Dual-Jacobian Visual Interaction Model

The visual interaction model (3.9) indicates that the elements of system matrix $\mathbf{A}_i$ and vector C_i are function of target velocity. Thus, expression (3.9) can be rewritten such that

$$\dot{X}_i = (\mathbf{A}_i X_i + C_i) + \mathbf{B}_i u = \mathbf{J}_i V_f^t + \mathbf{B}_i u\,, \tag{3.10}$$

where

$$\mathbf{J}_i = \begin{bmatrix} -k_x(\frac{x_i}{\mathrm{f}_x}\cos\phi\sin\theta_f^m + \cos\theta_f^m) & -k_x\frac{x_i}{\mathrm{f}_x}\sin\phi & -k_x(\frac{x_i}{\mathrm{f}_x}\cos\phi\cos\theta_f^m - \sin\theta_f^m) \\ -k_y(\frac{y_i}{\mathrm{f}_y}\cos\phi\sin\theta_f^m + \sin\phi\sin\theta_f^m) & -k_y(\frac{y_i}{\mathrm{f}_y}\sin\phi - \cos\phi) & -k_y(\frac{y_i}{\mathrm{f}_y}\cos\phi\cos\theta_f^m + \sin\phi\cos\theta_f^m) \\ -k_x\frac{d_x}{\mathrm{f}_x}\cos\phi\sin\theta_f^m & -k_x\frac{d_x}{\mathrm{f}_x}\sin\phi & -k_x\frac{d_x}{\mathrm{f}_x}\cos\phi\cos\theta_f^m \end{bmatrix}.$$

Expression (3.10) shows that the visual interaction model consists of two parts: first, the effect of target motion $\dot{X}_i^t \equiv \begin{bmatrix}\dot{x}_i^t & \dot{y}_i^t & \dot{d}_x^t\end{bmatrix}^T = \mathbf{J}_i V_f^t$, and second, the effect of mobile robot motion $\dot{X}_i^m \equiv \begin{bmatrix}\dot{x}_i^m & \dot{y}_i^m & \dot{d}_x^m\end{bmatrix}^T = \mathbf{B}_i u$. Thus, (3.10) can be rewritten as a dual-Jacobian equation such

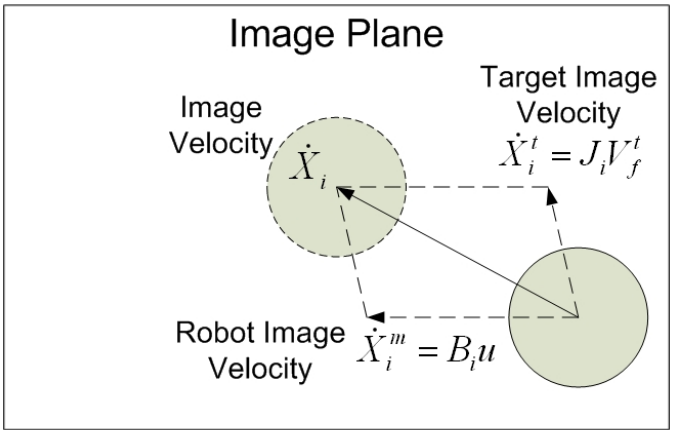


Fig. 3-4: Depicts the concept of dual-Jacobian visual interaction model (3.11).

that

$$\dot{X}_i = \dot{X}_i^t + \dot{X}_i^m = \mathbf{J}_i V_f^t + \mathbf{B}_i u \,, \tag{3.11}$$

where matrix $\mathbf{J}_i$, which is named *target image Jacobian*, transfers the target velocity V_f^t into target image velocity $\dot{X}_i^t$; matrix $\mathbf{B}_i$, which denotes *robot image Jacobian*, transfers the mobile robot control velocity u into robot image velocity $\dot{X}_i^m$. In other words, the image velocity $\dot{X}_i$ is caused by the combination of target image velocity $\dot{X}_i^t$ and robot image velocity $\dot{X}_i^m$. Figure 3-4 shows the concept of dual-Jacobian equation (3.11). Therefore, the visual interaction between robot and target in the image coordinate frame can be modeled as a dual-Jacobian visual interaction model (3.11), which combines the motion effect of mobile robot with moving target together.

Remark 3.1: The scalars $k_x = \mathrm{f}_x / z_c$ and $k_y = \mathrm{f}_y / z_c$ in (3.7) depend on the depth information between camera and target. The estimation of depth information is a demanding task in visual tracking control design; especially only one camera is used. Thus, an algorithm or sensor which provides the depth information is usually required during visual tracking process. In order to simplify the depth estimation problem, an alternative is to assume that the width of target is known *a priori*. Therefore, the scalars k_x and k_y can be calculated using the state variable d_x directly based on the fact that $k_x = \mathrm{f}_x / z_c = d_x / \mathrm{W}$ and $k_y = k_x \mathrm{f}_y / \mathrm{f}_x$, where W denotes the width of target for a specific target.

3.3 Visual Tracking Controller (VTC) Design

In this section, a visual tracking control law based on the proposed dual-Jacobian visual interaction model (3.11) for tracking a target of interest in the image plane is derived by exploiting feedback linearization and pole placement approaches.

3.3.1 Error Coordinate Transformation

In order to control the system from an initial state to the desired state, an error-state model will be helpful for us to design the tracking controller. Define the error state in the image plane such that

$$X_e = [x_e \quad y_e \quad d_e]^T = [\bar{\mathrm{x}}_i - x_i^* \quad \bar{\mathrm{y}}_i - y_i^* \quad \bar{\mathrm{d}}_x - d_x^*]^T, \tag{3.12}$$

where $\bar{\mathrm{X}}_i = [\bar{\mathrm{x}}_i \quad \bar{\mathrm{y}}_i \quad \bar{\mathrm{d}}_x]^T$ is the vector of fixed desired states in the image plane; $X_i^* = [x_i^* \quad y_i^* \quad d_x^*]^T$ is the vector of estimated states from the VSE (see Chapter 4). Based on the error state (3.12), the dynamic error state model in the image plane can be derived directly by taking the derivative of (3.12) such that

$$\dot{X}_e = -\dot{X}_i^t - \dot{X}_i^m = -\mathbf{J}_i V_f^t - \mathbf{B}_i u. \tag{3.13}$$

With the new coordinate X_e, the visual tracking control problem is transformed into a stability problem. If X_e converges to zero, then the visual tracking control problem is solved.

3.3.2 Visual Feedback Control Design

Based on the dynamic error state model (3.13), we choose the feedback linearization control law such that

$$u = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \mathbf{J}_i V_f^t) = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \dot{X}_i^t)\,, \tag{3.14}$$

where $\mathbf{K}_g$ is an 3-by-3 gain matrix. Substituting (3.14) into (3.13) yields

$$\dot{X}_e = -\mathbf{K}_g X_e\,. \tag{3.15}$$

Next, we choose the gain matrix such that

$$\mathbf{K}_g = diag(\alpha_1, \alpha_2, \alpha_3)\,, \tag{3.16}$$

in which $(\alpha_1, \alpha_2, \alpha_3)$ are three positive constants. Substituting (3.16) into (3.15) yields

$$\dot{X}_e = -\mathbf{K}_g X_e = -diag(\alpha_1, \alpha_2, \alpha_3) X_e\,. \tag{3.17}$$

Suppose that the initial error-state $X_e(t_0)$ is within the image plane. Then expression (3.17) indicates that

$$X_e(t)\,|_{t_0, X_e(t_0)} \equiv X_e(t; t_0, X_e(t_0)) = diag(e^{-\alpha_1(t-t_0)}, e^{-\alpha_2(t-t_0)}, e^{-\alpha_3(t-t_0)}) X_e(t_0)\,, \tag{3.18}$$

for some $t_0 \geq 0$. Because $(\alpha_1, \alpha_2, \alpha_3) > 0$ are positive constants, expression (3.18) leads to the following inequality:

$$\left\| X_e(t; t_0, X_e(t_0)) \right\| \leq e^{-\lambda_{\min}(\mathbf{K}_g)(t-t_0)} \left\| X_e(t_0) \right\| \quad \text{for all} \quad t \geq t_0\,, \tag{3.19}$$

where $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of matrix **A**, and $\|X\|$ denotes the 2-norm value of the vector *X*. From (3.19), it is clear that the system error-state satisfies $\left\| X_e(t; t_0, X_e(t_0) \right\| \leq \left\| X_e(t_0) \right\|$ and $\lim_{t \to \infty} \left\| X_e(t; t_0, X_e(t_0) \right\| = 0$, and thus the visual tracking control problem is solved. Summarizing the above discussions, we obtain the following theorem.

Theorem 3.1**:** Suppose the the initial system state X_i is within the image plane. Let

$(\alpha_1, \alpha_2, \alpha_3) > 0$ be three positive constants. Consider the closed-loop visual tracking system (3.13). If the matrix $\mathbf{B}_i$ is nonsingular, then the closed-loop visual tracking system (3.13) is exponentially stable by using the control law

$$u = \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \mathbf{J}_i V_f^t) \tag{3.20-1}$$

$$= \mathbf{B}_i^{-1}(\mathbf{K}_g X_e - \dot{X}_i^t)\,, \tag{3.20-2}$$

where matrices $\mathbf{B}_i$ and $\mathbf{J}_i$ are defined in (3.9) and (3.10), respectively; X_e is defined in (3.12); and $\mathbf{K}_g$ is a 3-by-3 diagonal gain matrix such that

$$\mathbf{K}_g = diag(\alpha_1, \alpha_2, \alpha_3)\,. \qquad \blacksquare$$

Proof: Consider the closed-loop visual tracking system (3.13). We first define a positive-definite Lyapunov function associated with the system error-state

$$V(x_e, y_e, d_e) = \frac{1}{2}(x_e^2 + y_e^2 + d_e^2)\,. \tag{3.21}$$

Taking the derivative of (3.21) yields

$$\dot{V}\,|_{(3.13)} = X_e^T \dot{X}_e\,|_{(3.13)} = -(X_e^T \mathbf{J}_i V_f^t + X_e^T \mathbf{B}_i u) = -X_e^T(\dot{X}_i^t + \mathbf{B}_i u) \equiv -f(u)\,, \tag{3.22}$$

where $f(u) = X_e^T(\dot{X}_i^t + \mathbf{B}_i u)$. In view of Lyapunov theory [46], expression (3.22) tells us that if $f(u) > 0$ then the equilibrium point of (3.13) is asymptotically stable. Substituting the control law (3.20) into $f(u)$, we then have

$$f(u) = X_e^T \mathbf{K}_g X_e\,, \tag{3.23}$$

where $\mathbf{K}_g = diag(\alpha_1, \alpha_2, \alpha_3)$, and $(\alpha_1, \alpha_2, \alpha_3)$ are three positive constants. Since $\mathbf{K}_g$ is a symmetric positive definite (SPD) matrix, the following inequality holds:

$$f(u) \geq \lambda_{\min}(\mathbf{K}_g)\|X_e\|^2 = \min(\alpha_1, \alpha_2, \alpha_3)\|X_e\|^2 > 0\,, \tag{3.24}$$

where $\lambda_{\min}(\mathbf{A})$ denotes the minimum eigenvalue of matrix **A**. Expression (3.24) concludes that the closed-loop visual tracking system (3.13) is exponentially stable and hence completes the proof. ■

Remark 3.2: Although the proposed image control law (3.20) results in a smooth convergence in the image plane, it still has to proof that the robot should have followed the target. The discussion of this problem is presented in Appendix D.

3.3.3 Singularity Analysis

The feedback linearization control law (3.20) poses a singularity problem of matrix $\mathbf{B}_i$. Let B_{mn} denote an element of matrix $\mathbf{B}_i$ corresponding to the *m*-th row and *n*-th column. By directly computing, the determinant of matrix $\mathbf{B}_i$ is given by

$$\det(\mathbf{B}_i) = B_{11}B_{22}B_{33} + B_{12}B_{23}B_{31} - B_{21}B_{12}B_{33} - B_{11}B_{23}B_{32}\,. \tag{3.25}$$

Based on (3.25), the singularity condition of matrix $\mathbf{B}_i$ can be found such that

$$\mathrm{f}_y = (y_i + \mathrm{S}d_x)\tan\phi\,, \tag{3.26}$$

where $\mathrm{S} = (\mathrm{f}_y\delta y)/(\mathrm{f}_x\mathrm{W})$ is a fixed scalar factor. Because of $k_x = \mathrm{f}_x/z_c = d_x/\mathrm{W}$ and $k_y = k_x\mathrm{f}_y/\mathrm{f}_x$, (3.26) can be rewritten such that

$$\tan\phi = \frac{\mathrm{f}_y}{y_i + k_y\delta y}\,. \tag{3.27}$$

Moreover, since $\mathrm{f}_y = k_y z_c$ and $y_i = k_y y_c$, equation (3.27) equals:

$$\tan\phi = \frac{z_c}{y_c + \delta y}\,. \tag{3.28}$$

As shown in Fig. 3-5, let ϕ' be the angle related to the location of the target, we have the following geometric relationship:

$$\tan(\phi + \phi') = \frac{z_c}{y_c + \delta y}\,. \tag{3.29}$$

From equations (3.28) and (3.29), it is clear that the matrix $\mathbf{B}_i$ becomes singular when ϕ' equals to 0 or π. The physical meaning of this is that the target is directly above or directly

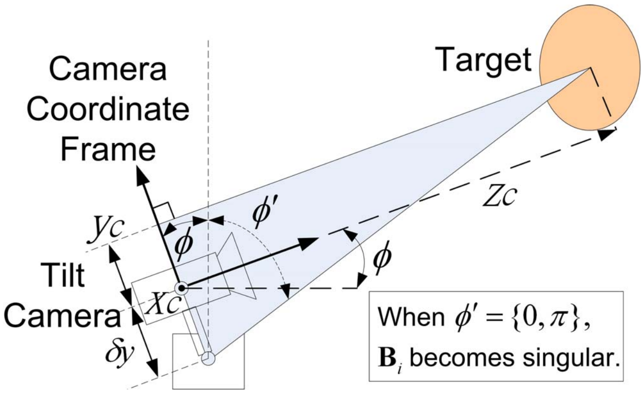


Fig. 3-5: Physical meaning of the singularity condition (3.26).

below the robot, and the robot will be unable to approach the target in any way due to insufficient degrees-of-freedom. Therefore, the robot will stop tracking temporarily under such circumstances.

Remark 3.3: The proposed visual interaction model (3.11) poses a question that the derivation of $d_x = k_x \mathrm{W}$ is not always hold for a cylindrical target used in the system modeling. As shown in Fig. 3-6, parameters W, f_x and z_c remain the same, but the cylindrical target is shifted along x_c. The projection d_x is not the same and includes an error δd_x. Because the camera is supposed to have a limited field of view, the error δd_x would be small and could be modeled as a system uncertainty. In the next section, the robustness analysis is presented in order to handle this kind of uncertainty.

3.4 Robustness Against System Model Uncertainty

In this section, the robustness of the VTC (3.20) is investigated against model uncertainties from camera parameters $(\mathrm{f}_x, \mathrm{f}_y)$, robot parameters (θ_f^m, ϕ), and target parameters $(\mathrm{W}, \dot{x}_i^t, \dot{y}_i^t, \dot{d}_x^t)$, etc. Consider the following closed-loop visual tracking system with parametric uncertainties:

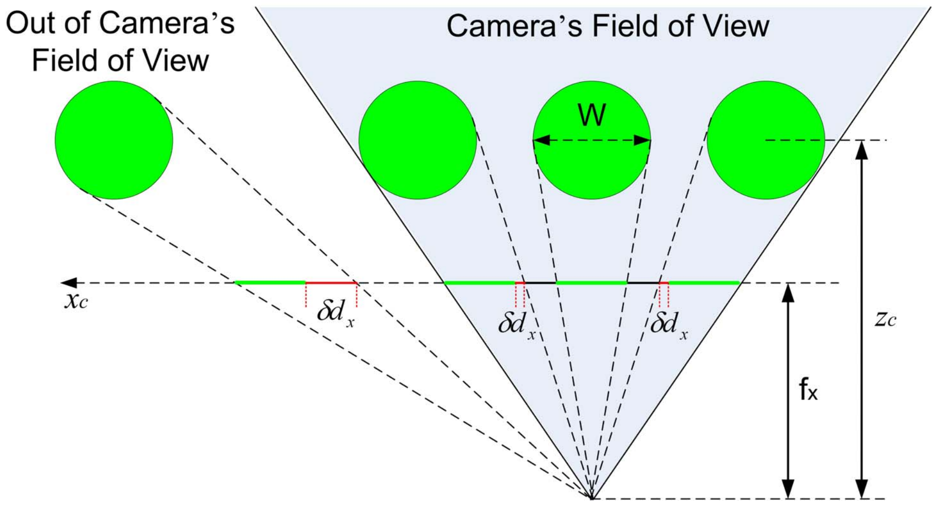


Fig. 3-6: Projection error in d_x.

$$\dot{X}_e = -\bar{\mathbf{J}}_i V_f^t - \bar{\mathbf{B}}_i u = -(\mathbf{J}_i + \delta\mathbf{J}_i)V_f^t - (\mathbf{B}_i + \delta\mathbf{B}_i)u \,, \tag{3.30}$$

where $\delta\mathbf{J}_i$ and $\delta\mathbf{B}_i$ are unknown bounded matrix-disturbances. Recall the positive-definite Lyapunov function defined in (3.21), the derivative of (3.21) with parametric uncertainties becomes

$$\dot{V}\,|_{(3.30)} = X_e^T \dot{X}_e\,|_{(3.30)} = -[X_e^T \bar{\mathbf{J}}_i V_f^t + X_e^T \bar{\mathbf{B}}_i u] = -[f(u) + \delta f(u)] = \dot{V}\,|_{(3.13)} - \delta f(u)\,, \tag{3.31}$$

where $\delta f(u) = X_e^T \delta\mathbf{J}_i V_f^t + X_e^T \delta\mathbf{B}_i u$ is unknown. Assume that $\delta\mathbf{J}_i$ and $\delta\mathbf{B}_i$ are bounded and there exists two positive values C_M and C_N such that

$$\delta f(u) \le C_M \|X_e\|^2 + C_N \|X_e\|\,. \tag{3.32}$$

We now introduce the following definition.

***Definition 3.1*:** The system (3.13) is said to be *practically stable* at the origin if a compact set S in $\Re^n$ exists containing the origin such that for each $X_e(t_0) \in S$, $X_e(t) \in S$ for all $t \ge t_0$.

In practical applications, the practical stability problem can be more easily handled. Now, the main result is presented as follows.

Theorem 3.2**:** Consider the closed-loop visual tracking system (3.13) with unknown bounded parametric uncertainties $\delta\mathbf{J}_i$ and $\delta\mathbf{B}_i$ defined in (3.30). Let C_M and C_N be two positive values defined in (3.32) and assume that the target's motion velocity $\|V_f^t\| \geq 0$ is bounded and small enough. Choose the controller u as given in expression (3.20) with gain matrix $\mathbf{K}_g = diag(\alpha_1, \alpha_2, \alpha_3) > 0$. If

(i) $[\lambda_{\max}(\mathbf{K}_g)]^{-1}\lambda_{\min}(\mathbf{K}_g) \to 1$.

(ii) $[\lambda_{\max}(\mathbf{K}_g)]^{-1} C_N \to 0$.

Then the origin is asymptotically stable under the condition $\|V_f^t\| = 0$, and the origin is practically stable under the condition $\|V_f^t\| > 0$. ■

Proof: Choose the controller u in (3.20) with parametric uncertainties such that

$$u = \bar{\mathbf{B}}_i^{-1}(\mathbf{K}_g X_e - \bar{\mathbf{J}}_i V_f^t) = \bar{\mathbf{B}}_i^{-1}(\mathbf{K}_g X_e - \dot{\bar{X}}_i^t), \tag{3.33}$$

where $\dot{\bar{X}}_i^t = \bar{\mathbf{J}}_i V_f^t$. Substituting (3.33) into $\delta f(u)$ defined in (3.31) yields

$$\begin{aligned} \delta f(u) &= X_e^T \delta\mathbf{J}_i V_f^t + X_e^T \delta\mathbf{B}_i [\bar{\mathbf{B}}_i^{-1}(\mathbf{K}_g X_e - \bar{\mathbf{J}}_i V_f^t)] \\ &= X_e^T \delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1} \mathbf{K}_g X_e + X_e^T(\delta\mathbf{J}_i - \delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1} \bar{\mathbf{J}}_i) V_f^t \\ &\leq \lambda_{\max}(\mathbf{K}_g)\|\delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1}\|\|X_e\|^2 + \|\delta\mathbf{J}_i - \delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1}\bar{\mathbf{J}}_i\|\|V_f^t\|\|X_e\| \\ &= C_M\|X_e\|^2 + C_N\|X_e\|, \end{aligned} \tag{3.34}$$

where $C_M = \lambda_{\max}(\mathbf{K}_g)\|\delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1}\|$, $C_N = \|\delta\mathbf{J}_i - \delta\mathbf{B}_i \bar{\mathbf{B}}_i^{-1}\bar{\mathbf{J}}_i\|\|V_f^t\|$, $\lambda_{\max}(\mathbf{A})$ denotes the maximum eigenvalue of matrix **A**, and $\|\mathbf{A}\|$ denotes the Euclidean norm value of the matrix **A**. From (3.31), (3.33) and (3.34), it follows that

$$\begin{aligned} \dot{V}|_{(3.13)} = \dot{V}|_{(3.30)} + \delta f(u) &= -X_e^T \mathbf{K}_g X_e + C_M\|X_e\|^2 + C_N\|X_e\| \\ &\leq -\{[\lambda_{\min}(\mathbf{K}_g) - C_M]\|X_e\| - C_N\}\|X_e\|. \end{aligned} \tag{3.35}$$

Expression (3.35) implies that if $[\lambda_{\min}(\mathbf{K}_g) - C_M]\|X_e\| - C_N > 0$ can be guaranteed, then $\dot{V}|_{(3.13)} < 0$ is satisfied and thus the system has the robust property w.r.t. the parametric uncertainties.

From (3.34), the condition $[\lambda_{\min}(\mathbf{K}_g) - C_M]\|X_e\| - C_N > 0$ can be rewritten such that

$$\|X_e\| > [\lambda_{\min}(\mathbf{K}_g) - C_M]^{-1} C_N = \left[\rho_\lambda(\mathbf{K}_g) - \left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\|\right]^{-1}\left[\lambda_{\max}(\mathbf{K}_g)\right]^{-1} C_N, \tag{3.36}$$

where $\rho_\lambda(\mathbf{K}_g) = \left[\lambda_{\max}(\mathbf{K}_g)\right]^{-1}\lambda_{\min}(\mathbf{K}_g)$. In general, we have that $\left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\| < 1$. By assumption (i) of the theorem, expression (3.36) follows that when $\rho_\lambda(\mathbf{K}_g) \to 1$ satisfies, $\exists\varepsilon > 0$ such that

$$X_e \to \left\{\zeta \in \Re^n : \|\zeta\| < \left[\rho_\lambda(\mathbf{K}_g) - \left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\|\right]^{-1}\left[\lambda_{\max}(\mathbf{K}_g)\right]^{-1} C_N + \varepsilon\right\}. \tag{3.37}$$

Because of $C_N = \left\|\delta\mathbf{J}_i - \delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\mathbf{J}_i\right\|\left\|V_f^t\right\|$, the condition $\left\|V_f^t\right\| = 0$ follows that

$$\|X_e\| < \varepsilon \quad \text{and} \quad \dot{V}|_{(3.13)} \le -\left[\rho_\lambda(\mathbf{K}_g) - \left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\|\right]\|X_e\|^2 < 0. \tag{3.38}$$

Thus, there exists a constant $\delta > 0$, $\delta < \varepsilon$, $\exists T$ such that

$$\|X_e(t_0)\| < \varepsilon \Rightarrow \|X_e(t)\| < \delta \quad \forall t \ge t_0 + T, \tag{3.39}$$

which means that the origin is asymptotically stable under the condition $\left\|V_f^t\right\| = 0$.

On the other hand, the condition $\left\|V_f^t\right\| > 0$ follows that when $\rho_\lambda(\mathbf{K}_g) \to 1$ satisfies, $\exists\varepsilon > 0$ such that

$$\dot{V}|_{(3.13)} < 0 \quad \text{for all} \quad \|X_e\| < \left[\rho_\lambda(\mathbf{K}_g) - \left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\|\right]^{-1}\left[\lambda_{\max}(\mathbf{K}_g)\right]^{-1} C_N + \varepsilon. \tag{3.40}$$

Since $\left\|V_f^t\right\|$ is supposed to be small enough, $C_N \propto \left\|V_f^t\right\|$ is sufficiently small. By assumption (ii) of the theorem, there exists a sufficiently large $\lambda_{\max}(\mathbf{K}_g)$ and a constant $\delta_1 > 0$ such that $\|X_e\| < c_0$, where $c_0 = \left[\rho_\lambda(\mathbf{K}_g) - \left\|\delta\mathbf{B}_i\overline{\mathbf{B}}_i^{-1}\right\|\right]^{-1}\delta_1 + \varepsilon$. Choose the set $S = \left\{\zeta \in \Re^n : \|\zeta\| \le c_1\right\}$, where $c_1 > 0$ satisfies $c_1 > c_0$. Then, any trajectory that begins at any $X_e(t_0) \in S$ will remain in S for all $t \geq t_0$ under the control law (3.20). Thus, the origin is practically stable under the condition $\left\| V_f^t \right\| > 0$. This completes the proof. ■

In Chapter 5, the result of *Theorem 3.2* will be validated by practical experiments. Further, in realization of the control schemes, it was noted that the quantization error in velocity commands degrade the performance of the controller and might make the system unstable. It is therefore interesting to study the robustness issues related to velocity quantization uncertainty. In the following section, a robust control law based on Lyapunov's direct method will be derived to overcome the velocity quantization uncertainty in practical control systems.

3.5 Robustness Against Velocity Quantization Error

When tracking a target, it is desirable for the robot to have a smooth motion in human-robot interaction. But in such circumstances, one will face the problem caused by velocity quantization error in practical implementation. In this section, a robust control law is derived to eliminate the velocity quantization error encountered in practical control systems based on the dynamic error state model defined in (3.13). To do so, a stability necessary condition (SNC) is first derived for ensuring asymptotic stability and practical stability of the closed-loop visual tracking system through Lyapunov's direct method. The robust control law is then proposed to guarantee that the visual tracking system satisfies the SNC and hence complete the controller design.

3.5.1 Stability necessary condition (SNC)

Digital control systems usually have *uniform* quantization errors due to the finite-length effects on the sample values [49]. In other words, the ideal (theoretical) control command u is quantized such that

$$\bar{u} = u + \delta u\,, \tag{3.41}$$

where $\bar{u}$ denotes the practical (actual) control command sent to the robot actuator, and δu represents the uniform quantization error encountered in the system. Thus, in practice, (3.22) becomes

$$\dot{V}\,|_{\substack{(3.13)\\(3.41)}} = -f(\bar{u}) = -[\bar{f}(u) - \delta f(\bar{u}) + \delta\bar{f}(\delta u)] = \dot{V}\,|_{(3.13)} - \delta\bar{f}(\delta u) < 0\,, \tag{3.42}$$

where $\delta\bar{f}(\delta u) = X_e^T\bar{\mathbf{B}}_i\delta u - X_e^T\delta\mathbf{B}_i\delta u \cong X_e^T\bar{\mathbf{B}}_i\delta u$. Expression (3.42) shows that if the controller u satisfies the assumptions (i) and (ii) of Theorem 3.2, then the equilibrium point of the system (3.13) still can be unstable under the condition $\delta\bar{f}(\delta u) < 0$. Therefore, one has the following SNC in practical control implementation:

SNC**:** Consider the closed-loop visual tracking system (3.13) with unknown bounded parametric uncertainties $\delta\mathbf{J}_i$ and $\delta\mathbf{B}_i$ defined in (3.30). Let δu denote the velocity output quantization error in practical systems. Suppose that the controller u given in (3.20) satisfies the assumptions (i) and (ii) of Theorem 3.2. Then, the result of Theorem 3.2 holds under the condition $\delta\bar{f}(\delta u) = X_e^T\bar{\mathbf{B}}_i\delta u \geq 0$ ■

3.5.2 Proposed robust control law

SNC implies that a practical system may become unstable under the condition $\delta\bar{f}(\delta u) < 0$. Our goal is to design a robust control law which not only guarantees SNC to be always satisfied but also increases the convergence rate of the control system.

First, we expand $\delta\bar{f}(\delta u)$ such that

$$\delta\bar{f}(\delta u) = \gamma_1\delta v_f^m + \gamma_2\delta w_f^m + \gamma_3\delta w_t^m\,, \tag{3.43}$$

where

$$\gamma_1 = \overline{B}_{11}x_e + \overline{B}_{21}y_e + \overline{B}_{31}d_e,\ \ \gamma_2 = \overline{B}_{12}x_e + \overline{B}_{22}y_e + \overline{B}_{32}d_e,\ \ \gamma_3 = \overline{B}_{13}x_e + \overline{B}_{23}y_e + \overline{B}_{33}d_e,$$

$\overline{B}_{mn}$ denotes the element of matrix $\overline{\mathbf{B}}_i$ corresponding to the *m*-th row and *n*-th column. Next, based on the velocity transformation

$$\begin{bmatrix} v_l^m \\ v_r^m \\ w_t^m \end{bmatrix} = \begin{bmatrix} 1 & -\mathrm{D}/2 & 0 \\ 1 & \mathrm{D}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_f^m \\ w_f^m \\ w_t^m \end{bmatrix} \equiv \mathbf{\Lambda} \begin{bmatrix} v_f^m \\ w_f^m \\ w_t^m \end{bmatrix}, \tag{3.44}$$

where D represents the distance between two drive wheels, expression (3.43) becomes

$$\delta\bar{f}(\delta u) = [(\gamma_1/2) + (\gamma_2/\mathrm{D})]\delta v_r^m + [(\gamma_1/2) - (\gamma_2/\mathrm{D})]\delta v_l^m + \gamma_3 \delta w_t^m. \tag{3.45}$$

Expression (3.45) tells us that if each term in (3.45) is equal to a nonnegative value, then $\delta\bar{f}(\delta u) \geq 0$ can be guaranteed. Based on this idea, a variable structure robust control law is derived such that:

$$\bar{v}_r^{m*} = \begin{cases} \bar{v}_r^m - \varepsilon_r sign(\delta v_r^m), & if\ \ sign\,([(\gamma_1/2) + (\gamma_2/\mathrm{D})]\delta v_r^m) = -1 \\ \bar{v}_r^m, & otherwise. \end{cases},$$

$$\bar{v}_l^{m*} = \begin{cases} \bar{v}_l^m - \varepsilon_l sign(\delta v_l^m), & if\ \ sign\,([(\gamma_1/2) - (\gamma_2/\mathrm{D})]\delta v_l^m) = -1 \\ \bar{v}_l^m, & otherwise. \end{cases},$$

$$\bar{w}_t^{m*} = \begin{cases} \bar{w}_t^m - \varepsilon_t sign(\delta w_t^m), & if\ \ sign\,(\gamma_3 \delta w_t^m) = -1 \\ \bar{w}_t^m, & otherwise. \end{cases}, \tag{3.46}$$

where

$$sign(x) = \begin{cases} -1, & if\ \ x < 0 \\ 0, & if\ \ x = 0 \\ 1, & if\ \ x > 0 \end{cases},$$

$(\bar{v}_l^{m*}, \bar{v}_r^{m*}, \bar{w}_t^{m*})$ are the outputs of the proposed robust control law for tracking control of the robot, and $(\varepsilon_l, \varepsilon_r, \varepsilon_t)$ are three positive constants such that

$$(\varepsilon_l, \varepsilon_r, \varepsilon_t) = (\sup|\delta v_l^m|, \sup|\delta v_r^m|, \sup|\delta w_t^m|). \tag{3.47}$$

The value of constants $(\varepsilon_l, \varepsilon_r, \varepsilon_t)$ is calculated based on the range of quantization errors $(\delta v_l^m, \delta v_r^m, \delta w_t^m)$ that are dependent on the resolution of robot motion control module. It is easy to show that the outputs of the proposed robust control law guarantee each term of (3.45) is equal to a nonnegative value, and thus SNC is satisfied. Further, because of $\dot{V}|_{\substack{(3.13)\\(3.41)}} = \dot{V}|_{(3.13)} - \delta\bar{f}(u) \le \dot{V}|_{(3.13)} < 0$, an increase of the convergence rate of the practical system is obtained. In Chapter 5, the stability characteristic and convergence performance of the proposed control law are verified in practical experiments.

3.6 Summary

A novel dual-Jacobian visual interaction model that represents the visual interaction between a mobile robot and a dynamic moving target in the image plane has been derived in this chapter. Based on this visual interaction model, a robust VTC is proposed to solve dynamic moving target visual tracking control problem with/without the information of target's 3D motion velocity. A study of system model and velocity quantization uncertainties in visual tracking control of a wheeled mobile robot has also been discussed in this chapter. In the parametric robustness analysis, we have shown that the proposed VTC provides some degree of robustness against system model uncertainties. Moreover, based on Lyapunov theory, the robust control law efficiently overcomes the unmodelled quantization effect in the velocity commands. In Chapter 5, several experimental results will be presented and discussed to verify the effectiveness of the proposed VTC, in terms of tracking performance, system convergence, and robustness.

Chapter 4

Design of Visual State Estimator

4.1 Introduction

Recall Theorem 3.1, the visual tracking control law (3.20) requires information about target 3D velocity V_f^t or target image velocity $\dot{X}_i^t$. If V_f^t is known, the first visual tracking control law (3.20-1) only needs an estimate of target status X_i to calculate the control signal *u*. However, in practical applications, it is difficult to estimate V_f^t when using only one camera in real-time operations. In this situation, the second visual tracking control law (3.20-2) provides a useful solution which only needs the target image velocity $\dot{X}_i^t$ in the image plane. In this chapter, two VSEs will be proposed in order to estimate the necessary information for later used by the VTC. The first VSE is developed under the condition that target velocity V_f^t is known, and the second VSE is designed by releasing this condition. These designs will facilitate more general performance of the proposed tracking control scheme in the image plane. Another advantage of the proposed VSEs is that they can estimate the optimal target state from the observed image, which has both random noise and temporary occlusion uncertainties. Therefore, with the proposed VSEs, the visual tracking control system can be robust to the uncertainties of image noise and temporary occlusion.

The rest of this chapter is organized as follows. Section 4.2 presents the design of VSE with target velocity information to estimate the optimal system state in the image plane for handling the uncertainties caused by image noise and temporary occlusion. In Section 4.3, the results of VSE design without target velocity information is presented. Section 4.4 summarizes the contributions of this work. The results of computer simulations and practical experiments to validate the effectiveness of the proposed VSEs, in terms of estimation performance and robustness, will be reported in Chapter 5.

4.2 VSE Design with Target Velocity Information (VSE-WTV)

In the case of knowing the target 3D velocity V_f^t, we develop the VSE-WTV based on the system model (3.9) to estimate the optimal target status X_i in the image plane. To achieve this, a propagation model is required in order to help the design of VSE-WTV. This subsection presents the derivation of the required propagation model and the design of the proposed VSE-WTV algorithm.

4.2.1 Propagation Model for VSE-WTV

Since actual image processing is discrete, the first step of the derivation of propagation model is to discretize the system model (3.9) into corresponding discrete form. By the definition $\dot{x}(t)=\lim_{\mathrm{T}\to 0}[x(t)-x(t-\mathrm{T})]/\mathrm{T}$, T denotes the sampling time of the digital system, we can approximate the system model (3.9) as

$$X_i^p[n]=(\mathbf{I}_3+\mathrm{T}\mathbf{A}_i)X_i^*[n-1]+\mathrm{T}\mathbf{B}_i u_{n-1}+\mathrm{T}C_i\text{, for } n=1,2,... \qquad (4.1)$$

where $X_i^p[n]$ is the propagated system state at time step *n*, $\mathbf{I}_3$ is a 3-by-3 identity matrix, $X_i^*[n-1]=\begin{bmatrix}x_i^* & y_i^* & d_x^*\end{bmatrix}^T$ denotes the estimated system state at time step *n*-1, and $u_{n-1}=\begin{bmatrix}v_f^m & w_f^m & w_t^m\end{bmatrix}^T$ is the output of VTC at time step *n*-1.

The second step of the derivation of propagation model is to analyze the error covariance matrix of discrete-time propagation equation (4.1). We first introduce the following variables

$$x_i^* = x_i + \delta x_i ,\ \ y_i^* = y_i + \delta y_i ,\ \ d_x^* = d_x + \delta d_x ,$$

$$v_f^{m*} = v_f^m + \delta v_f^m ,\ \ w_f^{m*} = w_f^m + \delta w_f^m ,\ \ w_t^{m*} = w_t^m + \delta w_t^m , \tag{4.2}$$

where (x_i, y_i, d_x) denote the system state variables; (x_i^*, y_i^*, d_x^*) denote the estimated state variables and $(\delta x_i, \delta y_i, \delta d_x)$ are the corresponding state estimation errors; $(v_f^{m*}, w_f^{m*}, w_t^{m*})$ and (v_f^m, w_f^m, w_t^m), respectively, denote the velocity inverse transformation of the estimated velocities $(v_{lc}^m, v_{rc}^m, w_{tc}^m)$ and velocity commands (v_l^m, v_r^m, w_t^m) based on (3.44) such that

$$\begin{bmatrix} v_f^{m*} \\ w_f^{m*} \\ w_t^{m*} \end{bmatrix} = \mathbf{\Lambda}^{-1} \begin{bmatrix} v_{lc}^m \\ v_{rc}^m \\ w_{tc}^m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ -1/\mathrm{D} & 1/\mathrm{D} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{lc}^m \\ v_{rc}^m \\ w_{tc}^m \end{bmatrix}, \text{ and } \begin{bmatrix} v_f^m \\ w_f^m \\ w_t^m \end{bmatrix} = \mathbf{\Lambda}^{-1} \begin{bmatrix} v_l^m \\ v_r^m \\ w_t^m \end{bmatrix}, \tag{4.3}$$

where D is the distance between two drive wheels. Using (4.3), the estimated velocity errors $(\delta v_f^m, \delta w_f^m, \delta w_t^m)$ can be estimated by the velocity inverse transformation

$$\begin{bmatrix} \delta v_f^m \\ \delta w_f^m \\ \delta w_t^m \end{bmatrix} = \begin{bmatrix} v_f^{m*} - v_f^m \\ w_f^{m*} - w_f^m \\ w_t^{m*} - w_t^m \end{bmatrix} = \mathbf{\Lambda}^{-1} \begin{bmatrix} v_{lc}^m - v_l^m \\ v_{rc}^m - v_r^m \\ w_{tc}^m - w_t^m \end{bmatrix} = \mathbf{\Lambda}^{-1} \begin{bmatrix} \delta v_l^m \\ \delta v_r^m \\ \delta w_t^m \end{bmatrix}. \tag{4.4}$$

Next, substituting (4.2) into (4.1) and canceling common terms, the state estimation errors can be approximated by neglecting the higher-order terms in the discrete-time error propagation equation such that

$$\delta X_i[n] = \mathbf{A}_\delta \delta X_i[n-1] + \mathrm{T}\mathbf{B}_i \delta u_{n-1} , \tag{4.5}$$

where $\delta X_i[n-1] = \begin{bmatrix} \delta x_i & \delta y_i & \delta d_x \end{bmatrix}^T$ denotes the error propagation state, $\delta u_{n-1} = \begin{bmatrix} \delta v_f^m & \delta w_f^m & \delta w_t^m \end{bmatrix}^T$ is the estimated velocity error, and

$$\mathbf{A}_\delta = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

where

$$a_{11} = 1 + \mathrm{T}\left\{ A_1 + \frac{1}{\mathrm{f}_x}[(k_x v_f^{m*} + 2x_i^* w_f^{m*})\cos\phi - k_x(\delta y + \frac{y_i^*}{k_y})w_t^{m*}] \right\},$$

$$a_{12} = -\mathrm{T}\frac{\mathrm{f}_x}{\mathrm{f}_y}(w_f^{m*}\sin\phi + \frac{x_i^*}{\mathrm{f}_x}w_t^{m*}),$$

$$a_{13} = \frac{\mathrm{T}}{d_x^*}\left[x_i^* A_1 + k_x(\frac{x_i^*}{\mathrm{f}_x}v_f^{m*}\cos\phi - \delta y w_f^{m*}\sin\phi - \frac{x_i^*\delta y}{\mathrm{f}_x}w_t^{m*} + v_f^z \sin\theta_f^m - v_f^x\cos\theta_f^m) \right],$$

$$a_{21} = \mathrm{T}\frac{\mathrm{f}_y}{\mathrm{f}_x}(\sin\phi + \frac{y_i^*}{\mathrm{f}_y}\cos\phi)w_f^{m*},$$

$$a_{22} = 1 + \mathrm{T}\left[A_2 + \frac{v_f^{m*}}{\mathrm{f}_y}\cos\phi + \frac{x_i^*}{\mathrm{f}_x}w_f^{m*}\cos\phi - \frac{1}{\mathrm{f}_y}(2y_i^* + k_y\delta y)w_t^{m*} \right],$$

$$a_{23} = \frac{\mathrm{T}}{d_x^*}\left\{ y_i^* A_2 + k_y[(\sin\phi + \frac{y_i^*}{\mathrm{f}_y}\cos\phi)v_f^{m*} - \frac{y_i^*\delta y}{\mathrm{f}_y}w_t^{m*} + v_f^y\cos\phi - v_f^x\sin\phi\sin\theta_f^m - v_f^z\sin\phi\cos\theta_f^m] \right\},$$

$$a_{31} = \mathrm{T}\frac{d_x^*}{\mathrm{f}_x}w_f^{m*}\cos\phi,$$

$$a_{32} = -\mathrm{T}\frac{d_x^*}{\mathrm{f}_y}w_t^{m*},$$

$$a_{33} = 1 + \mathrm{T}\left[2A_1 + (\frac{2k_y}{\mathrm{f}_y}v_f^{m*} + \frac{x_i^*}{\mathrm{f}_x}w_f^{m*})\cos\phi - \frac{(2k_y\delta y + y_i^*)w_t^{m*}}{\mathrm{f}_y} \right].$$

Because $\delta X_i[n-1]$ and $\delta u[n-1]$ are uncorrelated, the covariance matrix propagation equation can be obtained by adopting (4.5) such that

$$\mathbf{P}_n = E\{\delta X_i[n]\delta X_i^T[n]\} = \mathbf{A}_\delta \mathbf{P}_{n-1}^* \mathbf{A}_\delta^T + \mathrm{T}^2 \mathbf{B}_i \mathbf{W}_{n-1} \mathbf{B}_i^T, \qquad (4.6)$$

where $\mathbf{W}_{n-1} = E\{\delta u[n-1]\delta u^T[n-1]\}$ is the covariance matrix of the estimated velocity error. Applying (4.1) and (4.6), the system state and the corresponding covariance matrix in the next time steps can be propagated.

4.2.2 Observation and Correction for VSE-WTV

In this subsection, the propagated system state and the propagation covariance matrix will be corrected using the observation of camera

$$Z_n = \mathbf{I}_3 X_i[n] + \delta Z_n \, ,$$

where $\delta Z_n \sim N(0, \mathbf{R}_n)$ denotes Gaussian observation uncertainty with zero mean and covariance matrix $\mathbf{R}_n$ at instant *n*. The correction procedure will be [50]

$$X_i^*[n] = X_i^p[n] + \mathbf{K}_n \{ Z_n - X_i^p[n] \}, \text{ and} \tag{4.7}$$

$$\mathbf{P}_n^* = (\mathbf{I}_3 - \mathbf{K}_n)\mathbf{P}_n \, , \tag{4.8}$$

where $\mathbf{K}_n$ is the Kalman gain matrix given by

$$\mathbf{K}_n = \mathbf{P}_n (\mathbf{P}_n + \mathbf{R}_n)^{-1} \, . \tag{4.9}$$

Finally, the corrected system state $X_i^*[n]$ and the corresponding covariance matrix $\mathbf{P}_n^*$ are the optimal estimates at sample instant *n*.

4.2.3 Summary of the Proposed VSE-WTV Algorithm

Based on the propagation equations (4.1) and (4.6) and the correction equations (4.7)-(4.9), the VSE-WTV can be summarized as follows:

1) Assume that the initial position of target is located in the field-of-view of the camera, then initialize the estimated system state $X_i^*[0]$ and propagation covariance matrix $\mathbf{P}_0$ by the first observation such that $X_i^*[0] = Z_0$ and $\mathbf{P}_0 = \mathbf{R}_0$. The proposed VTC starts working.

2) Compute the propagated system state $X_i^p[n]$ and the corresponding covariance matrix $\mathbf{P}_n$ using (4.1) and (4.6), respectively.

3) If the target to be tracked is detected in the observed image, then compute the Kalman gain matrix $\mathbf{K}_n$ using (4.9); else set $X_i^*[n] = X_i^p[n]$ and $\mathbf{P}_n^* = \mathbf{P}_n$, go to step 5.

4) Correct the estimated state vector $X_i^*[n]$ and the corresponding covariance matrix $\mathbf{P}_n^*$ using (4.7) and (4.8), respectively.

5) Let $X_i^*[n-1] = X_i^*[n]$ and $\mathbf{P}_{n-1}^* = \mathbf{P}_n^*$, then go to step 2. ■

Remark 4.1: Because the observation uncertainty usually varies with the conditions of target motion (such as orientation and rotation of the target) and working environment (such as light variation and occlusion), the corresponding covariance matrix $\mathbf{R}_n$ would be time-varying for different operating conditions. In order to overcome this problem, a real-time self-tuning algorithm to choose a suitable observation covariance matrix $\mathbf{R}_n$ in varying environmental conditions will be proposed in next section and adopted into the design of both VSEs.

4.3 VSE Design without Target Velocity Information (VSE-WoTV)

The VSE-WoTV aims to estimate the optimal target status X_i and target image velocity $\dot{X}_i^t$ from image space directly when the target 3D velocity V_f^t is unknown. In this case, the dual-Jacobian visual interaction model (3.11) plays an essential role in the estimator design. The same procedure presented in Section 4.2 will be adopted into the design of VSE-WoTV algorithm.

4.3.1 Propagation Model for VSE-WoTV

To derive the required propagation model for the design of VSE-WoTV, the first step is to discretize the system model (3.11) into corresponding discrete form such that

$$X_i^p[n] = X_i^*[n-1] + \mathrm{T}\dot{X}_i^t[n-1] + \mathrm{T}\mathbf{B}_i u_{n-1}, \text{ for } n = 1, 2, \ldots \tag{4.10}$$

Suppose that the target motion is close to a smooth motion during a sampling time, and then the target image velocity can be approximated as a constant velocity between two consecutive sample instants

$$\dot{X}_i^t[n] = \dot{X}_i^t[n-1]. \tag{4.11}$$

Based on (4.10) and (4.11), the propagation model of the VSE-WoTV is given by

$$X_n^p = \begin{bmatrix} \mathbf{I}_3 & \mathrm{T}\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} X_{n-1}^* + \begin{bmatrix} \mathrm{T}\mathbf{B}_i \\ \mathbf{0}_3 \end{bmatrix} u_{n-1} \equiv \mathbf{A}_{est} X_{n-1}^* + \mathbf{B}_{est} u_{n-1}, \tag{4.12}$$

where $(X_n^p)^T = [(X_i^p[n])^T \quad (\dot{X}_i^t[n])^T]$ is the propagated system state at an instant *n*, $\mathbf{0}_3$ is a 3-by-3 zero matrix, and $(X_{n-1}^*)^T = \left[(X_i^*[n-1])^T \quad (\dot{X}_i^t[n-1])^T\right]$ denotes the estimated system state at the previous instant *n*-1.

Next, the covariance matrix of propagation equation (4.12) at an instant *n* is given by

$$\mathbf{P}_n = \mathbf{A}_{est} \mathbf{P}_{n-1}^* \mathbf{A}_{est}^T + \mathbf{Q}_{n-1}, \tag{4.13}$$

where $\mathbf{P}_{n-1}^*$ is the estimated covariance matrix at the previous instant *n*-1, and $\mathbf{Q}_n$ is the covariance matrix of the Gaussian propagation uncertainty. Note that (4.11) is an oversimplified assumption and will induce propagation error when the target motion is not smooth. However, this kind of error can be corrected by the observation information.

Remark 4.2: A major difference between VSE-WTV and VSE-WoTV is that the propagation covariance matrix of VSE-WoTV includes the covariance matrix of the Gaussian propagation uncertainty, $\mathbf{Q}_n$. The main reason is that if V_f^t is known *a-priori*, the prediction of the target state would be more precise with small uncertainty. Thus, the covariance matrix $\mathbf{Q}_n$ can be approximated by the matrix $\mathrm{T}^2\mathbf{B}_i\mathbf{W}_{n-1}\mathbf{B}_i^T$ in the propagation covariance matrix of VSE-WTV. On the other hand, if V_f^t is unknown, the uncertainty of the target's prediction state would become larger. Therefore, the propagation covariance matrix of VSE-WoTV should take the covariance matrix $\mathbf{Q}_n$ into account.

4.3.2 Observation and Correction for VSE-WoTV

Because the observed image only contains the information of target status X_i in each instant, the observation model of the VSE-WoTV is given by

$$Z_n = \begin{bmatrix}\mathbf{I}_3 & \mathbf{0}_3\end{bmatrix} X_n + \delta Z_n \equiv \mathbf{H}_{est} X_n + \delta Z_n, \tag{4.14}$$

where $\delta Z_n \sim N(0, \mathbf{R}_n)$ is the observation uncertainty with zero mean and covariance matrix $\mathbf{R}_n$. Based on equations (4.12) to (4.14), the optimal estimate and the corresponding covariance matrix at a sample instant *n* are given by [50]

$$X_n^* = X_n^p + \mathbf{K}_n (Z_n - \mathbf{H}_{est} X_n^p) \text{ and } \mathbf{P}_n^* = (\mathbf{I}_6 - \mathbf{K}_n \mathbf{H}_{est}) \mathbf{P}_n, \tag{4.15}$$

where $\mathbf{K}_n = \mathbf{P}_n \mathbf{H}_{est}^T (\mathbf{H}_{est} \mathbf{P}_n \mathbf{H}_{est}^T + \mathbf{R}_n)^{-1}$ is the Kalman gain matrix, and $\mathbf{I}_6$ is a 6-by-6 identity matrix.

4.3.2 Self-Tuning Algorithm

Although expression (4.15) provides the best linear estimates at each instant, the filter performance still depends on the covariance matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$. Thus, a difficult problem in Kalman filter applications is to determine the values of matrices $\mathbf{Q}_n$ and $\mathbf{R}_n$ for computing Kalman gain matrix $\mathbf{K}_n$ [51]. Moreover, the observation uncertainty usually varies with the conditions of target motion and working environment; the corresponding covariance matrix $\mathbf{R}_n$ would be time-varying for various operating conditions. These problems motivate us to combine a self-tuning algorithm with Kalman filter to choose a suitable observation covariance matrix $\mathbf{R}_n$ in varying environmental conditions. On the other hand, because the propagation uncertainty and the corresponding covariance matrix $\mathbf{Q}_n$ are difficult to estimate online, the propagation covariance matrix $\mathbf{Q}_n$ will be fixed at initialization without updating in this design.

The proposed self-tuning algorithm attempts to estimate the minimum variance of a set of observation data recorded over time. To do so, a linear-least-squares regression method is adopted to analyze the observed time series data [52]. The typical linear regression model for a discrete time series is given by

$$y_n = an + b + \varepsilon_n , \tag{4.16}$$

where the residual ε_n is a random variable with zero mean, (a, b) are the parameters to be determined by minimizing the variance of residuals. Figure 4-1 shows the concept of the linear-least-squares regression, in which the solid line is the observed time series, and the dotted line indicates the best linear fitting $\hat{y}_n = an + b$ with minimum residual variance σ^2.

Let k denote the length of the observed time series. Based on the linear regression model (4.16), the observed time series can be modeled as

$$Y = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ \vdots & \vdots \\ k & 1 \end{bmatrix} \theta + \varepsilon \equiv \mathbf{A}_{st}\theta + \varepsilon , \tag{4.17}$$

where $Y = [y_1 \quad y_2 \quad \cdots \quad y_k]^T$ is the vector of observed data over time, and $\varepsilon = [\varepsilon_1 \quad \varepsilon_2 \quad \cdots \quad \varepsilon_k]^T$ represents the corresponding residuals. $\theta = [a \quad b]^T$ is the parameter vector to be detected such that

$$\theta^* = \min_\theta \operatorname{var}(\varepsilon) = \min_\theta \|\varepsilon\| = \min_\theta \|Y - \mathbf{A}_{st}\theta\| , \tag{4.18}$$

where $\operatorname{var}(X)$ is the variance of vector *X*, and $\|X\|$ is the 2-norm value of vector *X*. The optimal solution of (4.18) will be the least-squares solution such that

$$\theta^* = \mathbf{A}_{st}^{+} Y , \tag{4.19}$$

where $\mathbf{A}_{st}^{+} = (\mathbf{A}_{st}^T \mathbf{A}_{st})^{-1} \mathbf{A}_{st}^T$ denotes the pseudo-inverse matrix of $\mathbf{A}_{st}$. Substituting (4.19) into (4.17), the residual vector with minimum variance can be obtained by

$$\varepsilon^* = Y - \mathbf{A}_{st}\mathbf{A}_{st}^{+} Y = (\mathbf{I}_k - \mathbf{A}_{st}(\mathbf{A}_{st}^T \mathbf{A}_{st})^{-1} \mathbf{A}_{st}^T) Y \equiv \mathbf{T}_{st} Y , \tag{4.20}$$

where $\mathbf{T}_{st} = \mathbf{I}_k - \mathbf{A}_{st}(\mathbf{A}_{st}^T \mathbf{A}_{st})^{-1} \mathbf{A}_{st}^T$ is a fixed *k*-by-*k* coefficient matrix, and $\mathbf{I}_k$ is a *k*-by-*k*

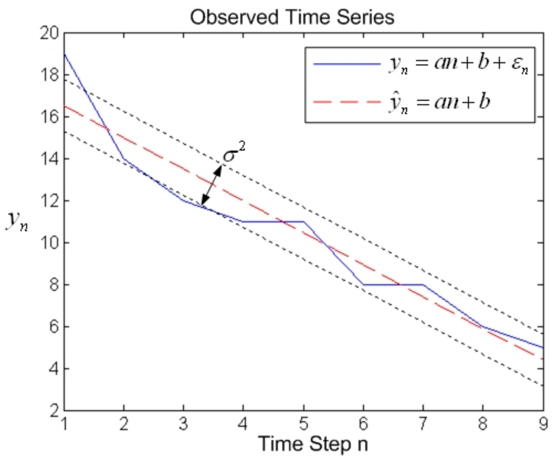


Fig. 4-1: Concept of time series linear-least-squares regression.

identity matrix. Expression (4.20) tells us that the minimum variance residual vector ε^* is the linear transformation of observed data vector Y through a fixed transformation matrix $\mathbf{T}_{st}$. This observation provides us an efficient method for detecting the minimum variance of an observed data sequence in real-time. For instance, let X_1^k, Y_1^k, and D_1^k denote, respectively, the observed data sequence of x_i, y_i, and d_x over time steps 1 to k. Using (4.20), the minimum variances of X_1^k, Y_1^k, and D_1^k are given by

$$\sigma_x^2 = \mathrm{var}(\mathbf{T}_{st} X_1^k),\ \sigma_y^2 = \mathrm{var}(\mathbf{T}_{st} Y_1^k),\text{ and }\ \sigma_d^2 = \mathrm{var}(\mathbf{T}_{st} D_1^k). \tag{4.21}$$

Based on (4.21), the covariance matrix $\mathbf{R}_n$ is updated as

$$\mathbf{R}_n = \mathbf{R}_0 + diag((\sigma_x^2)^2, (\sigma_y^2)^2, (\sigma_d^2)^2), \tag{4.22}$$

where $\mathbf{R}_0$ is the initial covariance matrix of $\mathbf{R}_n$.

4.3.4 Summary of the Proposed VSE-WToV Algorithm

Combining the self-tuning equations (4.21)-(4.22) with Kalman filter equations (4.12)-(4.15), the implemented VSE-WoTV is summarized in Fig. 4-2. The processing steps are listed as follows:

1) Choose two initial covariance matrices $\mathbf{Q}_0$ and $\mathbf{R}_0$.

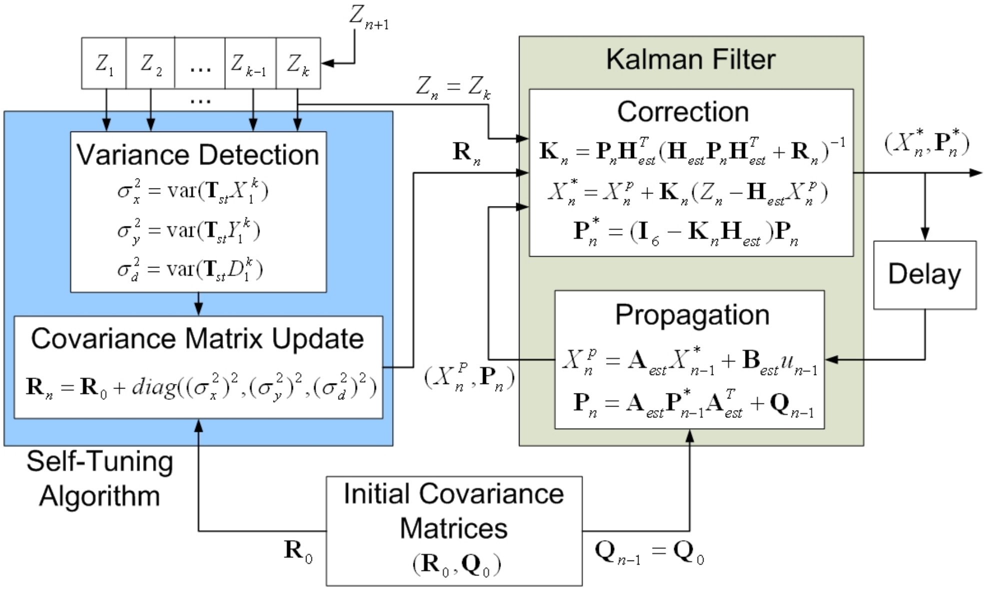

Z_{n+1}
Z_1 Z_2 … Z_{k-1} Z_k
$Z_n = Z_k$
Kalman Filter
Correction
$\mathbf{K}_n = \mathbf{P}_n\mathbf{H}_{est}^T(\mathbf{H}_{est}\mathbf{P}_n\mathbf{H}_{est}^T + \mathbf{R}_n)^{-1}$
$X_n^* = X_n^p + \mathbf{K}_n(Z_n - \mathbf{H}_{est}X_n^p)$
$\mathbf{P}_n^* = (\mathbf{I}_6 - \mathbf{K}_n\mathbf{H}_{est})\mathbf{P}_n$
$(X_n^*, \mathbf{P}_n^*)$
Delay
Variance Detection
$\sigma_x^2 = \mathrm{var}(\mathbf{T}_{st}X_1^k)$
$\sigma_y^2 = \mathrm{var}(\mathbf{T}_{st}Y_1^k)$
$\sigma_d^2 = \mathrm{var}(\mathbf{T}_{st}D_1^k)$
$\mathbf{R}_n$
Covariance Matrix Update
$\mathbf{R}_n = \mathbf{R}_0 + diag((\sigma_x^2)^2, (\sigma_y^2)^2, (\sigma_d^2)^2)$
$(X_n^p, \mathbf{P}_n)$
Propagation
$X_n^p = \mathbf{A}_{est}X_{n-1}^* + \mathbf{B}_{est}u_{n-1}$
$\mathbf{P}_n = \mathbf{A}_{est}\mathbf{P}_{n-1}^*\mathbf{A}_{est}^T + \mathbf{Q}_{n-1}$
Self-Tuning Algorithm
Initial Covariance Matrices
$(\mathbf{R}_0, \mathbf{Q}_0)$
$\mathbf{R}_0$
$\mathbf{Q}_{n-1} = \mathbf{Q}_0$

Fig. 4-2: Architecture of the proposed VSE-WoTV.

2) Assume that the initial position of the target is located in the field-of-view of the camera, then initialize the estimated system state X_0^* and propagation covariance matrix $\mathbf{P}_0$ by the first observation such that $X_0^* = [Z_0^T \quad 0 \quad 0 \quad 0]^T$ and $\mathbf{P}_0 = \mathbf{I}_6$.

3) Store current observed measurement in a shift register with length *k*. If the length of storage data is equal to *k*, then compute the variance of the observed data sequences by (4.21) and update covariance matrix $\mathbf{R}_n$ by (4.22); else set $\mathbf{R}_n = \mathbf{R}_0$; go to step 4.

4) Compute the ideal propagated state X_n^p defined in (4.12) and the corresponding propagation covariance matrix $\mathbf{P}_n$ using (4.13).

5) If the target is detected in the observed image, then compute the Kalman gain matrix $\mathbf{K}_n$ and update the estimated state vector X_n^* with the corresponding covariance matrix $\mathbf{P}_n^*$ using (4.15); else set $X_n^* = X_n^p$ and $\mathbf{P}_n^* = \mathbf{P}_n$; go to step 6.

6) Let $X_{n-1}^* = X_n^*$, $\mathbf{P}_{n-1}^* = \mathbf{P}_n^*$ and $\mathbf{Q}_{n-1} = \mathbf{Q}_0$; go to step 3. ■

4.4 Summary

In this chapter, two novel VSEs (VSE-WTV ans VSE-WoTV) were proposed by using a real-time self-tuning Kalman filter technique. The proposed designs can be applied into several visual tracking applications, such as visual tracking control, visual surveillance, and visual navigation, etc., to estimate the position and the velocity of the target in the image plane. Moreover, the proposed VSEs can estimate the optimal system state from the observation signal with random noise and overcome the temporary occlusion problem during visual tracking process. These advantages provide the visual tracking control system with robustness against the external uncertainties caused by image noise and temporary occlusion. In next chapter, computer simulations and experimental results of tracking a moving target will be presented to validate the estimation performance and robustness of the proposed designs.

Chapter 5

Simulation and Experimental Results

5.1 Introduction

In this chapter, computer simulations and experimental results of the proposed CFA interpolation algorithm, visual tracking control scheme and visual state estimation method are presented and evaluated. In CFA interpolation experiments, the performance of three renowned CFA interpolation methods will be employed to compare with the performance of the proposed CFA interpolation algorithm. Experimental results will show that the proposed method not only outperforms all of them in PSNR (dB) and S-CIELAB ΔE_{ab}^{*} measures, but also gives superior demosaiced fidelities in visual comparison.

In visual tracking control experiments, two experiments will be presented to validate the tracking performance and robustness of the proposed VTC. The first and second experiments evaluate the robustness of the proposed VTC against the velocity quantization uncertainty and system parametric uncertainty, respectively. In visual state estimation experiments, the computer simulations are first presented to verify the robustness of the proposed VSEs against the uncertainty caused by random image noise. The practical experiments are then presented to evaluate the performance of the proposed VSEs to overcome the uncertainty caused by temporary partial/full occlusion.

The rest of this chapter is organized as follows. Section 5.2 evaluates the performance of the proposed CFA interpolation algorithm by quantitative and visual comparison with three recently published methods. Section 5.3 validates the tracking performance and robustness of the proposed VTC design. In Section 5.4, the estimation performance and robustness of the proposed VSEs are presented and examined. Section 5.5 gives the concluding remarks.

5.2 Experimental Results of CFA Interpolation

In the experiments, twenty-five Kodak photographic images as shown in Fig. 2-4 were employed for demonstrating the demosaicing performance of the proposed CFA interpolation algorithm presented in Chapter 2. According to [53], the CFA operations in a digital-camera pipeline usually include a demosaiced image post-processing framework to provide more visually pleasing color output. Therefore, we introduce the post-processing framework in the experiments to complete the comparisons. Figure 5-1 illustrates the flowchart of the experiment, which contains interpolation and post-processing steps. In the interpolation step, the demosaiced results of the proposed method, HPHD color-difference based linear interpolation (HPHD-CDLi) and HPHD-CDEA interpolation methods, are compared with those using bilinear interpolation and three recently published methods: Lu's [3], Gunturk's [5] and Li's [6] methods. The above schemes are chosen due to their high citation rate in peer-reviewed literature [2-8, 40] and represent the state of the technology of CFA interpolation. For Gunturk's method, we make use of one-level (1-L) decomposition with eight projection iterations in the experiments. For Li's method, the universal threshold value $\delta_l = \delta_h = 4$ and maximum iteration number $iter = 20$ are chosen in the experiments. For the proposed method, an experiment of tweaking parameters (N, α) presented in Appendix A was set to find the local optimal parameters for these 25 test images. The local optimal parameters were given by $(N, \alpha) = (11, 0.6)$, which were chosen in the experiments. Subsequently, Lu's post-processing method was adopted as the post-processing procedure for each demosaicing method. The demosaiced results in each step were compared accordingly. As shown in Fig. 2-1, all test images were down-sampled to obtain the Bayer pattern and then reconstructed using the demosaicing methods under comparison in RGB color space.

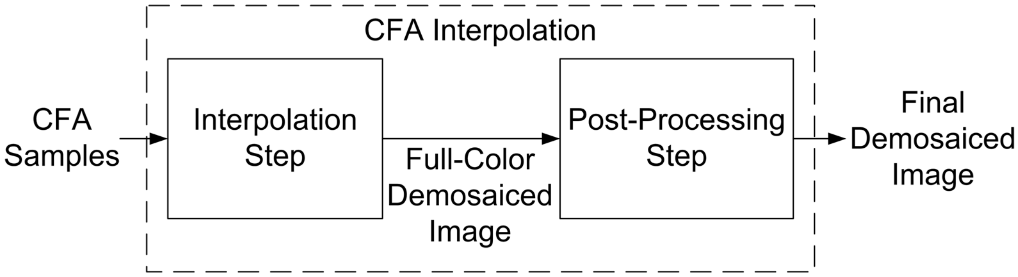


Fig. 5-1: Flowchart of the experiment. In the interpolation step, we compare the performance of bilinear, Lu's, Gunturk's, Li's and proposed HPHD-CDLi, HPHD-CDEA methods. In post-processing step, Lu's post-processing method is adopted into each CFA interpolation method.

Two performance measures were adopted in the experiments: PSNR and S-CIELAB ΔE_{ab}^{*} metric [3, 6, 54] to evaluate the quality of the demosaiced images. The PSNR (in dB) metric in this paper is defined as

$$PSNR(O,D)=10\log_{10}\left\{255^{2}\left(\frac{1}{UV}\sum_{1\le y\le V}\sum_{1\le x\le U}\left\|O(x,y)-D(x,y)\right\|^{2}\right)^{-1}\right\}, \tag{5.1}$$

where U, V are the total column and row number of the image, $O(x,y)$ is the color vector at the $(x,y)^{\text{th}}$ position of the original color image, and $D(x,y)$ is the corresponding color vector in the demosaiced color image. Note that, for a demosaiced image, high fidelity implies high PSNR and small S-CIELAB ΔE_{ab}^{*} measures.

5.2.1 Quantitative Comparison

Table 5-1 records the PSNR values and S-CIELAB ΔE_{ab}^{*} measures of the demosaiced results obtained by the proposed interpolation method together with those from other methods for comparison. In each step, the bold font denotes the highest PSNR and smallest ΔE_{ab}^{*} values across each row. Moreover, since Gunturk's and Li's methods are iterative and others are non-iterative, we categorized these methods into iterative and non-iterative groups for more detailed comparisons. From Table 5-1, one can see that Li's and HPHD-CDEA methods provide improved demosaiced fidelity in most of the test images in the interpolation step. However, when one compares the average PSNR and ΔE_{ab}^{*} measures in the interpolation

Table 5-1: Performance comparison among recent proposed methods: PSNR (dB) and ΔE_{ab}^{*} measures of demosaiced images in the interpolation and post-processing steps.

Step	Interpolation Step						Post-Processing Step					
Group	Iterative		Non-Iterative				Iterative		Non-Iterative			
Method	Gunturk [5]	Li [6]	Bilinear	Lu [3]	HPHD- CDLi	HPHD- CDEA	Gunturk [5]	Li [6]	Bilinear	Lu [3]	HPHD- CDLi	HPHD- CDEA
1	29.3765 1.7666	28.4957 1.8899	24.8843 2.7289	31.0257 1.5357	30.5902 1.6579	**31.2393** **1.5342**	29.2676 1.7845	28.3192 1.9083	28.8878 1.9067	30.7940 1.5466	30.5619 1.6239	**31.0132** **1.5355**
2	33.2296 1.5972	**33.6676** **1.5396**	21.5611 5.1618	31.6889 1.7135	31.3683 1.7700	31.9611 1.6584	33.6595 1.5445	33.9846 1.4974	27.3176 2.8440	33.8433 1.4668	34.0348 1.4397	**34.5002** **1.3893**
3	34.7577 1.6598	35.2213 1.5958	28.3682 3.1280	35.7152 1.4910	35.4998 1.5541	**36.0240** **1.4463**	34.6331 1.6721	35.0579 1.6008	33.2804 1.9659	35.7232 1.4943	36.0086 1.4923	**36.2940** **1.4255**
4	36.6168 0.9774	36.3808 0.9766	29.7242 1.7728	37.3966 0.9094	37.3608 0.9143	**38.0755** **0.8633**	36.7206 0.9635	36.3960 0.9615	35.3870 1.0873	38.0096 0.8576	37.9466 0.8596	**38.5172** **0.8222**
5	34.9839 1.3508	34.8997 1.3260	28.8694 2.5128	**35.4482** 1.3020	34.9540 1.3565	35.2691 **1.3006**	34.9657 1.3075	34.7714 1.3213	34.4611 1.4752	**36.1356** **1.1861**	35.6365 1.2306	35.8899 1.1934
6	32.6411 2.1864	31.8126 2.3790	21.8873 5.7055	32.7081 2.0318	31.8449 2.3001	**32.8136** **2.0001**	32.6069 2.1709	31.6062 2.3857	29.3927 2.8618	33.7802 1.8551	33.4118 1.9444	**33.8814** **1.8254**
7	34.0239 1.2157	33.8198 1.2266	23.0206 3.6536	32.4465 1.2998	33.4723 1.2130	**34.0733** **1.1400**	34.3593 1.1896	34.5397 1.1590	28.6949 2.0249	34.0965 1.1592	35.6736 1.0243	**36.0181** **0.9950**
8	36.8763 1.1338	36.7725 1.1444	28.7405 2.1614	37.9098 0.9885	36.8668 1.0789	**37.9580** **0.9662**	36.6670 1.1576	36.4265 1.1699	35.3671 1.2509	**38.1854** 0.9694	37.5005 1.0442	38.1763 **0.9687**
9	30.8332 1.7679	**31.2495** 1.7192	18.7578 5.9022	29.7212 1.8327	29.2916 1.9456	30.5677 **1.6909**	31.1581 1.7214	31.4196 1.6968	23.9221 3.4143	31.3071 1.6277	31.3034 1.6376	**32.3919** **1.4892**
10	36.7662 0.8925	37.2501 0.8255	27.5750 1.8783	36.8133 0.8758	37.0465 0.8258	**37.5485** **0.8032**	37.0662 0.8491	37.2927 0.8226	33.3472 1.1263	37.8106 0.7919	38.3269 0.7465	**38.5816** **0.7375**
11	36.7975 0.8954	**37.0956** **0.8286**	27.6351 1.8566	36.8098 0.8715	36.5621 0.8716	37.0535 0.8417	37.0497 0.8536	37.0952 0.8263	34.1833 1.0627	37.5213 0.7926	37.7538 0.7828	**37.9091** **0.7717**
12	**34.5407** 1.4748	34.4102 1.4275	24.3568 3.8112	33.8725 1.4666	33.6367 1.5243	34.3895 **1.3974**	34.6820 1.4288	34.7541 1.3622	30.3304 2.1498	35.2610 1.3140	35.4407 1.3067	**35.9223** **1.2380**
13	37.8205 0.6731	37.7569 0.6760	28.7032 1.5011	37.3884 0.6695	37.7280 0.6527	**38.2135** **0.6266**	37.9377 0.6665	37.8628 0.6610	34.1471 0.8955	38.3279 0.6267	38.9659 0.5971	**39.2299** **0.5838**
14	29.7386 2.5595	**30.4264** **2.4457**	19.0903 7.1986	27.8600 2.4652	27.5973 2.9558	28.0727 2.7885	30.2466 2.8077	**30.8242** 2.3680	25.6167 3.7864	30.2549 2.3619	30.3762 2.3839	30.7444 **2.3128**
15	30.8370 1.9406	29.6090 2.1114	24.4266 4.1290	32.4833 1.7491	31.7108 1.8908	**32.6677** **1.6838**	30.6644 1.9284	29.3860 2.1159	30.1370 2.3280	32.6128 1.6518	31.9835 1.7319	**32.6813** **1.5960**
16	34.4301 1.4764	34.3050 1.4804	28.2748 2.5666	34.4161 1.3868	34.5631 1.4168	**34.7185** **1.3640**	34.3523 1.4682	34.2067 1.4704	33.7516 1.5735	34.9354 1.3388	35.1211 1.3436	**35.1681** **1.3196**
17	37.3602 0.9964	37.0917 1.0009	26.5250 2.8594	35.6650 1.0971	37.1741 0.9836	**37.7084** **0.9431**	37.6885 0.9740	37.8239 0.9477	31.8934 1.6581	37.2329 0.9865	39.1877 0.8509	**39.5172** **0.8314**
18	36.2947 1.4628	**36.4685** **1.3340**	27.2295 2.8610	35.7449 1.4857	35.4640 1.5146	35.9088 1.4615	36.5932 1.3572	36.6429 1.3100	33.7658 1.6517	36.8960 1.3056	36.9038 1.3098	**37.0386** **1.2863**
19	**32.3393** 2.3592	32.3295 2.3903	23.3178 5.0440	31.6767 2.2879	31.3516 2.3566	31.7392 **2.2609**	32.5119 2.3137	32.2416 2.4326	29.5746 2.8095	32.9921 **2.0898**	32.8655 2.1323	**33.0480** 2.0906
20	34.9738 1.3061	35.2707 **1.2493**	23.1302 3.6480	34.5020 1.3409	34.6201 1.3254	**35.3463** 1.2559	35.2671 1.2452	35.5570 1.2094	28.4598 2.0810	35.7424 1.1902	36.1790 1.1547	**36.6761** **1.1201**
21	35.7991 1.0396	35.7714 1.0294	26.8448 2.1996	35.8899 1.0016	35.1749 1.0650	**35.9477** **0.9915**	36.0108 1.0077	35.9894 0.9971	33.1566 1.2790	36.8055 0.9230	36.5337 0.9458	**36.9950** **0.9086**
22	**34.0980** **1.3142**	33.8535 1.3468	23.7154 3.4864	33.0809 1.3691	32.5035 1.4679	33.2871 1.3564	34.3656 1.2900	34.4198 1.2757	29.6633 1.9425	34.6893 1.2138	34.4820 1.2435	**35.0812** **1.1863**
23	32.8830 1.5024	32.9540 1.5250	25.5756 2.9269	**33.5303** 1.3922	32.9888 1.4540	33.4476 **1.3894**	32.8127 1.5307	32.8188 1.5364	30.7133 1.7699	**33.7291** **1.3651**	33.4516 1.4050	33.6982 1.3690
24	37.0203 0.9664	37.0820 0.9820	30.3178 1.4579	38.0689 0.8977	37.9213 0.9063	**38.4486** **0.8702**	36.9022 0.9867	36.9148 0.9956	36.3310 0.9942	38.1993 0.8945	38.2472 0.8976	**38.6336** **0.8706**
25	29.8870 1.4933	**30.0755** 1.5055	21.9902 3.6050	29.4449 1.4432	29.7334 1.5206	29.8614 **1.4261**	30.0602 1.4858	30.0909 1.5123	27.8193 1.9570	30.0984 1.3599	**30.7241** 1.3714	30.5972 **1.3405**
Avg.	34.1970 1.4403	34.1628 1.4382	25.3809 3.3503	34.0523 1.4099	33.8810 1.4609	**34.4936** **1.3624**	34.3300 1.4145	34.2577 1.4217	31.1840 1.9158	34.9994 1.2948	35.1448 1.3000	**35.5282** **1.2483**
Add-up in Average							0.1330 -0.0258	0.0949 -0.0165	**5.8031** **-1.4345**	0.9471 -0.1151	1.2638 -0.1609	1.0346 -0.1141

step, HPHD-CDEA generates the highest fidelity demosaiced images, followed by the Gunturk's or other methods.

In the post-processing step, Table 5-1 indicates an interesting phenomenon that all non-iterative methods have significant improvement compared with iterative ones, especially the bilinear interpolation (BI). In average, the improvement of BI can add-up the PSNR and reduce ΔE_{ab}^{*} of the interpolation results by 5.8031dB and 1.4345 units, respectively. The other non-iterative methods also have noticeable improvement in average. In contrast, the iterative methods, e.g. Gunturk's and Li's methods, only have modest improvement through the post-processing step in average. These observations also can be seen in [53], where the post-processing step provides the most significant improvement with BI and the smallest improvement with Gunturk's method. Therefore, the experimental results presented in Table 5-1 as well as [53] pose a question why post-processing is more beneficial to the interpolation results of non-iterative approaches compared to that of iterative ones. The main reasons are as follows.

Many CFA interpolation schemes, especially the simple ones such as BI or HPHD-CDLi, usually induce visible artifacts due to the non-smooth local color ratios and color differences (red-green and blue-green). The function of current post-processing schemes is to correct the interpolated color values by enforcing the local color ratio rule [11, 53] and color difference rule [3] of initial demosaiced image. Similarly, the principle of iterative demosaicing approaches [5, 6] is to iteratively update the initial interpolation result by fitting the local color difference rule. For example, according to [8], the idea of Gunturk's iterative method is equivalent to the filtering of down sampled color difference images of the initial interpolated image by a 5x5 2-D low-pass filter for reducing the high frequency energy of reconstructed color difference images without changing original mosaic samples. In [6], Li utilized the Hamilton-Adams' method [55] and BI to get initial estimates of missing green and red/blue samples, respectively. The following iterative procedure is equivalent to linear low-pass filtering of the color difference image until the reconstructed results converge to a smooth one. In other words, the iterative demosaicing approaches can be regarded as an initial interpolation combined with a meta-algorithm that performs iterative linear low-pass filtering of color difference images to enforce the local color difference rule on initial interpolated image, which is also the main purpose of the latter post-processing step. Therefore, the post-processing only provides modest improvement for iterative approaches.

Summarizing the above discussion on the experimental results, we have the following conclusions:

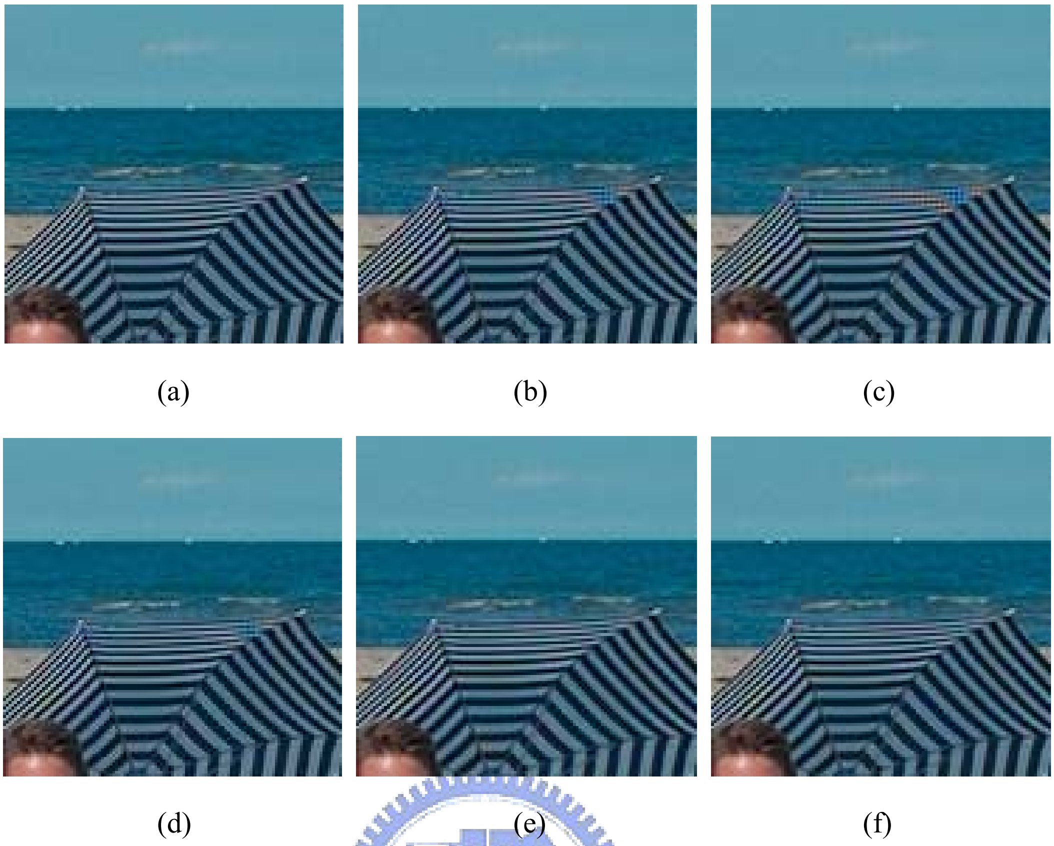
1) For iterative approaches, post-processing only provides the modest improvement due to both have the same purpose of enforcing the local color difference rule on the initial demosaiced image.

2) On the contrary, post-processing for non-iterative approaches, especially simple linear interpolation schemes such as BI or HPHD-CDLi schemes, provides significant improvement due to its enforcing on the smoothness of local color ratios and color differences.

3) Because the proposed HPHD-CDEA scheme is non-iterative and provides the best interpolation results in interpolation step, it also has great improvement and obtains the best results after the post-processing step.

5.2.2 Visual Comparison

Figures 5-2(a) and 5-3(a) show the zoom-in of test image No. 1 and 20, respectively. These scenes contain many fine detail features, such as fine fiber patterns (Fig. 5-2) and picket fences (Fig. 5-3), and can effectively challenge the performance of demosaicing methods. Figures 5-2(b), 5-2(c), 5-2(d), 5-2(e) and 5-2(f) are the demosaiced results obtained from Gunturk's, Lu's, Li's, HPHD-CDLi and HPHD-CDEA methods in the interpolation step, respectively. From visual comparison, one can see that the Gunturk's, Lu's and Li's methods induce more color artifacts in edge and textured regions than HPHD-CDLi or HPHD-CDEA does. Figures 5-3(b), 5-3(c), 5-3(d), 5-3(e) and 5-3(f) also show the similar comparison results. These experimental results validate that the proposed HPHD CFA interpolation method



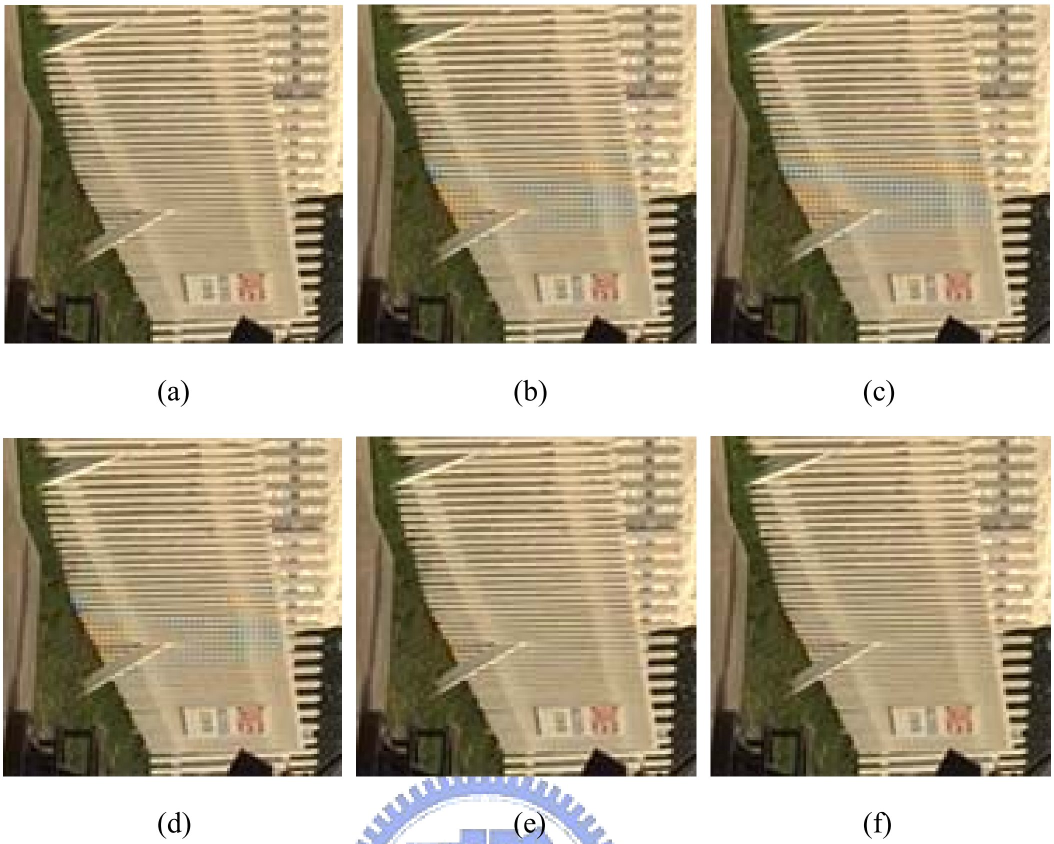
(a) (b) (c)

(d) (e) (f)

Fig. 5-2: Zoom-in demosaicing results of test image No. 1. (a) Original picture; Demosaiced result in interpolation step: (b) Gunturk's method, (c) Lu's method, (d) Li's method, (e) HPHD-CDLi method, and (f) HPHD-CDEA method.

performs satisfactorily not only in textured regions, but also in well-defined edges. More discussions and visual comparisons are presented in Appendixes B and C.

Furthermore, as can be seen in Figs. 5-2 and 5-3, HPHD-CDLi gives almost the same demosaiced results in edge and textured regions compared with HPHD-CDEA does. Hence, HPHD-CDLi can use instead of HPHD-CDEA in practical applications for HPHD-CDLi not only saves a great amount of computational cost, but also gives comparable visual results as HPHD-CDEA.



(a) (b) (c)

(d) (e) (f)

Fig. 5-3: Zoom-in demosaicing results of test image No. 20. (a) Original picture; Demosaiced result in interpolation step: (b) Gunturk’s method, (c) Lu’s method, (d) Li’s method, (e) HPHD-CDLi method, and (f) HPHD-CDEA method.

5.2.3 Computational Complexity

The calculation performed in reconstructing one color pixel in each stage of the algorithm are listed in Table 5-2, where N and α denote the parameter of window size and spatial classification, respectively. For two directional heterogeneity-projections (H.P.), (2.12) and (2.13) require a total of $2N$-2 additions, $2N$ multiplications and 2 absolute conversions for each color pixel. In the directional adaptive filtering (D.A.F) stage, if a $1\times W$ rectangular window was used to compute the local mean and variance by (2.15) and (2.16), then the total calculation of (2.17) and (2.18) needs $12W$-2 additions and $4W$+12 multiplications. In the hard-decision CFA interpolation (H.D.C.I), the total calculation of interpolation with $\alpha = 0$ and $\alpha = 1$ requires the maximum and minimum computation for each color pixel, respectively. Therefore, if $0 < \alpha < 1$, the total computational load of interpolation will be

Table 5-2: Calculations performed for reconstructing one color pixel.

		HPHD-CDEA					HPHD-CDLi				
Stage		ADDSs	MULTSs	ABSs	SHIFTs	COMPs	ADDSs	MULTSs	ABSs	SHIFTs	COMPs
H.P.		2*N*-2	2*N*	2	0	0	2*N*-2	2*N*	2	0	0
D.A.F.		12*W*-2	4*W*+12	0	0	0	12*W*-2	4*W*+12	0	0	0
H.D.C.I.	$\alpha=1$	59.5	17	13	16	1.5	10.5	0	0	3.5	1.5
	$\alpha=0$	92.5	28	22	28	3	17.5	0	0	6	3
Total	$\alpha=1$	2*N*+12*W* +55.5	2*N*+4*W*+29	15	16	1.5	2*N*+12*W* +6.5	2*N*+4*W*+12	2	3.5	1.5
	$\alpha=0$	2*N*+12*W* +88.5	2*N*+4*W*+40	24	28	3	2*N*+12*W* +13.5	2*N*+4*W*+12	2	6	3

ADDSs: Additions; MULTSs: Multiplications; ABSs: Absolute conversions;

SHIFTs: Bit-shift operations; COMPs: Compare operations.

Table 5-3: Computational cost per color pixel of different CFA interpolation techniques.

		ADDSs	MULTSs	ABSs	SHIFTs	COMPs	LUTs	OPs
Bilinear Interpolation (BI)		4	0	0	3	0	0	7
Lu's method [3]		81	16	29	18	0	0	144
Gunturk's method [5]		400	384	4	4.5	1	0	793.5
Hirakawa's method [7]		106	50	12	0	103	6	277
HPHD-CDEA (*N*=11, *W*=9)	$\alpha=1$	185.5	87	15	16	1.5	0	305
	$\alpha=0$	218.5	98	24	28	3	0	371.5
HPHD-CDLi (*N*=11, *W*=9)	$\alpha=1$	136.5	70	2	3.5	1.5	0	213.5
	$\alpha=0$	143.5	70	2	6	3	0	224.5
HPHD-CDEA (*N*=9, *W*=7)	$\alpha=1$	157.5	75	15	16	1.5	0	265
	$\alpha=0$	190.5	86	24	28	3	0	331.5
HPHD-CDLi (*N*=9, *W*=7)	$\alpha=1$	108.5	58	2	3.5	1.5	0	173.5
	$\alpha=0$	115.5	58	2	6	3	0	184.5

LUTs: Look-up tables; OPs: Operations;

between that with $\alpha=1$ and $\alpha=0$.

Table 5-3 tabulates the computational cost of the proposed algorithm with those of the methods published in [3], [5] and [7]. Although BI and Lu's methods require fewer operations in interpolation step, they will induce more color artifacts in edge and textured regions of the demosaiced results. Gunturk's method provides better demosaicing results; however, it requires more operations in interpolation step. Hirakawa's method represents the state of the technology of decision-based CFA interpolation. In his method, the latter decision stage requires much more computation compared with the interpolation stage, i.e. it requires 103 compare operations for choosing the best interpolation result from two directional interpolations. Moreover, if the interpolation stage includes a smooth interpolation step, the calculation of decision stage will increase greatly, because it will need to evaluate three interpolation results for each color pixel. In contrast, the calculation of the proposed hard-decision method depends only on the parameters of window size, N and W. The evaluation of horizontal, vertical and smooth interpolations depends on the parameter α and only needs at most 3 compare operations for each color pixel. From Table 5-3, it is clear that the computational operations can be reduced efficiently by combining linear interpolation with the proposed HPHD algorithm. Therefore, the proposed method provides an efficient solution for decision-based CFA interpolation techniques.

***Remark 5.1*:** Another advantage of the proposed HPHD demosaicing algorithm is that the algorithm is parallelizable and is faster on dual core processors. For instance, an application program of raw photo editing, RawTherapee [56], uses the proposed HPHD algorithm as the default demosaicing method, because HPHD not only has better resolution compared to the existing algorithms, but also is at least twice as fast compared with the enhanced adaptive homogeneity-directed (EAHD) demosaicing algorithm [7]. Please see [56] for more details.

5.3 Simulation and Experimental Results of Visual Tracking Control

In order to verify the performance of the proposed controller design, the target's motion velocity is supposed to be known *a-priori* in the visual tracking control experiments. To do so, we set a moving robot as the target with *a-priori* known motion velocity and combine the VSE-WTV presented in Section 4.2 with the proposed VTC to estimate the optimal target state in the image plane. Figure 5-4 depicts two experimental mobile robots developed in our Lab for the study of mobile robot visual tracking control. In Fig. 5-4, left robot (called *tracking robot*) is equipped with a USB camera and a tilt camera platform to track another robot on which a green, cylindrical target of interest was installed (called *target robot*). Figure

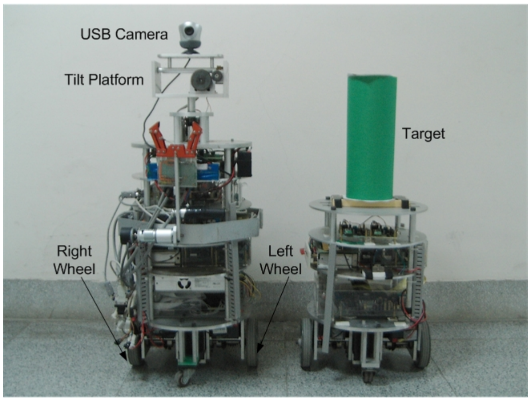


Fig. 5-4: Two experimental mobile robots developed in the Intelligent System Control Integration (ISCI) Lab, National Chiao Tung University.

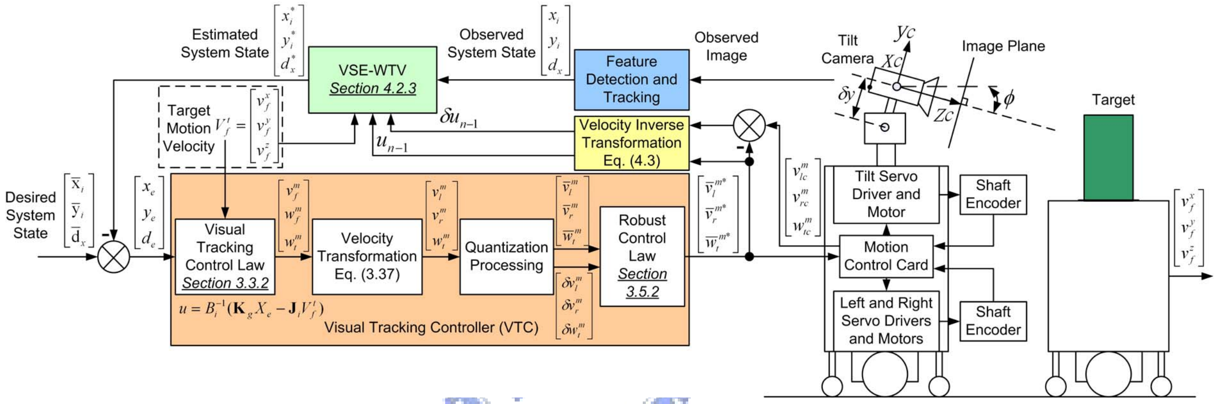


Fig. 5-5: Implemented VTC (presented in Chapter 3) combined with the VSE-WTV (presented in Section 4.2) to test in the visual tracking control experiments.

5-5 illustrates the block diagram of the visual tracking control system, which combines the VTC with the VSE-WTV to test in the visual tracking control experiments. The function of each block shown in Fig. 5-5 is listed below:

1) Feature detection and tracking: perform image-processing algorithm to extract and track the observed system state $[x_i \quad y_i \quad d_x]^T$ in the image captured from the camera.
2) Visual state estimator with target velocity (VSE-WTV): estimate the optimal system state $[x_i^* \quad y_i^* \quad d_x^*]^T$ in the image plane. The performance of the VSE-WTV will be verified in Section 5.4.2.

3) Visual tracking control law: compute the desired velocity control commands $\begin{bmatrix} v_f^m & w_f^m & w_t^m \end{bmatrix}^T$ using (3.20-1).

4) Velocity transformation: transform the desired velocities into desired left- and right-wheel velocity commands using (3.44).

5) Quantization processing: quantize the desired velocity control commands dependent on the resolution of motion control module. The resolution of the self-made motion control card used in the experiments is 8-bit, which means it can command the linear wheel velocity from -128 to127 cm/s in integer. For example, suppose that the desired left-wheel velocity command v_l^m is 2.9925 cm/s. After quantization processing, the quantized velocity command $\bar{v}_l^m$ is $\lfloor 2.9925 \rfloor = 2$ cm/s, where $\lfloor x \rfloor$ is the largest integer smaller than x, and the corresponding quantization error δv_l^m is $\bar{v}_l^m - v_l^m = -0.9925$ cm/s. In other words, the maximum velocity quantization error of left- and right-wheel velocity commands $\begin{bmatrix} \delta v_l^m & \delta v_r^m \end{bmatrix}^T$ is smaller than ± 1 cm/s in the experiment. Similarly, the maximum quantization error of tilt velocity command δw_t^m is smaller than ± 0.028 rad/s. Thus, the three positive constants defined in (3.47) can be determined such that $(\varepsilon_l, \varepsilon_r, \varepsilon_t) = (1, 1, 0.028)$.

6) Robust control law: compute robust velocity control commands $\begin{bmatrix} \bar{v}_l^{m*} & \bar{v}_r^{m*} & \bar{w}_t^{m*} \end{bmatrix}^T$ to overcome the velocity quantization uncertainty using (3.46).

7) Motion control card: where the velocity servo loop is implemented. It also estimates robot's current left- and right-wheel velocities $\begin{bmatrix} v_{lc}^m & v_{rc}^m & w_{tc}^m \end{bmatrix}^T$ for the VSE-WTV to calculate the error covariance matrix of the propagated system state (see Section 4.2 for details).

8) Velocity inverse transformation: transform the estimated left- and right-wheel velocities

into estimated linear and angular velocities of the robot using (4.3).

One computer simulation and two practical experiments have been carried out to validate the performance of the proposed control schemes: computer simulation and experiment 1 aim to validate to robustness against the uncertainty of velocity quantization error, and experiment 2 is to verify the result of Theorem 3.2. In computer simulation and experiment 1, the target is static ($v_f^t(t) = 0$ for any $t \geq 0$). Experiment 2 considers the situation that the target is always moving ($v_f^t(t) \neq 0$ for any $t \geq 0$). The parameters used in simulation and both experiments are listed in Table 5-4. Note that different control gains were used in experiment 2 in order to verify Theorem 3.2. Because the target is static in experiment 1, this leads (3.13) to be simplified such that

$$\dot{X}_e = -\mathbf{B}_i u \, . \tag{5.2}$$

This implies the parametric uncertainties encountering in visual tracking system are only caused by matrix $\delta\mathbf{B}_i$. In contrast, the visual tracking system in experiment 2 will be influenced by the parametric uncertainties of matrices $\delta\mathbf{A}_i$, $\delta\mathbf{B}_i$ and vector δC_i. Hence the parametric uncertainties encountered in experiment 2 are larger than that in experiment 1. This fact will be shown in the experimental results discussed below. Based on Theorem 3.2, we note that the larger parameters $(\alpha_1, \alpha_2, \alpha_3) > 0$, the better the robust performance against parametric uncertainties in the system. Therefore, the control gains used in experiment 2 are larger than that used in experiment 1 in order to overcome the larger parametric uncertainties in the experiment. Moreover, the experimental mobile robots have a limitation on motor speed and hence the wheel speed (≤ 25 cm/s) for safety reason. In experiment 1, the controller usually generates control signals larger than 25 cm/s. In order to guarantee the controller's velocity commands to satisfy this speed limitation, the control gains $(\alpha_1, \alpha_2, \alpha_3)_S$ used in experiment 1 is quite small; consequently, the quantization effect will be increased. Therefore,

Table 5-4: Parameters used in the simulation and experiment of visual tracking of a moving robot.

Symbol	Quantity	Description
$(\mathrm{f}_x,\mathrm{f}_y)$	(294,312) pixels	Camera focal length in retinal coordinates.
W	12 cm	Width of the target.
D	30 cm	Distance between two drive wheels.
δy	10 cm	Distance between the center of robot tilt platform and the onboard camera
T	100 ms	Sampling period of the control system
$(\bar{\mathrm{x}}_i,\bar{\mathrm{y}}_i,\bar{\mathrm{d}}_x)$	(0,0,35)	Desired system state in the image plane.
$(\alpha_1,\alpha_2,\alpha_3)_S$	(5/64,6/16,4/64)	Control gains used in the first and second experiments
$(\alpha_1,\alpha_2,\alpha_3)_L$	(5/16,6/8,4/16)	Control gains used in the second experiment
$(\varepsilon_l,\varepsilon_r,\varepsilon_t)$	(1,1,0.028)	Max. quantization errors
$(z_f^m,x_f^m,\theta_f^m,\phi)\|_{t=0}$	(0,0,0,0)	Initial pose of tracking robot.

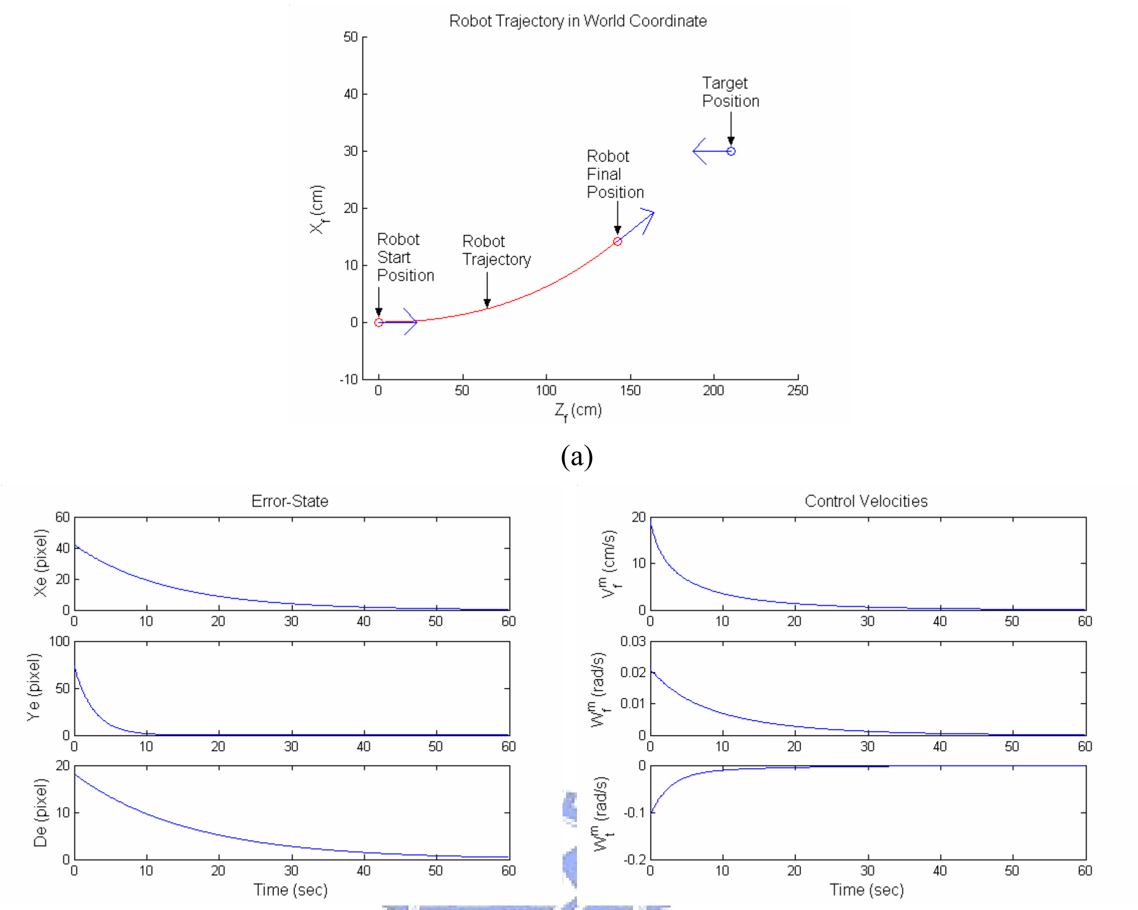
the proposed robust control law (3.46) plays an important role to overcome this problem and guarantee the asymptotic stability of the closed-loop system with a small gain controller.

5.3.1 Computer Simulations

A simulation environment is setup using MATLAB to evaluate the tracking performance of the proposed VTC. Figure 5-6 presents the computer simulation results of the stationarytarget case without velocity quantization error. The initial pose of the tracking robot is $(0\,\mathrm{cm}, 0\,\mathrm{cm}, 0)$, and that of the target robot is $(210\,\mathrm{cm}, 30\,\mathrm{cm}, \pi)$. Figure 5-6(a) shows the robot trajectory in the world coordinate system. Figure 5-6(b) illustrates the tracking errors in the image plane. The simulation results show that the tracking errors decay exponentially to zero using the proposed control law and hence the robot can track the target smoothly. In Table 5-4, we set $(\alpha_1,\alpha_2,\alpha_3)_S$ equal to (5/64, 6/16, 4/64). Thus, using (3.18), we have

$$X_e(t) = diag(-\frac{5}{64}t,-\frac{6}{16}t,-\frac{4}{64}t)X_e(0) \quad \text{for} \quad t_0 = 0\,. \qquad (5.3)$$

The tracking error y_e will decay to zero faster than x_e and d_e. The simulation shown in Fig. 5-6(b) validates this result. Figure 5-6(c) depicts the control velocities of the robot center



(a)

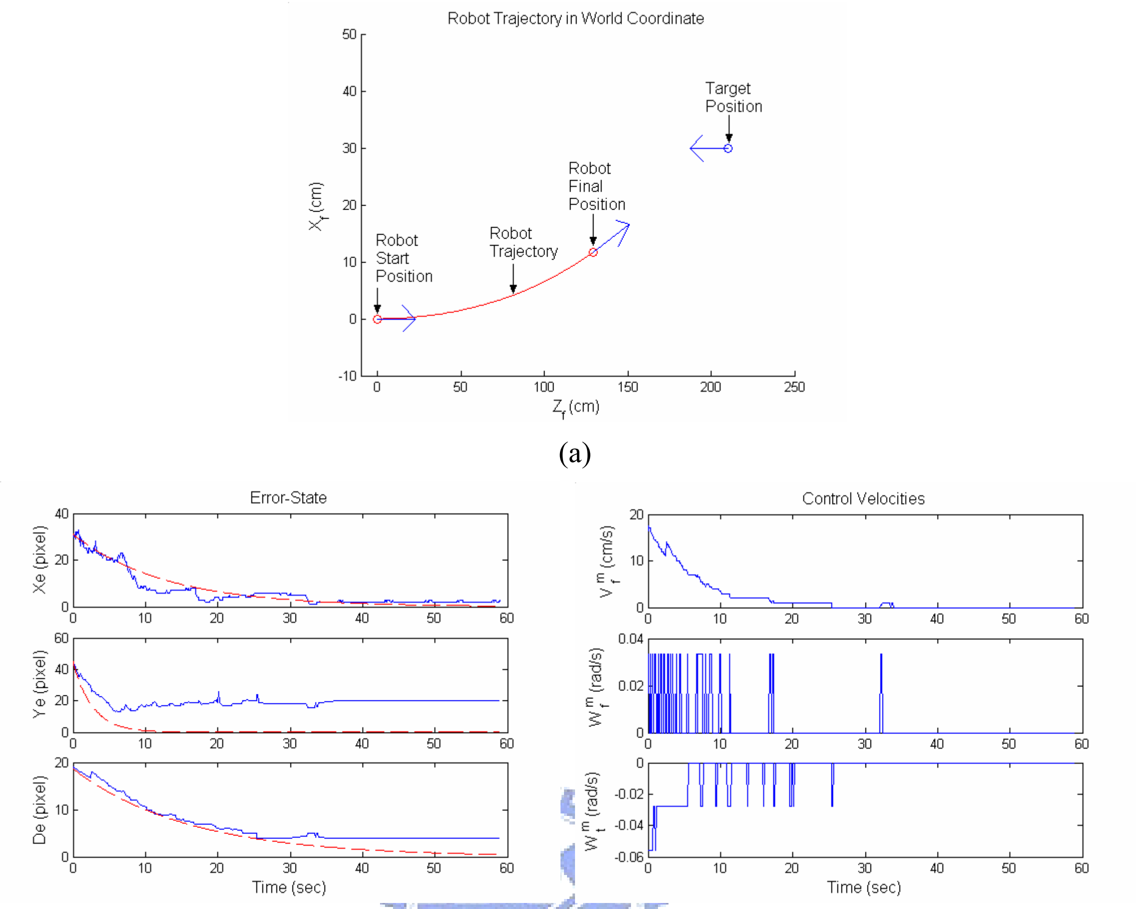
(b) (c)

Fig. 5-6: Computer simulation results of experiment 1 without velocity quantization error. (a) Robot trajectory in the world coordinates. (b) Tracking errors in the image plane. (c) Control velocities of the center point and tilt camera of tracking robot.

point and the tilt camera of the tracking robot. It shows that the control signals also achieve exponential convergence. Therefore, the simulation results validate the result of Theorem 3.1.

5.3.2 Experiment 1: Robustness to Velocity Quantization Error

In this experiment, the proposed VTC is implemented on the tracking robot in order to validate the performance of the proposed robust control law (3.46). Figure 5-7 shows practical experimental results of this case without using the robust control law (3.46). Figure 5-7(a) illustrates the robot trajectory in the world coordinates. Figure 5-7(b) indicates the tracking errors, in which the dotted lines are the theoretical values of (5.3) and the solid lines are the



(a)

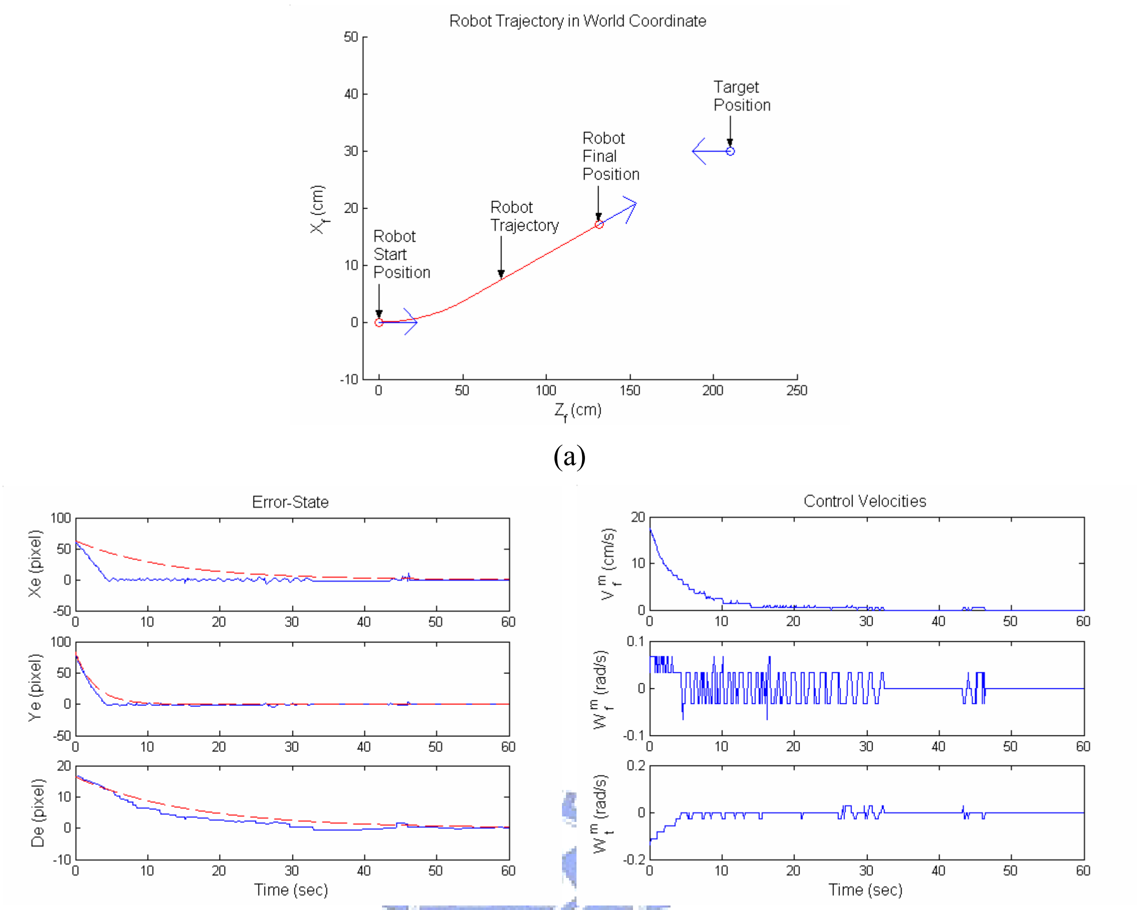
(b)

(c)

Fig. 5-7: Experimental results of experiment 1 without using the proposed robust control law (3.46). (a) Robot trajectory in the world coordinates. (b) Tracking errors in the image plane. (c) Control velocities of the center point and tilt camera of tracking robot.

experimental results of tracking errors. From Fig. 5-7(b), we see that the system states in the practical experiment do not converge to the desired states. This is caused by the system quantization error in the velocity commands, which degrades the performance of the controller. Figure 5-7(c) shows the control velocities of the tracking robot. In Fig. 5-7(c), it can be seen that when the angular velocity was smaller than $0.04\ \mathrm{rad/s}$, the quantization error greatly affected the actual angular velocity. Therefore, the system state cannot converge to the desired state as desired.

We repeated then the experiment by combining the proposed robust control law (3.46) with the VTC. Figure 5-8 presents the experimental results using the proposed robust control



(a)

(b)

(c)

Fig. 5-8: Experimental results of experiment 1 using the proposed robust control law (3.46). (a) Robot trajectory in the world coordinates. (b) Tracking errors in the image plane. (c) Control velocities of the center point and tilt camera of tracking robot.

law. Figure 5-8(a) depicts the robot trajectory in world coordinates. Figure 5-8(b) indicates the tracking errors in the image plane. We observe that the convergence rate of experimental results is faster than that of (5.3). This verifies that the proposed robust control law not only guarantees the tracking errors to decay to zero asymptotically but also increases the convergence rate in the practical system. Moreover, the experimental results also verify the result of Theorem 3.2 that the closed-loop control system (3.13) under the control law given in (3.20) is asymptotically stable at the origin when $\left\|V_f^t\right\| = 0$ satisfies. Figure 5-8(c) shows the control velocities of the tracking robot. A video clip of experimental 1 is available online in [57].

Remark 5.2: Although quantization errors are analyzable using techniques developed for finite word-length controllers [58, 59], the reported methods are only useful for the linear time-invariant (LTI) systems but not suitable for the Quasi-LPV systems. On the contrary, the proposed control method can be applied to both LTI and Quasi-LPV systems. This is the merit of the proposed method. On the other hand, using a more responsive controller with high control gain is an alternative to reduce the effect of quantization error; however, it usually generates large control outputs. In many applications in human-centered service robotics, one expects relatively smooth motion for safety considerations. The proposed control method with smaller control gain guarantees that the equilibrium point of practical system is also asymptotically stable.

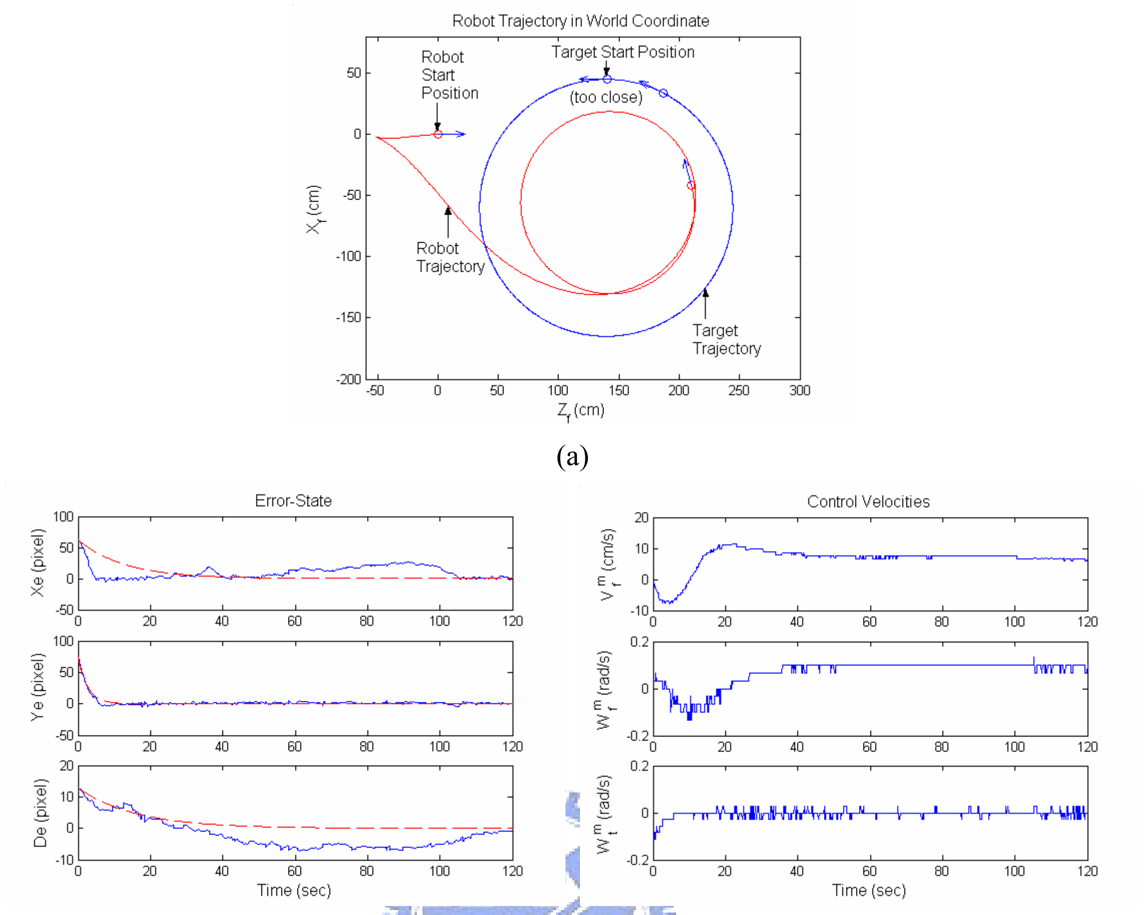
5.3.3 Experiment 2: Robustness to System Model Uncertainty

In this experiment, the target is always moving in order to validate the parametric robustness of the proposed VTC. The initial pose of the target robot are set as $(140\,\mathrm{cm}, 30\,\mathrm{cm}, \pi)$. The target robot is moving along a counterclockwise circular path with velocity

$$(v_f^x, v_f^y, v_f^z) = (v_f^t \sin\theta_f^t, 0, v_f^t \cos\theta_f^t)\,, \tag{5.4}$$

where $v_f^t = 10.5\,\mathrm{cm/sec}$ and $\theta_f^t(new) = \theta_f^t(old) + 0.01\,\mathrm{rad}$ with $\theta_f^t(0) = \pi$. The target velocity information is then used for the VSE-WTV to estimate the state of the target and overcome the occlusion problem even the target is fully occluded. The experimental results of the VSE-WTV will be presented in Section 5.4.2. The initial pose of the tracking robot is also $(0\,\mathrm{cm}, 0\,\mathrm{cm}, 0)$.

In order to validate the result of Theorem 3.2, small control gains $(\alpha_1, \alpha_2, \alpha_3)_S$ were first applied for the control law (3.20), which implies a small robustness against the system parametric uncertainties. Later, larger control gains $(\alpha_1, \alpha_2, \alpha_3)_L$ were used in order to



(a)

(b)

(c)

Fig. 5-9: Experimental results of experiment 2 using control gains $(\alpha_1,\alpha_2,\alpha_3)_S$ = (5/64,6/16,4/64). (a) Robot trajectory in the world coordinates. (b) Tracking errors in the image plane. (c) Control velocities of the center point and tilt camera of tracking robot.

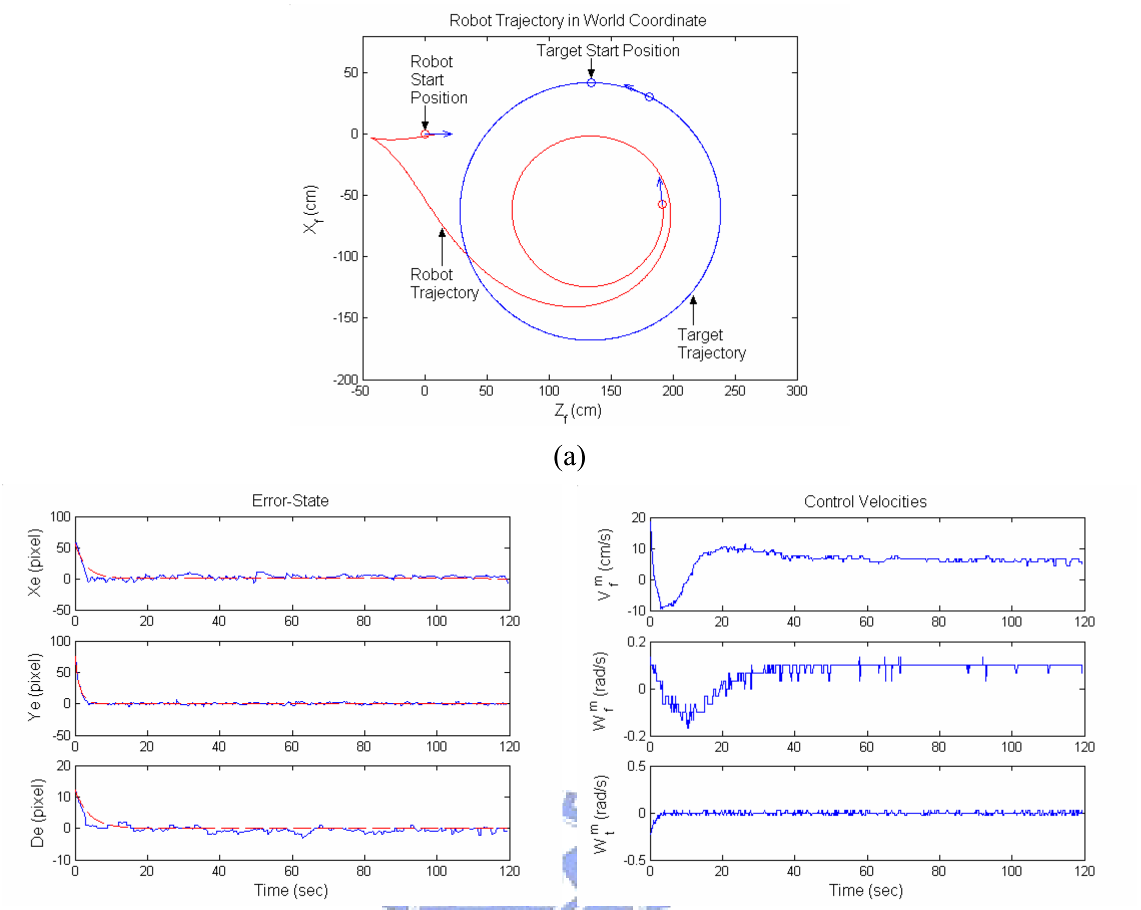
increase the robustness and improve the tracking performance of the closed-loop visual tracking system.

Figure 5-9 presents the recorded responses of this experiment using the same control gains as in experiment 1. In Fig. 5-9(a), the trajectories of two robots were recorded in the world coordinates. Because these two robots were set face to face in the initial condition, the tracking robot moved backward in the beginning and then moved forward to track the target robot. Figure 5-9(b) depicts the tracking errors in the image plane. In Fig. 5-9(b), the dotted lines illustrate the theoretical result from (5.3) while the solid lines show the experimental results of tracking errors. From Fig. 5-9(a), we observe that the tracking robot followed the target robot with a poor behavior, which also can be seen in Fig. 5-9(b). In Fig. 5-9(b), the error states x_e and d_e in the experiment do not converge to zero asymptotically. The main reason is explained in the following. Recall Theorem 3.2 in Section 3.4, the closed-loop visual tracking system is practically stable only for $\max(\alpha_1,\alpha_2,\alpha_3)_S = 6/16 >> C_N$ and $\min(\alpha_1,\alpha_2,\alpha_3)_S / \max(\alpha_1,\alpha_2,\alpha_3)_S = 1/6 > \left\| \delta\mathbf{B}_i \overline{\mathbf{B}}_i \right\|$ in the experiment. This implies that the proposed controller provided a modest robustness against the parametric uncertainties. This result can be observed in Fig. 5-9(b) which indicates that d_e has a poor convergence performance due to the system parametric uncertainties.

Based on Theorem 3.2, the proposed controller with larger control gains provides more robust property w.r.t. the system parametric uncertainties. Hence larger control gains $(\alpha_1,\alpha_2,\alpha_3)_L$ as listed in Table 5-4 were used to repeat experiment 2. Figure 5-10 presents the recorded responses of this experiment. Similarly, the tracking robot also moved backward in the beginning and then moved forward to track the target robot as shown in Fig 5-10(a). However, the tracking robot tracked the target robot in a circular motion with improved performance compared with that of Fig. 5-9. Figure 5-10(b) depicts the tracking errors in the image plane. We observe that the system state in the experiment converges practically to the desired state as expected. These experimental results verify Theorem 3.2 and the robust control law (3.46) as well. A video clip of experiment 2 can be seen online in [57].

5.4 Simulation and Experimental Results of Visual State Estimation

In this section, the computer simulations and practical experiments are presented to validate the estimation performance and robustness of the proposed VSE-WTV and VSE-WoTV described in Chapter 4. First, MATLAB was used to study the estimation performance of the proposed VSE-WoTV. Next, two experiments were performed on the experimental mobile robots to validate the robustness against the occlusion uncertainty. Since



(a)

(b)

(c)

Fig. 5-10: Experimental results of experiment 2 using control gains $(\alpha_1,\alpha_2,\alpha_3)_L$ = (5/16,6/8,4/16). (a) Robot trajectory in the world coordinates. (b) Tracking errors in the image plane. (c) Control velocities of the center point and tilt camera of tracking robot.

the estimation without velocity information is more difficult compared with that with velocity information, only the simulation results of the VSE-WoTV are presented.

5.4.1 Computer Simulations

In order to evaluate the estimation performance of the proposed VSE-WoTV, a simulation environment is setup using MATLAB. Figure 5-11 shows the architecture of the simulation setup. In Fig. 5-11, X_n, which includes the target state $X_i[n]$ and target image velocity $\dot{X}_i^t$, denotes the ideal state needed to be estimated by the VSE-WoTV. $X_i[n]$ is obtained from the coordinate transformations (3.4) and (3.7), and $\dot{X}_i^t$ is calculated by (3.11)

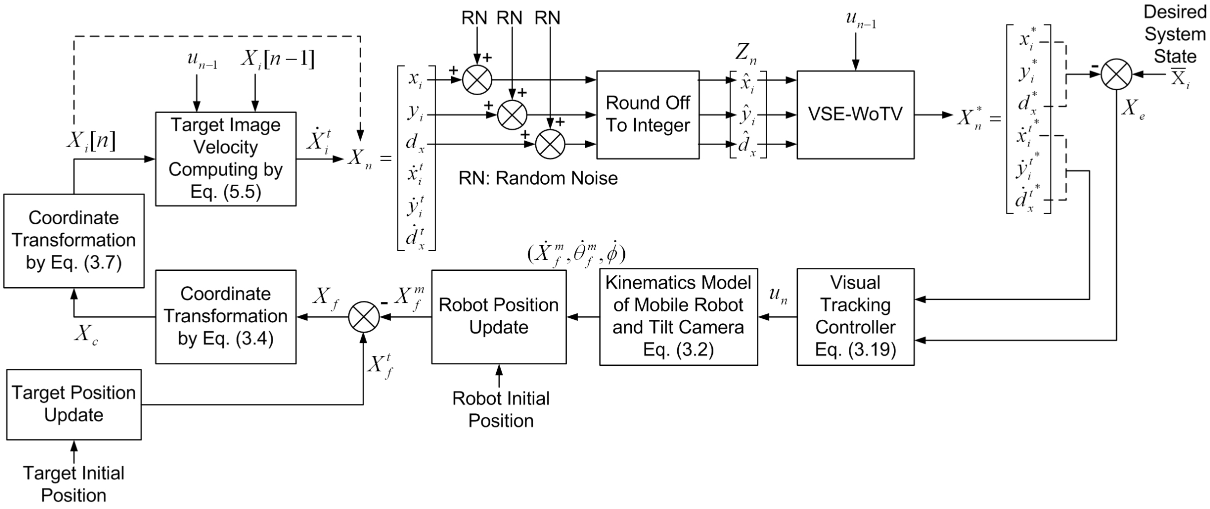

RN RN RN
u_{n-1}
Desired System State
$\bar{X}_i$
u_{n-1}
$X_i[n-1]$
$X_i[n]$
Target Image Velocity Computing by Eq. (5.5)
$\dot{X}_i^t$
$X_n =$
x_i y_i d_x $\dot{x}_i^t$ $\dot{y}_i^t$ $\dot{d}_x^t$
Round Off To Integer
Z_n
$\hat{x}_i$ $\hat{y}_i$ $\hat{d}_x$
VSE-WoTV
$X_n^* =$
x_i^* y_i^* d_x^* $\dot{x}_i^{t*}$ $\dot{y}_i^{t*}$ $\dot{d}_x^{t*}$
X_e
RN: Random Noise
Coordinate Transformation by Eq. (3.7)
$(\dot{X}_f^m, \dot{\theta}_f^m, \dot{\phi})$
Coordinate Transformation by Eq. (3.4)
X_f
X_f^m
Robot Position Update
Kinematics Model of Mobile Robot and Tilt Camera Eq. (3.2)
u_n
Visual Tracking Controller Eq. (3.19)
X_c
X_f^t
Robot Initial Position
Target Position Update
Target Initial Position

Fig. 5-11: Simulation setup for the performance evaluation of the proposed VSE-WoTV presented in Section 4.3.

such that

$$\dot{X}_i^t = \dot{X}_i - \dot{X}_i^m = \frac{X_i[n] - X_i[n-1]}{\mathrm{T}} - \mathbf{B}_i u_{n-1}. \tag{5.5}$$

The observation signal Z_n is obtained by the rounding off the value of $X_i[n]$ with random noise (RN) to integer. In this paper, the random noise is given by

$$RN = \begin{cases} K_n \sigma_1 (0.5 - \sigma_2), & if \ \ (\sigma_3 < \rho) \\ (1 + \sigma_1)(0.5 - \sigma_2), & otherwise \end{cases} \tag{5.6}$$

where $K_n > 1$ is the noise gain; $\sigma_i \in [0,1]$, i=1~3, are three random signals with uniform distribution; and $\rho \in [0,1]$ is a constant threshold value. Expression (5.6) indicates that the intensity of the noise is time-varying and dependent on a random condition. If the condition $(\sigma_3 < \rho)$ is satisfied, then the random noise will have large noise gain; otherwise the random noise will only have noise gain smaller than 2. Thus, the threshold value ρ determines the probability of the event of appearing large observation noise. This kind of random noise is usually happened during practical visual tracking process of the mobile robot, since the intensity of the observation uncertainty usually is position-dependent and light-dependent. The parameters used in the simulations are listed in the Table 5-5.

Figure 5-12 presents the computer simulation results of the visual tracking control

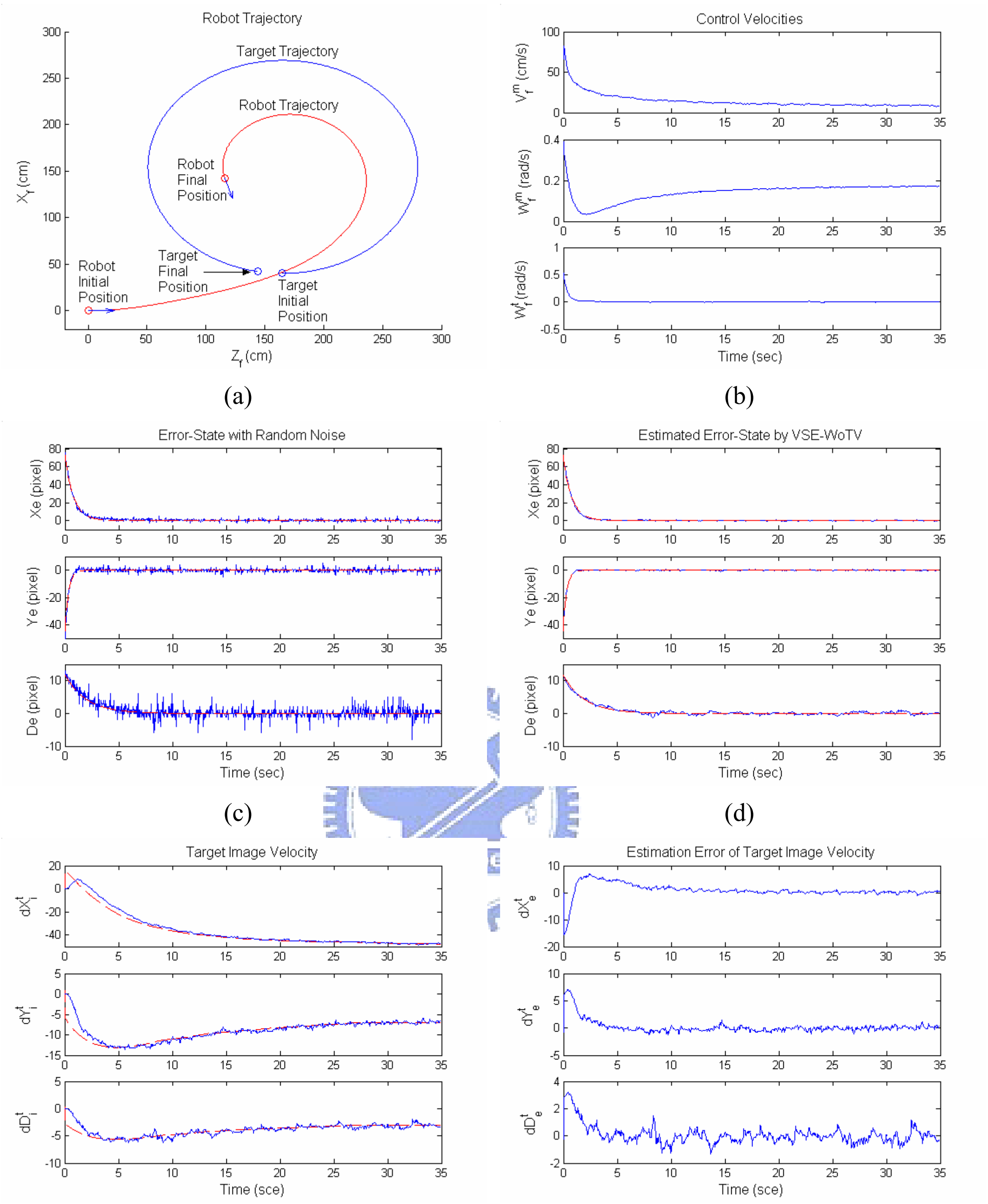
Table 5-5: Parameters used in the simulations of visual state estimation.

Symbol	Quantity	Description
$(\mathrm{f}_x,\mathrm{f}_y)$	(294,312) pixels	Camera focal length in retinal coordinates.
W	12 cm	Width of the target.
D	40 cm	Distance between two drive wheels.
δy	10 cm	Distance between the center of robot tilt platform and the onboard camera
T	35 ms	Sampling period of the control system
$(\bar{\mathrm{x}}_i,\bar{\mathrm{y}}_i,\bar{\mathrm{d}}_x)$	(0,0,35)	Desired system state in the image plane.
$(\alpha_1,\alpha_2,\alpha_3)$	(5/4,3,1/2)	Positive control gains used in the experiments.
$\mathbf{Q}_0$	diag(1,1,1,2,2,2)	Initial covariance matrix
$\boldsymbol{K_n}$	15	Noise gain
ρ	0.75	Constant threshold value

system shown in Fig. 5-11. Figure 5-12(a) shows the robot trajectory in the world coordinate frame. In the simulation, the motion of the target is also set as a circular motion with velocity

$$(v_f^x, v_f^y, v_f^z) = (v_f^t \sin\theta_f^t, 0, v_f^t \cos\theta_f^t),$$

where $v_f^t = 20\,\mathrm{cm/sec}$ and $\theta_f^t(new) = \theta_f^t(old) + (\mathrm{T}\pi/18)\,\mathrm{rad}$ with $\theta_f^t(0) = 0$. From Fig. 5-12(a), we observe that the motion trajectory of tracking robot is also a circular path in order to follow the target. Figure 5-12(b) shows the control velocities of the center point and tilt camera of tracking robot. It reveals that the tracking robot's linear and angular velocities converge to constants when the tracking errors decay to zero. Therefore, the tracking robot kept tracking the target continuously. Figure 5-12(c) shows the tracking errors with random noise (5.6), and Fig. 5-12(d) is the corresponding tracking errors estimated by the VSE-WoTV. In Figs. 5-12(c) and 5-12(d), the dotted lines illustrate the ideal tracking errors while the solid lines show the observation and estimation results of tracking errors. A comparison of Fig. 5-12(c) with Fig. 5-12(d) shows that the random noise in each error state is removed efficiently, especially the error states y_e and d_e. Thus, the robustness of the proposed VSE-WoTV against the random noise uncertainty is validated. Moreover, in Fig. 5-12(d), each error state converges to zero exponentially and smoothly, which validates the tracking



(a) (b)

(c) (d)

(e) (f)

Fig. 5-12: The computer simulation results of the proposed VTC combined with the VSE-WoTV. (a) Robot trajectory in the world coordinate frame. (b) Control velocities of the center point and tilt camera of tracking robot. (c) Tracking errors with random noise. (d) Tracking errors estimated by the VSE-WoTV. (e) Estimated target image velocity. (f) Estimation errors.

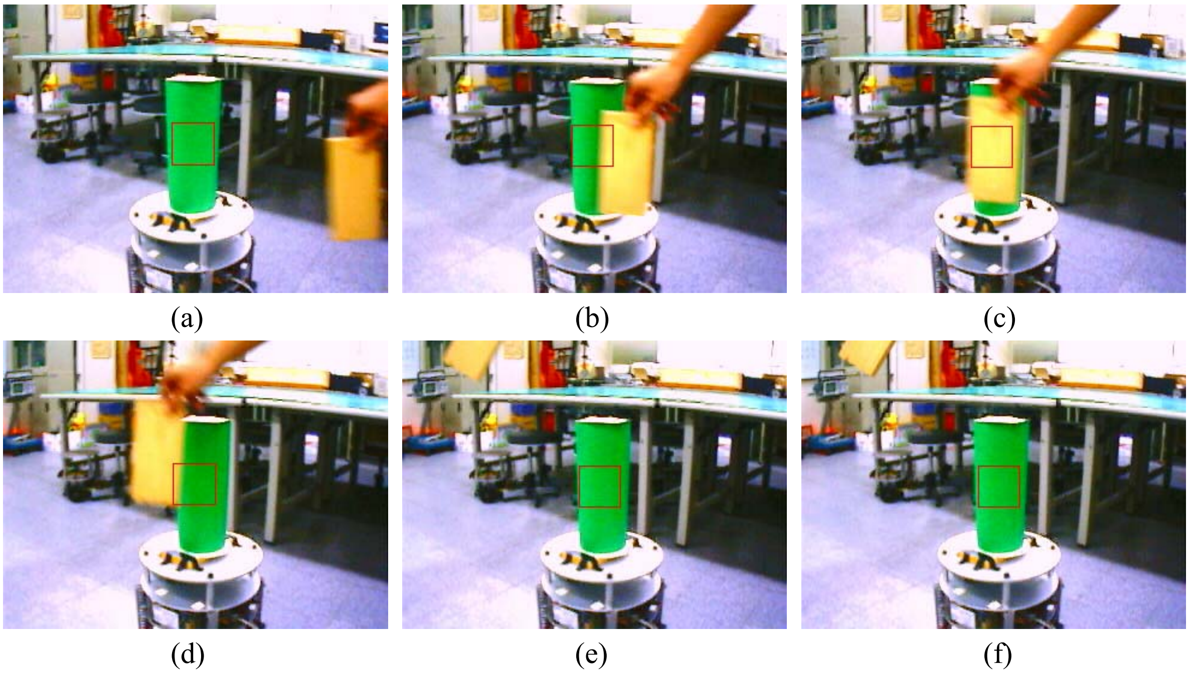
performance of the proposed VTC. Figures 5-12(e) and 5-12(f), respectively, present the estimation results and the estimation errors of target image velocity from the VSE-WoTV. In

Figs. 5-12(e), the dotted lines indicate the ideal target image velocity while the solid lines show the estimation results of target image velocity. It is clear that each estimate converges to the corresponding ideal one. This result also can be seen in Fig. 5-12(f), which shows that each estimation error converges to zero efficiently. Therefore, these simulation results validate the estimation performance of the proposed VSE-WoTV.

5.4.2 Experiment 3: Visual Tracking of a Moving Robot

This section presents the experimental results of tracking a moving target when it is temporarily partially and fully occluded during visual tracking to validate the occlusion robustness of the visual tracking control system shown in Fig. 5-5. The experimental mobile robots and control system parameters used in this experiment are all the same with that used in experiments 1 and 2. In order to verify the performance of the VSE-WTV proposed in Section 4.1, the motion velocity of the moving robot is supposed to be known *a-priori* in this experiment.

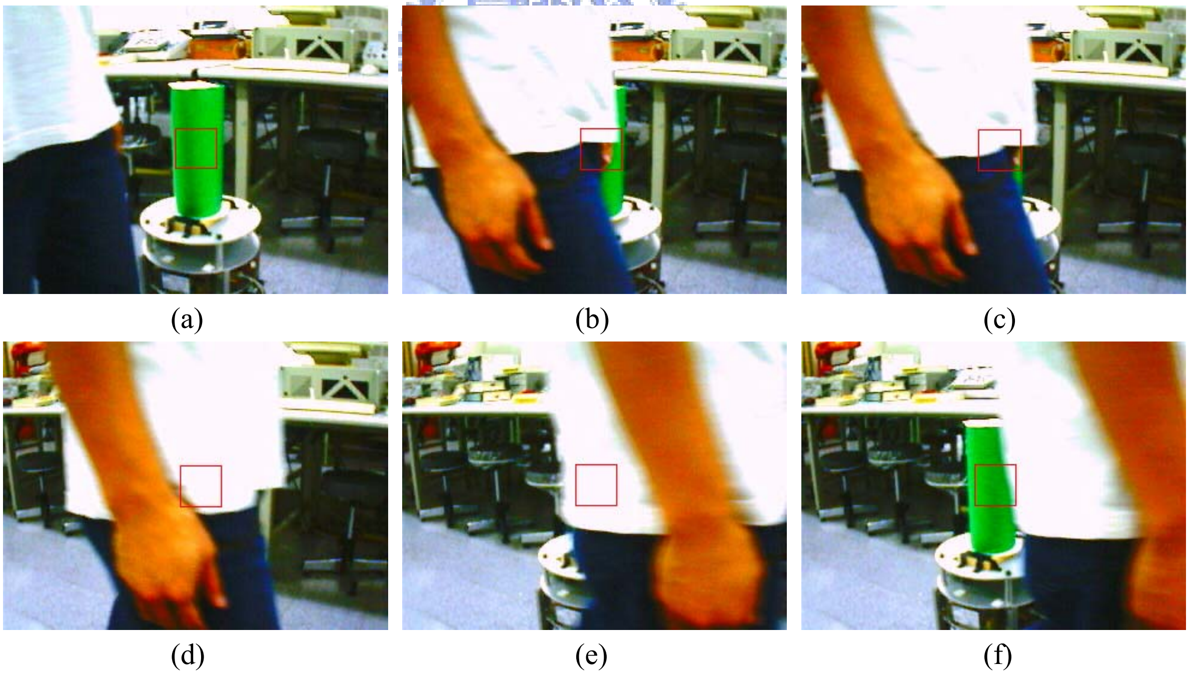
Figure 5-13 illustrates the photos of partial occlusion experimental results recorded from the tilt camera of the tracking robot (the robot with a camera). Figure 5-13(a) shows the tracked target before partial occlusion. In Figs. 5-13(b)-(d), the moving target is temporarily partially occluded by the moving object. Figures 5-13(e)-(f) show that the moving target is still tracked after partial occlusion. In Fig. 5-14, the target was fully blocked by a moving person. Figure 5-14(a) shows the tracked target before full occlusion. In Figs. 5-14(b)-(e), the moving target is temporarily fully occluded by a moving person. Because the target would not be observable in the observed image, the VSE-WTV estimated the moving target only using prediction information. Hence the moving target is still tracked even though it is unobservable. Figure 5-14(f) shows that the moving target is tracked successfully after full occlusion. Figure 5-15 compares the observed tracking errors (the dotted lines with spikes) with the estimated ones (the solid lines). From Fig. 5-15, we see that the random noise caused by the temporary



(a) (b) (c)

(d) (e) (f)

Fig. 5-13: Experimental results of tracking a moving target when it is temporarily partially occluded. (a) Before partial occlusion. (b)-(d) Partial occlusion occurred. (e)-(f) After partial occlusion, the moving target was still under tracking.



(a) (b) (c)

(d) (e) (f)

Fig. 5-14: Experimental results of tracking a moving target when it is temporarily fully occluded. (a) Before full occlusion. (b)-(e) Full occlusion occurred. The moving target is estimated only using prediction information. (f) After fully occlusion, the moving target was still under tracking.

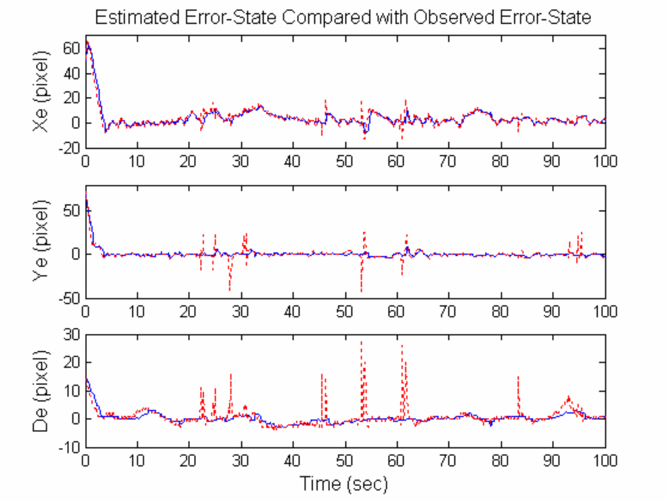


Fig. 5-15: Estimated tracking errors compared with observed tracking errors.

occlusion effect is removed efficiently by utilizing the proposed VSE-WTV. Therefore, based on the above occlusion experiments, the robust estimation performance of the visual tracking control system shown in Fig. 5-5 is verified. A video clip of the experimental results is available online in [57].

Remark 5.3**:** It is interesting to know how long of the occlusion can be handled by using the proposed visual tracking control system. In the current design, the optimal covariance matrix $\mathbf{P}_n^*$ of the self-tuning Kalman filter output is used to evaluate the reliability of the estimation result. If one of the diagonal values of the covariance matrix $\mathbf{P}_n^*$ is larger than a preset threshold, then it implies that the estimation result is not reliable, and thus the visual tracking control system will stop and reinitialize. Therefore, the occlusion handling time is dependent on the value of preset threshold. In the experiments, the value of preset threshold is 4800, and the occlusion handling time is about 3 sec, which is enough to overcome the temporary occlusion problem.

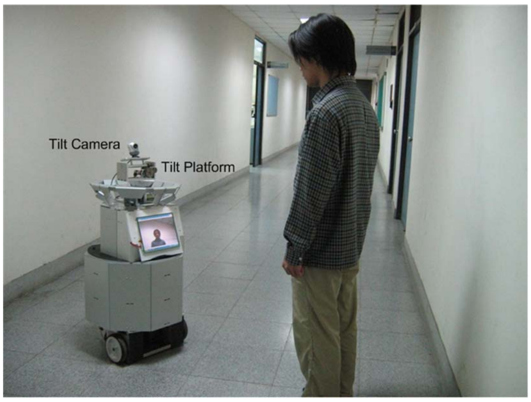


Fig. 5-16: Experimental mobile robot used to test the tracking performance of the proposed VTC combined with the VSE-WoTV presented in Section 4.3.

Table 5-6: Parameters used in the experiment of visual tracking of a moving person.

Symbol	Quantity	Description
$(\mathrm{f}_x, \mathrm{f}_y)$	(393.4,391.8) pixels	Camera focal length in retinal coordinates.
W	12 cm	Width of the target.
D	40 cm	Distance between two drive wheels.
δy	10 cm	Distance between the center of robot tilt platform and the onboard camera
T	100 ms	sampling period of the control system
$(\bar{\mathrm{x}}_i, \bar{\mathrm{y}}_i, \bar{\mathrm{d}}_x)$	(0,0,35)	Desired system state in the image plane.
$(\alpha_1, \alpha_2, \alpha_3)$	(5/4,3,1/2)	Positive control gains used in the experiments.
$\mathbf{Q}_0$	diag(5,5,5,20,20,20)	Initial covariance matrix
$\mathbf{R}_0$	diag(15,15,15)	Initial observation covariance matrix

5.4.3 Experiment 4: Visual Tracking of a Moving Person

In this section, the tracking performance of the proposed VTC combined with the VSE-WoTV is tested by tracking a moving person. Figure 5-16 shows the experimental mobile robot equipped with a tilt camera for the study of visual tracking of a moving target without its motion velocity information. Table 5-6 tabulates the parameters used for the VTC and VSE-WoTV in this experiment. In order to detect and track the user in the image plane, a real-time face detection and tracking algorithm presented in our previous work [60] is utilized to combine with the visual tracking control system. Figure 5-17 illustrates the complete visual tracking control system which encompasses the face detection/tracking algorithm, the VTC

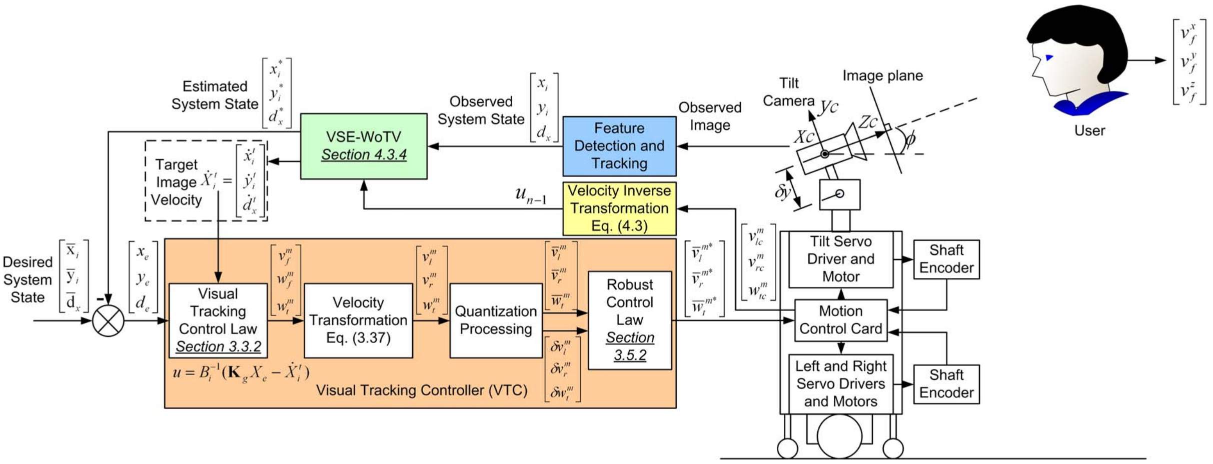

Estimated System State
$\begin{bmatrix} x_i^* \\ y_i^* \\ d_x^* \end{bmatrix}$
VSE-WoTV
Section 4.3.4
Observed System State
$\begin{bmatrix} x_i \\ y_i \\ d_x \end{bmatrix}$
Feature Detection and Tracking
Observed Image
Tilt Camera
Image plane
Yc
Xc
Zc
ϕ
δy
User
$\begin{bmatrix} v_f^x \\ v_f^y \\ v_f^z \end{bmatrix}$
Target Image Velocity $\dot{X}_i^t = \begin{bmatrix} \dot{x}_i^t \\ \dot{y}_i^t \\ \dot{d}_x^t \end{bmatrix}$
u_{n-1}
Velocity Inverse Transformation Eq. (4.3)
Desired System State
$\begin{bmatrix} \overline{x}_i \\ \overline{y}_i \\ \overline{d}_x \end{bmatrix}$
-
$\begin{bmatrix} x_e \\ y_e \\ d_e \end{bmatrix}$
Visual Tracking Control Law
Section 3.3.2
$\begin{bmatrix} v_f^m \\ w_f^m \\ w_t^m \end{bmatrix}$
Velocity Transformation Eq. (3.37)
$\begin{bmatrix} v_l^m \\ v_r^m \\ w_t^m \end{bmatrix}$
Quantization Processing
$\begin{bmatrix} \overline{v}_l^m \\ \overline{v}_r^m \\ \overline{w}_t^m \end{bmatrix}$
$\begin{bmatrix} \delta v_l^m \\ \delta v_r^m \\ \delta w_t^m \end{bmatrix}$
Robust Control Law
Section 3.5.2
$u = B_i^{-1}(\mathbf{K}_g X_e - \dot{X}_i^t)$
Visual Tracking Controller (VTC)
$\begin{bmatrix} \overline{v}_l^{m*} \\ \overline{v}_r^{m*} \\ \overline{w}_t^{m*} \end{bmatrix}$
$\begin{bmatrix} v_{lc}^m \\ v_{rc}^m \\ w_{tc}^m \end{bmatrix}$
Tilt Servo Driver and Motor
Shaft Encoder
Motion Control Card
Left and Right Servo Drivers and Motors
Shaft Encoder

Fig. 5-17: Block diagram of the visual tracking control system tested in the experiment of visual tracking of a moving person.

presented in Section 3.3 and the VSE-WoTV described in Section 4.3. Because the velocity of human motion is unknown, the VSE-WoTV aims to estimate the image velocity instead of the motion velocity for the VTC used and thus overcome the temporary occlusion problem.

Figure 5-18 presents the recorded images and responses of the mobile robot and tilt camera in the case 1 experiment, which includes occlusions to validate the robustness of the proposed visual tracking control system. Figures 5-18(a1-a7) show the recorded pictures from a digital video (DV) camera, and Figs. 5-18(b1-b7) are the corresponding pictures recorded by the on-board USB camera. Figs. 5-18(c-e) and Figs. 5-18(f-h) depict the response of the tracking errors (x_e, y_e, d_e) and target image velocity estimates ($\dot{x}_i^t$, $\dot{y}_i^t$, $\dot{d}_x^t$), respectively. Figs. 5-18(i-k) illustrate the response of robot and tilt camera control velocities (v_f^m, w_f^m, w_t^m).

In the beginning, the user statically sat on a stool, and the robot started to track his face using the proposed visual tracking control system. From Figs. 5-18(f-h), one can see that the target image velocity estimates all approach to zero when robot started working about 5 sec. Next, the user stood up (Fig. 5-18(a2)) and the tilt camera worked to keep tracking his face. From Fig. 5-18(g), we observe that the target image velocity estimate $\dot{y}_i^t$ increased when the user stood up. In the following, the user walked forward (Fig. 5-18(a3)) and the robot moved

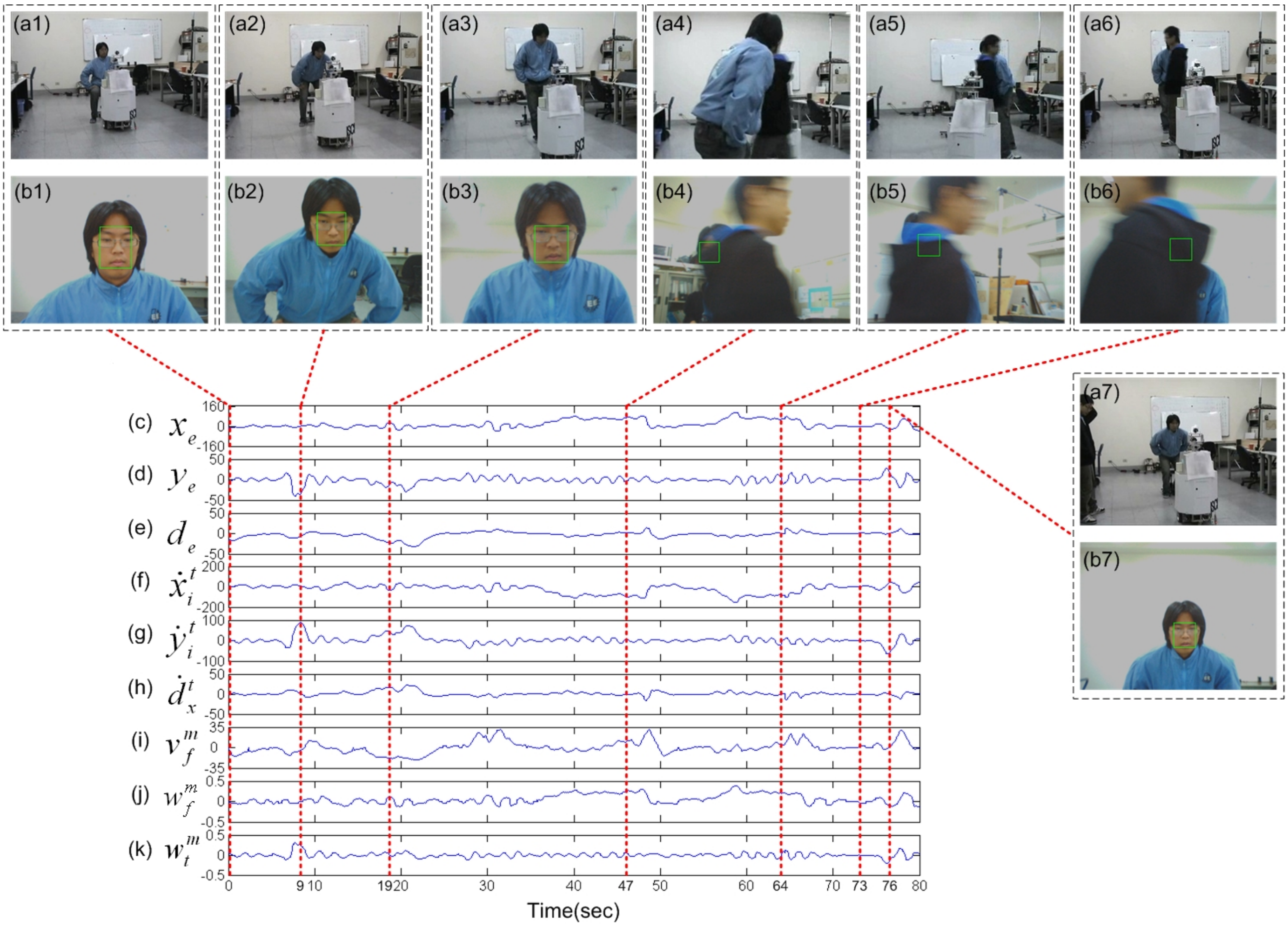


Fig. 5-18: Experimental results. (a1-a7): Image sequence recorded from a DV camera. (b1-b7): Corresponding image sequence recorded from on-board USB camera. (c-d): Recorded tracking errors in the image plane. (f-h): Target image velocity estimates. (i-j): Command linear and angular velocities of the mobile robot. (k): Command velocity of the tilt camera.

backward to keep tracking user's face. From Fig. 5-18(h), it is clear that the image velocity estimate $\dot{d}_i^t$ increased when the user walked forward. These estimation results are consistent with the practical situation.

When the user walked around in the room, the robot kept following and tracking the user's face. While the user was walking, another person walked across between them temporarily (Figs. 5-18(a4-a6)). Thus, in Figs. 5-18(b4-b6), the user's face was temporarily fully blocked by the walking person. In this situation, the variation of observed time sequence data will rise, and the corresponding variance will become large. Based on the proposed

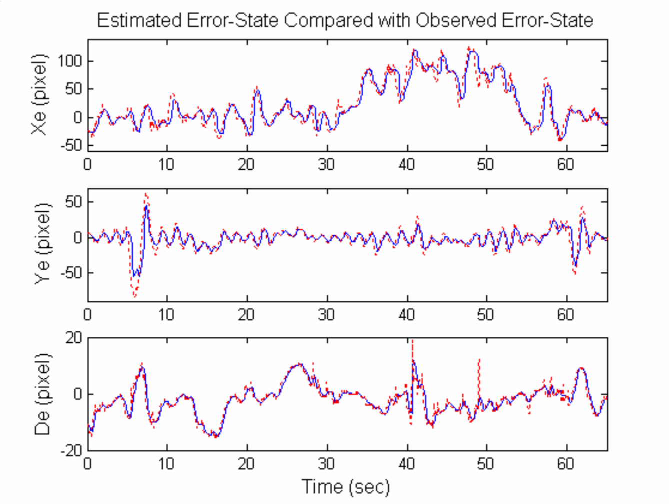


Fig. 5-19. Estimated tracking errors compared with observed tracking errors.

self-tuning algorithm, the prediction information will dominate the estimation results of the Kalman filter even if the target is fully unobservable. Therefore, as shown in Figs. 5-18(c-h), the self-tuning Kalman filter still estimated the positions and velocities of the unobservable moving target in the image plane successfully even during full occlusion. This occlusion experiment validates the robust estimation performance of the proposed visual tracking control system. Finally, the user sat down on the stool, and the robot tracked the user continuously. Figure 5-19 compares the observed tracking errors (the dotted lines with spikes) with the estimated ones (the solid lines). From Fig. 5-19, it is clear that the random noise is also removed efficiently by the proposed VSE-WoTV algorithm. Therefore, the above experiments verify the robust estimation performance of the visual tracking control system shown in Fig. 5-17. A video clip of experiment 4 is available online in [57].

Remark 5.4: The main differences between the proposed method and the existing video color object tracking (VCOT) methods, such as CamShift algorithm [61], are twofold. First, the existing VCOT methods usually suppose that the target has located in the camera's field of view and do not consider the camera motion effect. On the contrary, the proposed method considers both camera and target motion effects to increase the tracking performance and system robustness. Second, the existing VCOT methods usually do not deal with the temporary full occlusion problem. In contrast, the proposed method uses the propagation information to deal with the temporary full occlusion problem. Moreover, the propagation covariance matrix can be used to evaluate the reliability of the tracking state under the situation of full occlusion. Please see Remark 5.3 for the details.

5.4.4 Additional Experiment: Occlusion Robustness Property

Since the current VSE design is based on the Kalman filter algorithm, the estimation performance is dependent on the accuracy of covariance matrices $\mathbf{P}_n$ and $\mathbf{R}_n$. In order to demonstrate this property, the proposed visual tracking control system is extended to control a pan-tile camera platform in this experiment. Figure 5-20 shows the experimental pan-tile platform equipped with a camera to track the face of a user. The control velocities of pan-tile platform can be computed by simplifying the proposed control law (3.20) such that

$$\begin{bmatrix} w_f^{pan} \\ w_f^{tilt} \end{bmatrix} = \begin{bmatrix} B_{12} & B_{13} \\ B_{22} & B_{23} \end{bmatrix}^{-1} \begin{bmatrix} \alpha_1 x_e - \dot{x}_i^t \\ \alpha_2 y_e - \dot{y}_i^t \end{bmatrix}, \tag{5.7}$$

where w_f^{pan} is the pan control velocity, w_f^{tilt} is the tilt control velocity, and B_{mn} denotes an element of matrix $\mathbf{B}_i$ corresponding to the m-th row and n-th column.

Figure 5-21 presents the experimental results. Figure 5-21(a)-(c) show the recorded images, in which the green and magenta windows indicate the observation and propagation, respectively. Figures 5-21(d) and 5-21(e), respectively, illustrate the variance value of state x_i in propagation and observation covariance matrices. Because the face tracking algorithm employed in current system only uses the skin color to detect the human face in a local search window, the algorithm will track another person's face which moves across the user's face and camera. In this situation, the variance value of observation covariance matrix will increase greatly due to the rapid change in the observation. Thus, in Fig. 5-21(e), we see that

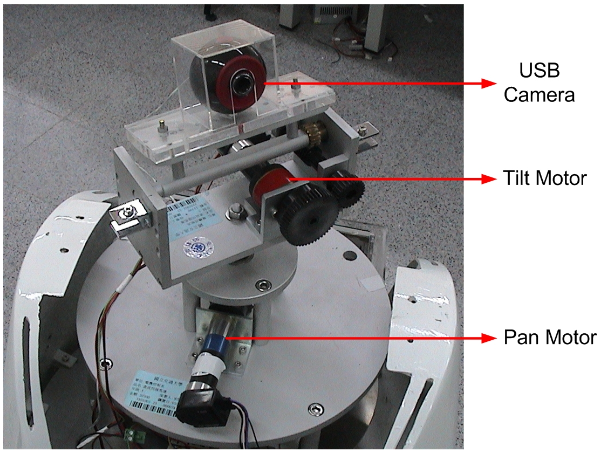


Fig. 5-20: Experimental pan-tile platform used to demonstrate the robust property of the proposed visual tracking scheme.

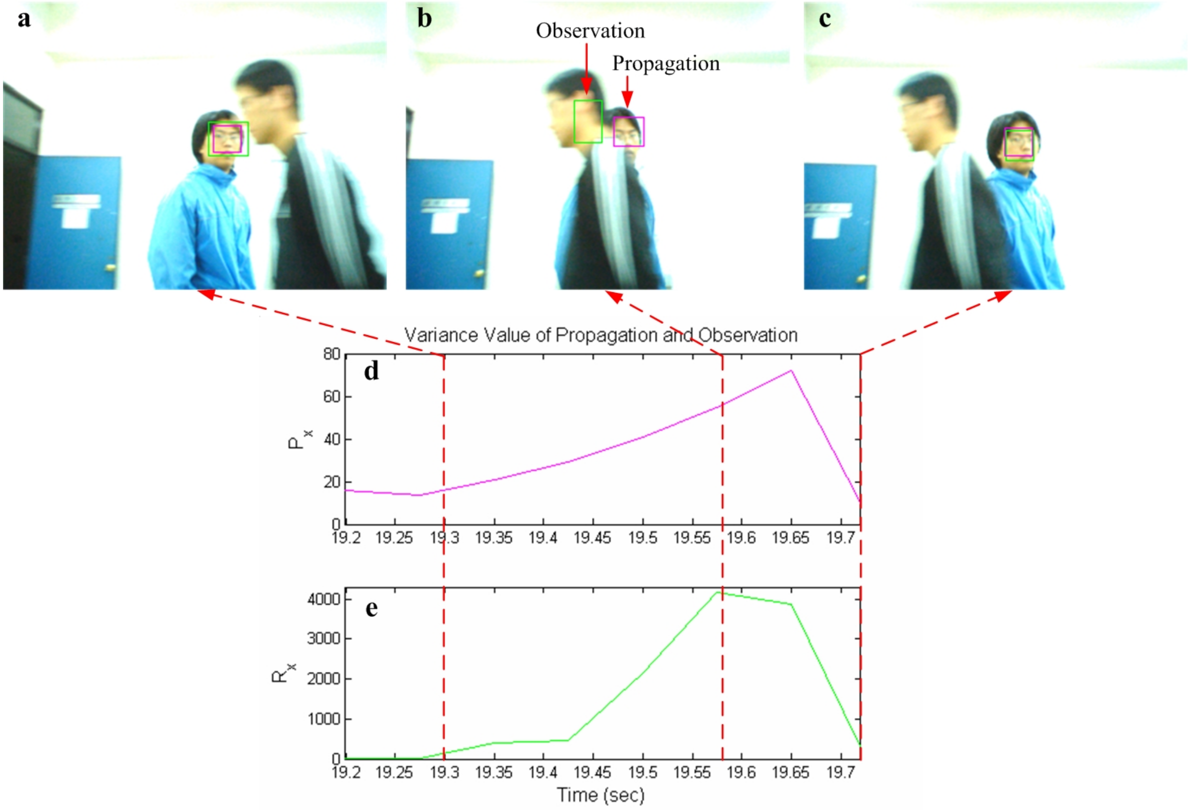


Fig. 5-21: The experimental results of occlusion using VSE-WoTV. (a)-(c) Recorded camera view, observation states and propagation states, (d) variance of propagation states, (e) variance of observation states.

the variance value of the observed state x_i (denoted by R_x) increases rapidly due to the sudden change in observation. On the other hand, the variance value of the propagated state x_i

(denoted by P_x) increases smoothly but much smaller than the observed one. Therefore, after the correction step in Kalman filtering algorithm, the propagation state will dominate the estimation result, which tracks the correct user's face. Finally, in Fig. 5-21(c), the face tracking algorithm detects the human face close to the estimation result, and the observation is corrected. A video clip of this experiment is available online in [57].

Remark 5.5: If there is an object with similar feature and motion to target, then the proposed VSE may track to this object when it moves across the target and camera. However, this problem can be resolved by combining an object recognition algorithm with visual tracking algorithm. In this thesis, we do not cover the object recognition problem and only focus the topic on visual tracking control problem.

5.5 Summary

This chapter evaluates the performance of the proposed CFA interpolation algorithm, visual tracking control scheme and visual state estimation method through computer simulations and experimental results. In CFA interpolation experiments, the performance of the proposed CFA interpolation algorithm has been compared with three renowned CFA interpolation methods. Experimental results show that the proposed method not only outperforms all of them in quantitative comparison, but also gives superior demosaiced fidelities in visual comparison.

In visual tracking control experiments, experimental results validate that the proposed VTC guarantees asymptotic stability and practical stability of the closed-loop visual tracking system with parametric and velocity quantization uncertainties. In visual state estimation experiments, the robustness of the proposed VSE-WoTV against the image noise uncertainty is validated by the computer simulations. The practical experiments then evaluate the performance of the proposed VSEs to overcome the temporary partial/full occlusion uncertainty. Therefore, the tracking performance of the proposed visual tracking control system is enhanced to cope with image noise, system parametric, velocity quantization, and temporary occlusion uncertainties.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

To render a full-color image from a single image sensor covered with a CFA, this work proposes a novel HPHD-CDEA CFA interpolation algorithm for color reproduction of a Bayer mosaic image is proposed. The proposed CFA interpolation algorithm, which consists of HPHD and CDEA CFA interpolation algorithms, effectively reconstructs fine detail features in both texture and edge regions of demosaiced images. The proposed HPHD CFA interpolation algorithm not only can combine with many existing CFA interpolation methods for improved performance, but also provides an efficient method for decision-based algorithms to make accurate direction-selection before performing interpolation. The proposed CDEA CFA interpolation algorithm aims to reduce color artifacts in both smooth and edge regions of demosaiced images by adding the high-frequency information of green channel to other color channels. Any existing image interpolation method can be combined with the proposed CDEA interpolation algorithm to reconstruct the green channel. By combining HPHD algorithm with CDEA CFA interpolation, the proposed method provides an efficient solution for color reproduction of Bayer mosaic images with high performance in both texture and edge regions.

To control a mobile robot for tracking a dynamic moving target in the image plane, this thesis proposes a robust VTC based on a novel dual-Jacobian visual interaction model. In the robustness analysis, we have shown that the proposed VTC possesses some degree of robustness against the system parametric uncertainty. Moreover, based on Lyapunov theory, a robust control law is developed to efficiently overcome the unmodelled quantization effect in the velocity commands. Therefore, the proposed VTC is robust to the uncertainties of system model and velocity quantization. Experimental results validate the effectiveness of the proposed VTC, in terms of tracking performance, system convergence, and robustness.

Two VSEs are designed by using a real-time self-tuning Kalman filter technique in order to estimate the optimal system state and target motion in the image plane directly. In computer simulations, it shows that the proposed VSE provides high robustness against the observation uncertainty with time-varying intensity. This advantage is very useful in practical applications, since the observation uncertainty usually varies with the conditions of target motion and working environment. In practical experiments, the results show that the proposed VSE efficiently overcomes the temporary occlusion uncertainty during visual tracking. Therefore, by combining the proposed VTC with the proposed VSE, the tracking performance of the visual tracking control system is enhanced to cope with image noise, system parametric, velocity quantization, and temporary occlusion uncertainties. This advantage not only provides a useful image-based smooth-motion control solution for wheeled mobile robots to track a target of interesting effectively and interactively, but also enhances the reliability of the visual tracking control system in practical applications.

1.2 Future Directions

Some directions for future study are recommended below:

1) For the future study of CFA interpolation, because the green plane has the most spatial information of the image to be reconstructed and has great influence on the perceptual quality of the image, the future research directions will focus on developing the single-plane reconstruction algorithms to reconstruct the green channel with minimum interpolation error.

2) For future visual tracking control studies, the general case of any task/command

dimension such as a mobile-manipulator system will be directed. In the current VTC design, the robot image Jacobian $\mathbf{B}_i$ in (3.11) is square only because the task was 3-dimensional (coordinates of the target's center and its width) and the robot had 3 commands (linear, angular and tilt velocity). If the camera were mounted on a more articulated manipulator, matrix $\mathbf{B}_i$ would not have been square. Thus, the future work on the study of VTC design should address the general case of any task/command dimension with exploiting redundancy. Moreover, according to Theorem 3.2, the closed-loop visual tracking control system (3.13) under the proposed controller given in (3.20) is practically stable at the origin when the target is moving. Therefore, another future work is to develop a non-smooth controller, such as a variable structure switching (VSS) controller, to achieve asymptotic convergence for tracking a moving target.

3) For the visual state estimation study, because the current VSE design is based on the Kalman filter technique, there are some restrictions on the proposed VSE due to the assumptions of Kalman filter such as Gaussian distribution uncertainty, smoothness motion, and uniform sampling rate. These assumptions restrict the performance of the proposed VSE, which might fail in tracking a highly dynamic moving target. Therefore, the future work on the study of visual state estimation will focus on developing other types of VSE, such as neural-networks based VSE, to solve this problem and improve the accuracy of the visual state estimation results.

4) Because the tracking performance of the visual tracking control system depends on the frame rate of the vision system, a digital high-speed camera system, which allows capture of up to 33,000 frames per second, will greatly increase the tracking performance of the proposed system. Therefore, it is worth to combine the proposed visual tracking control method with a high-speed camera system in the future work.

Appendix A

Parameter Tuning Experiment for the Proposed Color Interpolation Algorithm

Since the value of parameters (N, α) may drastically influence demosaicing performance and hence the comparison results, it is interesting to study how they affect the demosaicing performance of the proposed color interpolation algorithm. In order to evaluate the demosaicing performance, we first define the following criterion

$$PSNR_{Avg}(N, \alpha) = \frac{1}{25} \sum_{i=1}^{25} PSNR(O_i, D_i(N, \alpha)), \qquad \text{(A.1)}$$

where O_i and D_i indicate the *i*th test image and its corresponding demosaiced one by using the proposed HPHD-CDEA method. PSNR (in dB) denotes the metric of peak signal-to-noise ratio defined in (5.1). Based on the criterion (A.1), the parameter *N* is tweaked from 5 to 25 with interval 1, and α is tweaked from 0 to 1 with interval 0.1. Figure A-1 shows the experimental results of tweaking parameters (N, α). Figures A-1(a) and (b), respectively, represent the evolution of $PSNR_{Avg}$ as parameter *N* and α increase. In Fig. A-1(a), one can see that when $\alpha = 0$ (only the smooth set under consideration), the $PSNR_{Avg}$ independs on the parameter *N*. On the other hand, when $\alpha = 1$ (only the horizontal and vertical sets under consideration), the impact of *N* on $PSNR_{Avg}$ is increased. Thus, the influence of *N* on $PSNR_{Avg}$ depends on the parameter α, especially when $\alpha = 1$. Moreover, one can see in Fig. A-1(a) that the local optimal parameter *N* occurs at $N_{opt} = 11$ in the experiment.

Figure A-1(b) shows that the parameter α has significant influence on the $PSNR_{Avg}$. If parameter α increases from 0 to 0.6, the $PSNR_{Avg}$ also increases. However, when parameter α increases from 0.6 to 1, the criterion $PSNR_{Avg}$ becomes decreasing. This implies the local optimal parameter α should occur in the range from 0.5 to 0.6, and the optimal interpolation result will encompass horizontal, vertical and smooth interpolations together. Since parameter $\alpha = 0.6$ obtains the maximum $PSNR_{Avg}$ in post-processing step, we choose $\alpha_{opt} = 0.6$ as the local optimal parameter α.

Figure A-1(c) shows the influence of the parameters (N, α) on the performance gap $\Delta PSNR_{Avg}$ between post-processing and interpolation steps. It is clear that the performance gap mostly depends on the parameter α. Moreover, the maximum performance gap occurs when parameter $\alpha = 1$. This implies that the post-processing provides significant improvement on the horizontal and vertical interpolation results. Therefore, post-processing seems to be more beneficial to the existent soft-decision CFA interpolation algorithms, which only considers the horizontal and vertical interpolations.

Summarizing the tweaking parameter experiment, we have the following findings.

1) For the proposed method, the parameter α has significant influence on the demosaicing performance compared with parameter *N*.

2) When the interpolation only considers horizontal and vertical ones, the post-processing provides significant improvement on the interpolation result.

3) The optimal interpolation result requires encompassing horizontal, vertical and smooth interpolations together.

4) Based on the criterion (A.1), the local optimal parameters (N_{opt}, α_{opt}) of proposed HPHD-CDEA method can be found at $(11, 0.6)$.

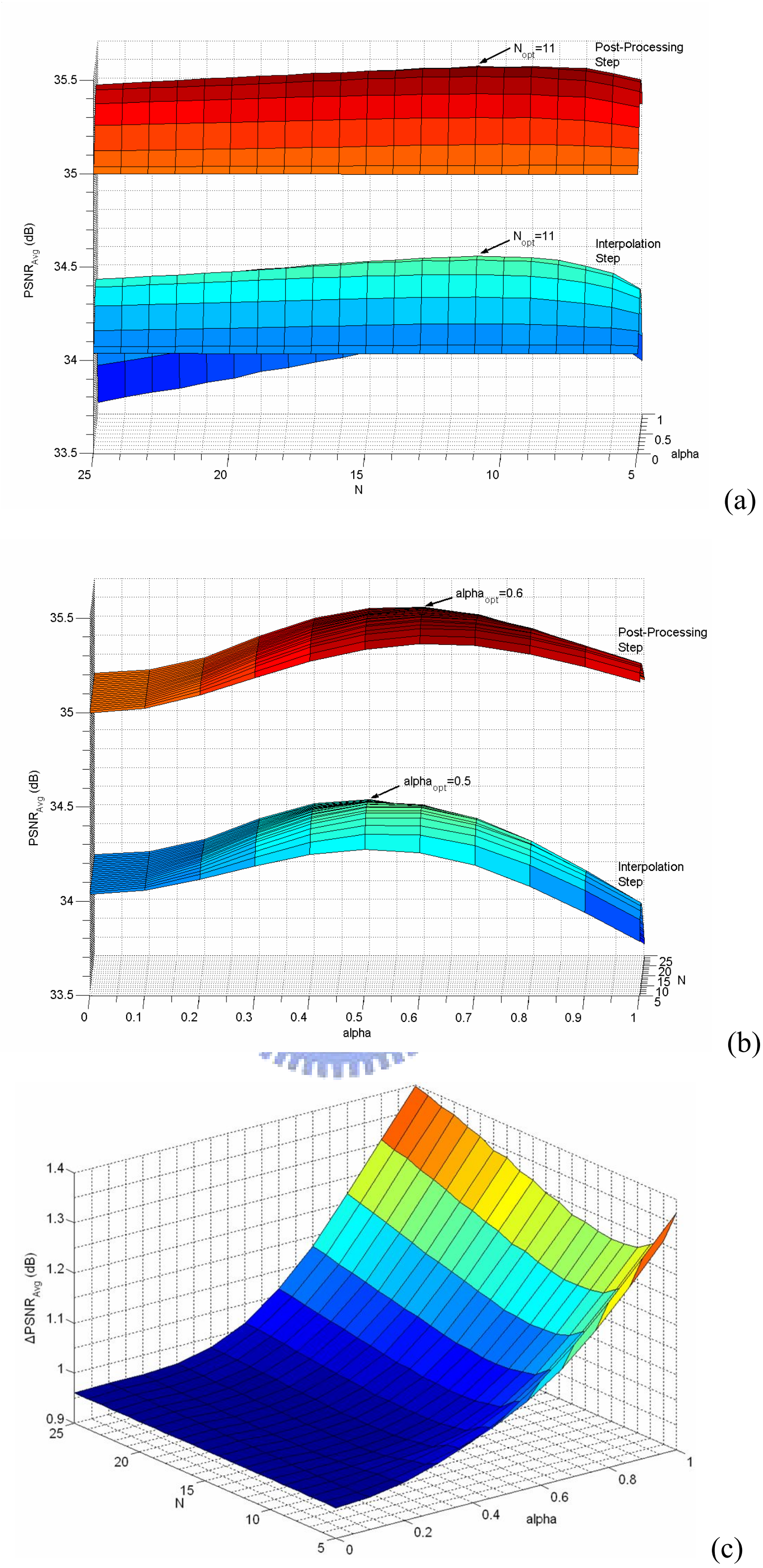


Fig. A-1: Experimental results of tuning parameters in each step. (a) Evolution of $PSNR_{Avg}$ as the parameter N increases. (b) Evolution of $PSNR_{Avg}$ as the parameter α increases. (c) Influence of the parameters (N,α) on the performance gap $\Delta PSNR_{Avg}$ between post-processing and interpolation steps.

Appendix B

Soft-Decision, Enhanced Soft-Decision and Hard-Decision CFA Interpolation Algorithms

To show the difference between the proposed hard-decision method and the existent soft-decision ones, we describe the key idea of these two types of decision-based CFA interpolation algorithms in this section. Figure B-1 presents the flowchart of each decision-based CFA interpolation algorithm. Figure B-1(a) shows the flowchart of the original soft-decision method, in which the former interpolation stage usually generates two interpolated images, one is horizontally interpolated and another one is vertically. The latter decision stage chooses a better one for each color pixel output. In other words, the output image of the original soft-decision method only contains horizontal and vertical interpolated color pixels without smooth ones. In the aforementioned section, we noticed that the optimal interpolation result needs to consider the horizontal, vertical and smooth interpolations together. Therefore, the demosaicing performance of the soft-decision method is limited because only two directional interpolations are under consideration.

To overcome this drawback, Omer *et al.* proposed the enhanced soft-decision CFA interpolation algorithm that regards the soft-decision processing as a meta-algorithm to improve the performance of traditional interpolation methods in places they tend to fail. Figure B-1(b) illustrates the flowchart of enhanced soft-decision CFA interpolation algorithm, in which the former interpolation stage generates not only two directional interpolated images, but also a smooth interpolated one using a standard CFA interpolation method such as edge-directed schemes [11] or [55]. In the latter decision stage, two natural image properties,

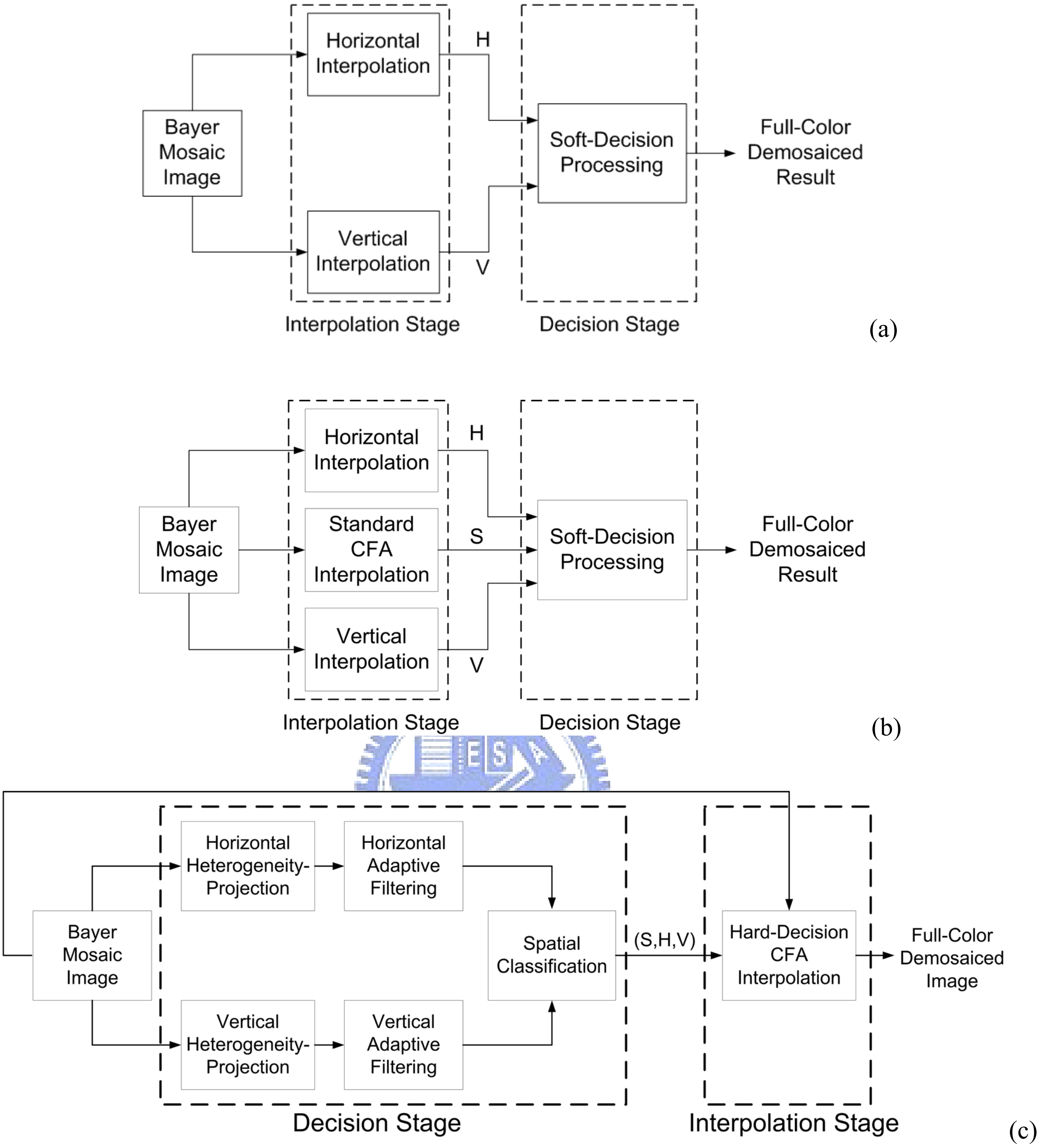


Fig. B-1: Flowchart of the (a) soft-decision; (b) enhanced soft-decision and (c) proposed hard-decision CFA interpolation algorithms.

i.e. color variation and corner value, are employed as the demosaicing hints to evaluate a correct interpolation result and two erroneous ones. Although the enhanced soft-decision CFA interpolation algorithm provides more pleasing demosaiced results, the computation load in latter decision stage is increased greatly because it needs to evaluate three interpolation results.

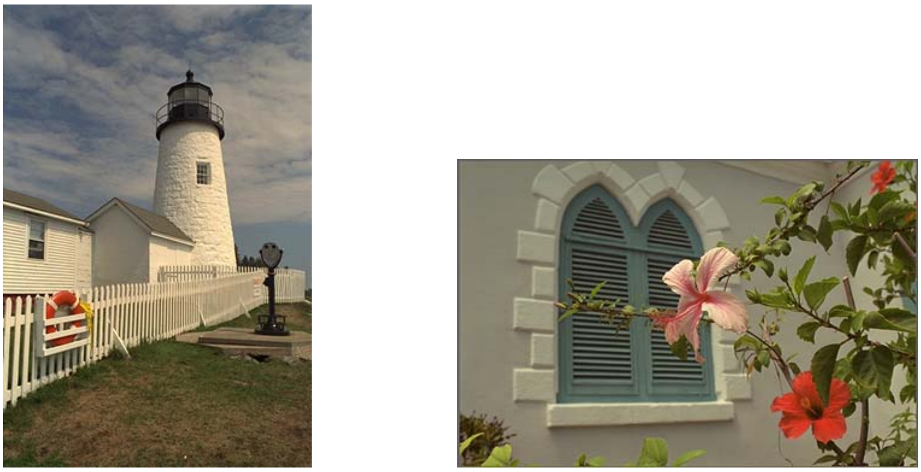
Fig. B-1(c) presents the flowchart of the proposed hard-decision CFA interpolation algorithm. Thanks to the proposed directional heterogeneity-projection and adaptive filtering schemes, the decision stage can be performed directly using the original Bayer mosaic image before interpolation stage. Moreover, the spatial classification also contains horizontal, vertical and smooth subsets for providing more accurate interpolation results in the latter stage. Therefore, the proposed hard-decision method provides comparable results using less computation than the enhanced soft-decision method does.

Appendix C

Extended Visual Comparison

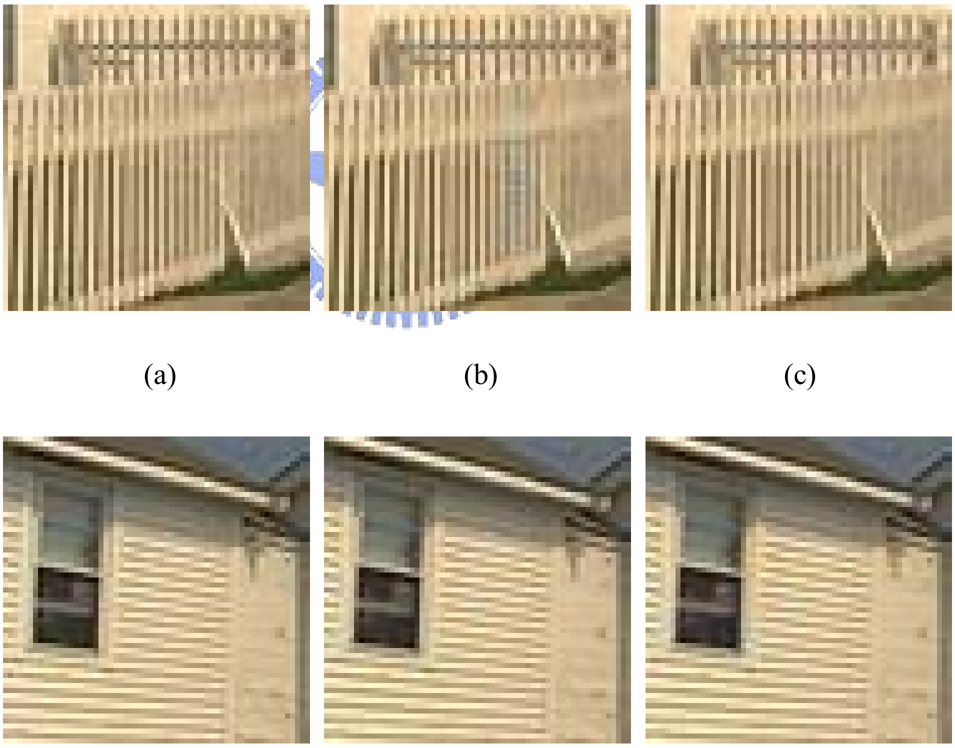
In this appendix, an extended study on visual comparison is presented using two images of 384×256 pixels taken from the Kodak database as shown in Fig. C-1. We visually compared the performance of the proposed demosaicing method with six notable ones: Lu's [3], Gunturk's [5], Li's [6], Muresan's [4], Grossman's [9] and Omer's [10] methods. The parameter setting of [5] and [6] is the same as that in the manuscript. The results of [4] and [9] are obtained directly from the authors' web page in TIF and BMP formats, respectively. For Omer's method, the Kimmel's interpolation method [11] was employed to provide the smooth interpolated image in the interpolation stage.

For the proposed method, we first use the optimal parameters $(N_{opt}, \alpha_{opt}) = (11,0.6)$ to reconstruct the Bayer mosaiced image of Lighthouse. The PSNR metric between the original and demosaiced images is 32.7315dB; however, there still are some noticeable color artifacts in the fence region. In order to reduce these noticeable color artifacts, we increase the values of (N, α) for estimating the horizontal and vertical edges accurately. After a tuning process, the suitable parameters are given by $(N, \alpha) = (24,0.8)$ and the PSNR metric reduces to 32.4413dB. One can see that there is a tradeoff between quantitative and visual qualities. Figure C-2 shows the zoom-in of the demosaiced Lighthouse images using proposed method with parameters $(N_{opt}, \alpha_{opt}) = (11,0.6)$ and $(N, \alpha) = (24,0.8)$ respectively. Figures C-2(a) and (d) show the zoom-in of original Lighthouse in the fence and house regions, respectively. Figures C-2(b) and (e) show the corresponding zoom-in of demosaicing results using parameters $(N_{opt}, \alpha_{opt}) = (11,0.6)$, and (c) and (f) show that using parameters $(N, \alpha) = (24,0.8)$.



(a) (b)

Fig. C-1: Test images used in the extended visual comparison. (a) Lighthouse. (b) Window.



(a) (b) (c)

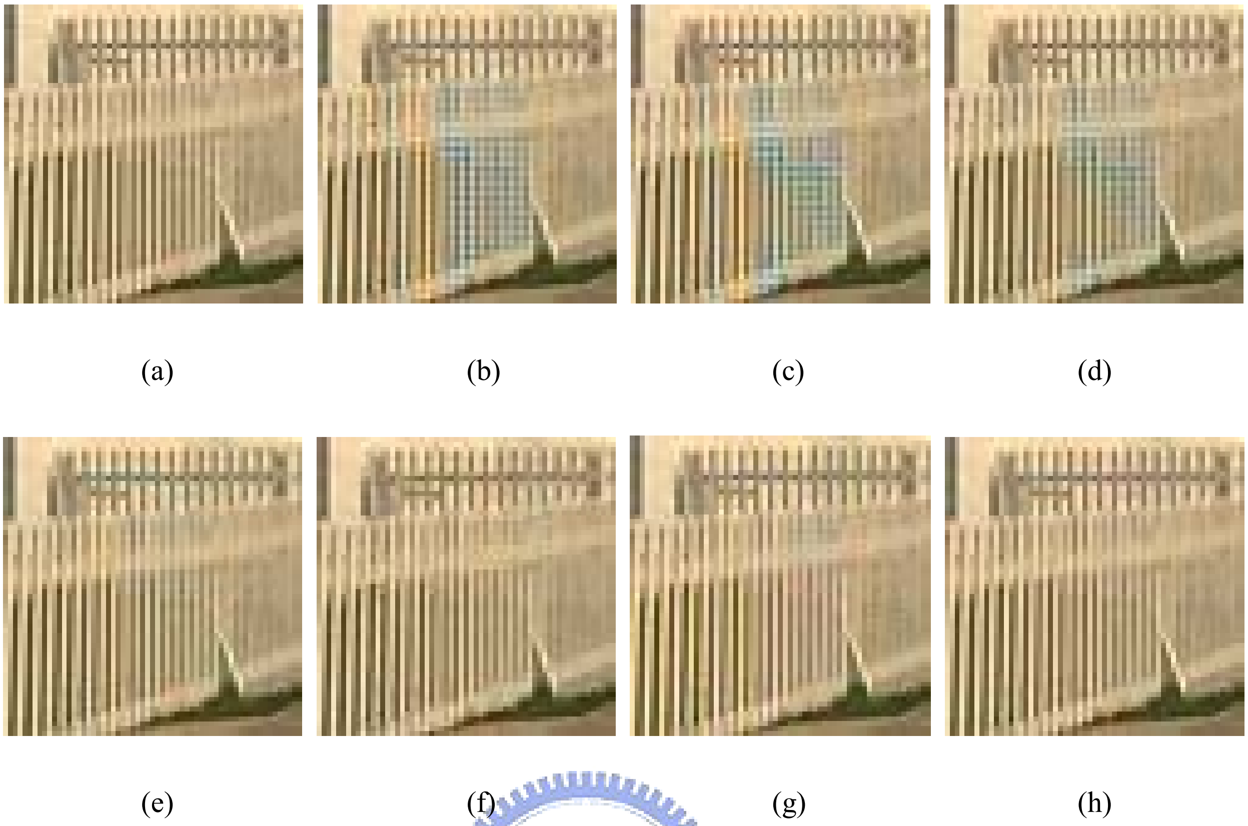
(d) (e) (f)

Fig. C-2: (a) and (d) show the zoom-in of the original Lighthouse image in the fence and house regions, respectively. (b) and (e) show the zoom-in of the demosaicing results using parameters $(N_{opt}, \alpha_{opt}) = (11,0.6)$. (c) and (f) show the zoom-in of the demosaicing results using parameters $(N, \alpha) = (24,0.8)$.

Visually comparing these images, one can see that the demosaiced image using optimal parameters presents better result in house region than that using increased parameters, but some color artifacts still remain in fence region. On the contrary, the demosaiced image using increased parameters presents less color artifacts in fence region; however, the quality in house region is reduced. Therefore, the demosaicing results using increased parameters can reduce the color artifacts in the fence region but cannot provide better PSNR metric than those using optimal parameters.

For further visual comparison, we choose larger parameters $(N,\alpha)=(24,0.8)$ for the proposed method, because it can provide more pleasing results in the fence region of the Lighthouse image. Figures C-3 and C-4, respectively, show the zoom-in of the Lighthouse demosaiced images in fence and house regions reconstructed by the methods under comparison. In Fig. C-3, one can see that Muresan's, Grossman's and the HPHD-CDEA methods provide better demosaicing results in the fence region than the others do. However, in Fig. C-4, one can see that the demosaicing result of Muresan's and Grossman's methods in house region induces more visible artifacts than the proposed method does. Therefore, the proposed method provides superior demosaicing result not only in fence region, but also in house region of the Lighthouse test image compared with other methods.

Figure C-5(a) shows the zoom-in of the original Window image in flower region, and Figs. C-5(b)-(h) present the corresponding demosaiced results of the methods under comparison. Visually comparing these images shown in Fig. C-5, one can see that the demosaiced images obtained by Lu's, Grossman's and the proposed method give more satisfactory results compared with others. Therefore, based on the above visual comparison, the performance improvement of the proposed HPHD-CDEA method on detail regions of the image is verified.



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

(e) (f) (g) (h)

Fig. C-3: Zoom-in demosaicing results of Lighthouse image in fence region. (a) Original picture; Demosaiced result in interpolation step: (b) Lu's method, (c) Gunturk's method, (d) Li's method, (e) Muresan's method, (f) Grossman's method, (g) Omer's method, (h) HPHD-CDEA method.

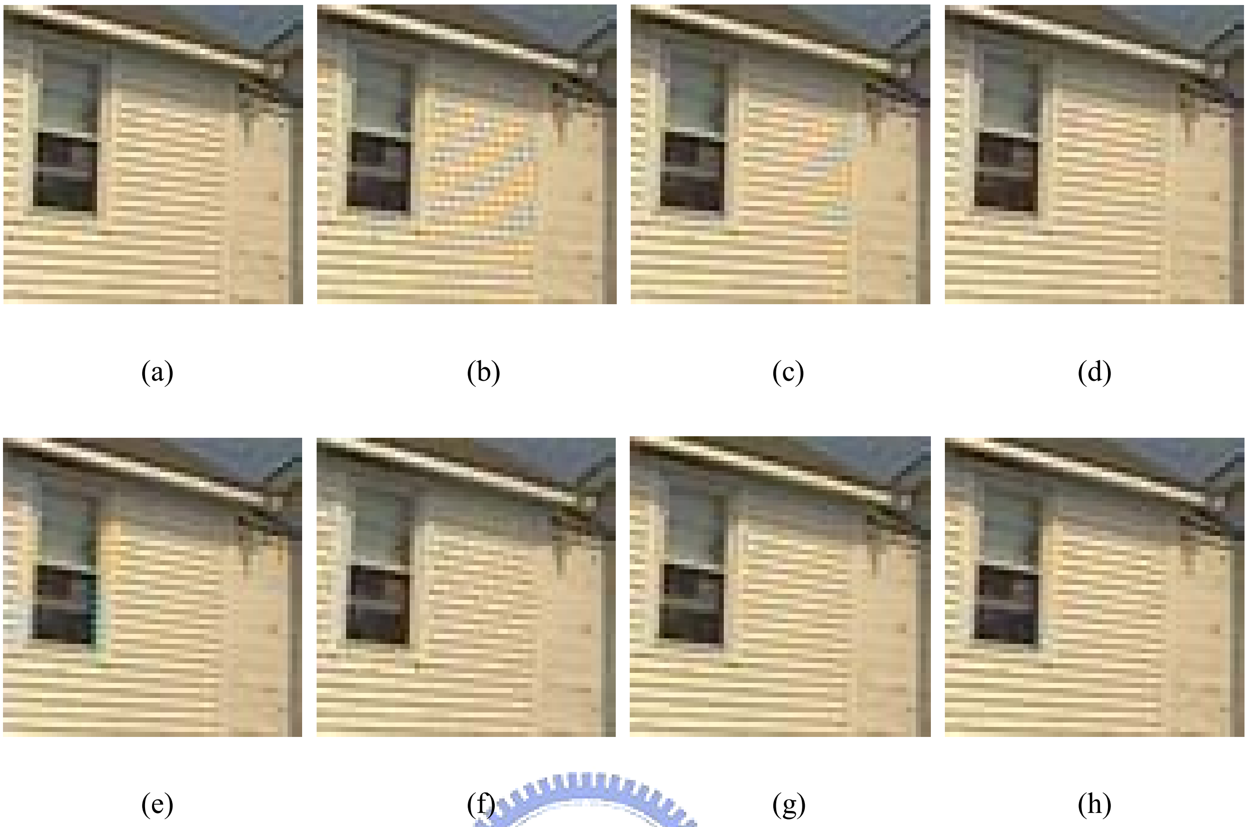
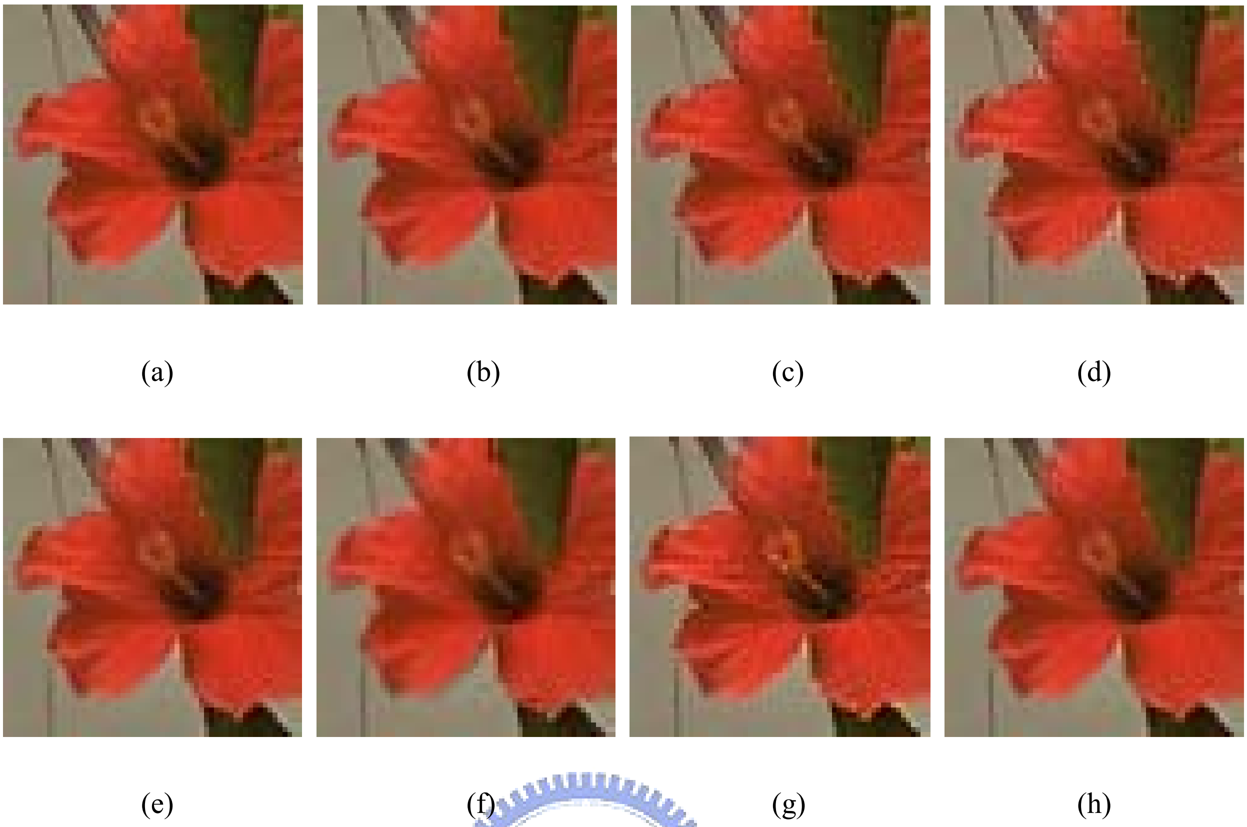


Fig. C-4: Zoom-in demosaicing results of Lighthouse image in house region. (a) Original picture; Demosaiced result in interpolation step: (b) Lu's method, (c) Gunturk's method, (d) Li's method, (e) Muresan's method, (f) Grossman's method, (g) Omer's method, (h) HPHD-CDEA method.



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

(e) (f) (g) (h)

Fig. C-5: Zoom-in demosaicing results of Window image. (a) Original picture; Demosaiced result in interpolation step: (b) Lu's method, (c) Gunturk's method, (d) Li's method, (e) Muresan's method, (f) Grossman's method, (g) Omer's method, (h) HPHD-CDEA method.

Appendix D

Extended Discussion on the Proposed VTC Scheme

Although Theorem 3.1 shows that the proposed VTC guarantees the state X_i toward the desired state $\overline{\mathrm{X}}_i$ in the image plane, it does not proof that the robot have followed the target. In this appendix, we will show that when X_i converges to $\overline{\mathrm{X}}_i$, the mobile robot has followed the motion target. Recall the diffeomorphism defined in (7):

$$X_i = \mathbf{P}_c^i X_c, \tag{D.1}$$

where $\mathbf{P}_c^i = diag(-k_x, k_y, k_x k_w)$. Suppose that X_i has converged to $\overline{\mathrm{X}}_i$, we then have the following result based on (D.1)

$$\overline{\mathrm{X}}_i = \mathbf{P}_c^i X_c^d \text{, and} \tag{D.2}$$

$$X_c^d = [x_c^d \quad y_c^d \quad z_c^d]^T = \mathbf{R}(\phi, \theta_f^m) X_f^d - \delta\mathrm{Y}, \tag{D.3}$$

where X_c^d and X_f^d, respectively, are the related position between mobile robot and motion target in camera and world coordinate frame when $X_i = \overline{\mathrm{X}}_i$. Because $\mathbf{P}_c^i$ is invertible, the following relation between X_f^d and $\overline{\mathrm{X}}_i$ can be obtained by substituting (D.3) into (D.2) such that

$$X_f^d = \mathbf{R}^T(\phi, \theta_f^m)[(\mathbf{P}_c^i)^{-1}\overline{\mathrm{X}}_i + \delta\mathrm{Y}]. \tag{D.4}$$

Let $\|A\|$ denote the 2-norm value of vector or matrix A. The key idea is that if $\|X_f^d\|$ is bounded, it implies that the mobile robot has followed the motion target. Using (D.4), $\left\|X_f^d\right\|$ is given by

$$\begin{aligned}\left\|X_f^d\right\| = \left\|\mathbf{R}^T(\phi,\theta_f^m)[(\mathbf{P}_c^i)^{-1}\overline{\mathrm{X}}_i + \delta\mathrm{Y}]\right\| &\le \left\|\mathbf{R}^T(\phi,\theta_f^m)\right\|\cdot\left\|(\mathbf{P}_c^i)^{-1}\overline{\mathrm{X}}_i + \delta\mathrm{Y}\right\| \\ &\le \left\|(\mathbf{P}_c^i)^{-1}\overline{\mathrm{X}}_i\right\| + \left\|\delta\mathrm{Y}\right\| \\ &\le \left\|(\mathbf{P}_c^i)^{-1}\right\|\cdot\left\|\overline{\mathrm{X}}_i\right\| + \left\|\delta\mathrm{Y}\right\|\end{aligned} \tag{D.5}$$

Because of $\left\|(\mathbf{P}_c^i)^{-1}\right\| = \left\|diag(-k_x^{-1},k_y^{-1},k_x^{-1}k_w^{-1})\right\| = z_c\left\|diag(-f_x^{-1},f_y^{-1},f_x^{-1}\mathrm{W}^{-1})\right\| = z_c\sqrt{\lambda_{\max}}$, where $z_c = f_x\mathrm{W}/d_x$ and $\lambda_{\max} = \max(f_y^{-1},f_x^{-1}\mathrm{W}^{-1})$, we have the following result

$$\left\|X_f^d\right\| \le \frac{f_x\mathrm{W}}{\mathrm{d}_x}\sqrt{\lambda_{\max}}\left\|\overline{\mathrm{X}}_i\right\| + \left\|\delta\mathrm{Y}\right\|. \tag{D.6}$$

From (D.6), it is clear that $\left\|X_f^d\right\|$ is bounded, and hence the proof is completed.

We use the simulation presented in Section 5.4.1 as an example to explain the physical meaning of (D.6). By using the parameters listed in Table 5-5, the term on the right hand side of (D.6) can be calculated by

$$\left\|X_f^d\right\| \le \frac{294\times 12}{35}\sqrt{0.0032}\times 35 + 10 = 209.7337 \quad \text{(cm)}, \tag{D.7}$$

which means that when X_i converges to $\overline{\mathrm{X}}_i$, the distance between mobile robot and motion target is bounded to 209.7337 cm. Figure D-1 shows the simulation result of the distance between mobile robot and motion target. In Fig. D-1, the solid line presents the 2-norm value of X_f, and the dotted line denotes the bounded distance calculated in (D.7). From Fig. D-1, we see that the distance between mobile robot and target finally converges to about 100 cm, which is satisfied in the bounded condition (D.7). Because the target is always moving and the distance between the robot and target is bounded, this implies that the robot has followed the

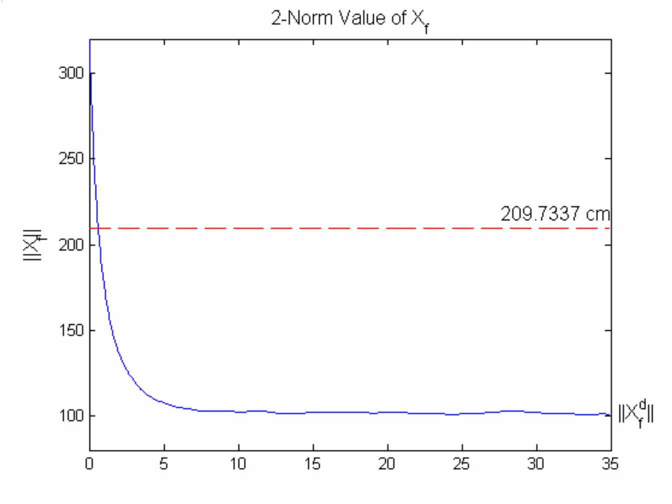


Fig. D-1: Simulation result of the distance between mobile robot and motion target, $\| X_f \|$.

target as we expected.

Bibliography

[1] D. R. Cok, “Reconstruction of CCD images using template matching,” in *Proceedings of International Science and Technology's 47th Annual Conference/International Congress on Imaging Science*, Rochester, USA, 1994, pp.380-385.

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: color filter array interpolation,” *IEEE Signal Processing Magazine*, Vol. 22, No. 1, pp. 44-54, January 2005.

[3] W.-M. Lu and Y.-P. Tan, “Color filter array demosaicking: new method and performance measures,” *IEEE Transactions on Image Processing*, Vol. 12, No.10, pp. 1194-1210, October 2003.

[4] D. D. Muresan and T. W. Parks, “Demosaicing using optimal recovery,” *IEEE Transactions on Image Processing*, Vol. 14, No. 2, pp. 267-278, February 2005.

[5] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Transactions on Image Processing*, Vol. 11, No.9, pp. 997-1013, September 2002.

[6] X. Li, “Demosaicing by successive approximation,” *IEEE Transactions on Image Processing*, Vol. 14, No.3, pp. 370-379, March 2005.

[7] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” *IEEE Transactions on Image Processing*, Vol. 14, No.3, pp. 360-369, March 2005.

[8] X.-L. Wu and N. Zhang, “Primary-consistent soft-decision color demosaicking for digital cameras (patent pending),” *IEEE Transactions on Image Processing*, Vol. 13, No.9, pp. 1263-1274, March 2004.

[9] L. D. Grossmann and Y. C. Eldar, "Enhancement of color images by efficient demosaicing," [Online]. Available: http://www.ee.technion.ac.il/Sites/People/YoninaEldar/Download/GY2-004.pdf

[10] I. Omer and M. Werman, “Using natural image properties as demosaicing hints,” in *Proceedings of IEEE International Conference on Image Processing*, Singapore, 2004, pp. 1665-1670.

[11] R. Kimmel, “Demosaicking: Image reconstruction from color CCD samples,” *IEEE Transactions on Image Processing*, Vol. 8, No. 9, pp.1221-1228, 1999.

[12] N. P. Papanikolopoulos, P. K. Khosla and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision,” *IEEE Transactions on Robotics and Automation*, Vol. 9, No. 1, pp. 14-35, 1993.

[13] S. Hutchinson, G. D. Hager and P. I. Corke, “A tutorial on visual servo control,” *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, 1996.

[14] J. Su, R. Qiu, H. Ma and P.-Y. Woo, “Calibration-free robotic eye-hand coordination based on an auto disturbance-rejection controller,” *IEEE Transactions on Robotics and Automation*, Vol. 20, No. 5, pp. 899-907, 2004.

[15] G. López-Nicolás, C. Sagüés, J.J. Guerrero, D. Kragic and P. Jensfelt, “Nonholonomic epipolar visual servoing,” *in Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, 2006, pp. 2378-2384.

[16] Y. Ma, J. Košecká and S. S. Sastry, “Vision guided navigation for a nonholonomic mobile robot,” *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 3, pp. 521-536, 1999.

[17] J.-B. Coulaud, G. Campion, G. Bastin, and M. D. Wan, “Stability analysis of a

vision-based control design for an autonomous mobile robot," *IEEE Transactions on Robotics,* Vol. 22, No. 5, pp. 1062-1069, 2006.

[18] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki, "Visual servoing for non-holonomic mobile robots," in *Proceedings of IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, Munich, Germany, 1994, pp. 1133-1140.

[19] R. Pissard-Gibollet and P. Rives, "Applying visual servoing techniques to control a mobile hand-eye system," in *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Aichi, Japan, 1995, pp. 166-171.

[20] F. Conticelli, B. Allotta and P. K. Khosla, "Image-based visual servoing of nonholonomic mobile robots," in *Proceedings of IEEE 38th Conference Decision and Control*, Phoenix, Arizona, USA, 1999, pp. 3496-3501.

[21] D. Burschka, J. Geiman and G. Hager, "Optimal landmark configuration for vision-based control of mobile robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003, pp. 3917-3922.

[22] H. Zhang and J. P. Ostrowski, "Visual motion planning for mobile robots," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 2, pp. 199-208, 2002.

[23] T. Nierobisch, W. Fischer, and F. Hoffmann, "Large view visual servoing of a mobile robot with a pan-tilt camera," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3307-3312.

[24] J. Chen, W. E. Dixon, D. M. Dawson and M. McIntyre, "Homography-based visual servo tracking control of a wheeled mobile robot," *IEEE Transactions on Robotics*, Vol. 22, No. 2, pp. 407-416, 2006.

[25] Y. Fang, W. E. Dixon, D. M. Dawson and P. Chawda, "Homography-based visual servo

regulation of mobile robots," *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, Vol. 35, No. 5, pp. 1041-1049, 2005.

[26] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Transactions on Robotics*, Vol. 23, No. 1, pp. 87-100, 2007.

[27] G. L. Mariottini, D. Prattichizzo, and G. Oriolo, "Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, USA, 2006, pp. 538-544.

[28] T. Goedemé, M. Nuttin, T. Tuytelaars, L. Van Gool, "Omnidirectional Vision based Topological Navigation," to appear in *International Journal on Computer Vision and International Journal on Robotics Research*, Joint Special Issue of IJCV and IJRR on Vision and Robotics, 2007.

[29] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision-based formation control framework," *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, pp. 813-825, 2002.

[30] R. Vidal, O. Shakernia, and S. Sastry, "Following the flock [formation control]," *IEEE Robotics and Automation Magazine*, Vol. 11, No. 4, pp. 14-20, 2004.

[31] J. A. Borgstadt and N. J. Ferrier, "Interception of a projectile using a human vision-based strategy," in *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, USA, 2000, pp. 3189-3196.

[32] L. Freda and G. Oriolo, "Vision-based interception of a moving target with a nonholonomic mobile robot," *Journal of Robotics and Autonomous Systems*, Vol. 55, No.

6, pp. 419-432, 2007.

[33] H. Y. Wang, S. Itani, T. Fukao and N. Adachi, “Image-based visual adaptive tracking control of nonholonomic mobile robots,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Maui, Hawaii, USA, 2001, pp. 1-6.

[34] K.-T. Song and C.-C. Chien, “Visual tracking of a moving person for a home robot,” *Journal of Systems and Control Engineering*, Vol. 219, No. 14, pp. 259-269, 2005.

[35] E. Malis and S. Benhimane, “A unified approach to visual tracking and servoing,” *Journal of Robotics and Autonomous Systems*, Vol. 52, No. 1, pp. 39-52, 2005.

[36] Y. Han and H. Hahn, “Visual tracking of a moving target using active contour based SSD algorithm,” *Journal of Robotics and Autonomous Systems*, Vol. 53, No. 3-4, pp. 265-281, 2005.

[37] C.-Y. Tsai and K.-T. Song, “Face tracking interaction control of a nonholonomic mobile robot,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2006, pp. 3319-3324.

[38] B. Bayer, Color imaging array, U.S. Patent 3,971,065, 1976.

[39] S.-C. Pei and I.-K. Tam, “Effective color interpolation in CCD color filter arrays using signal correlation,” *IEEE Transactions on Circuits and Systems Video Technol.*, Vol. 13, No.6, pp. 503-513, June 2003.

[40] L.-L. Chang and Y.-P. Tan, “Effective use of spatial and spectral correlations for color filter array demosaicking,” *IEEE Transactions on Consumer Electronics*, Vol. 50, No.1, pp.355-365, January 2004.

[41] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd edition, NJ: Prentice-Hall, 2002.

[42] H. Stark and Y. Yang, *Vector Space Projections: A Numerical Approach to Signal and Image Processing, Neural Nets, and Optics*, New York, John Wiley&Sons, INC., 1998.

[43] P. S. Maybeck, *Stochastic Models, Estimation, and Control Volume 1*, New York, Academic Press, INC., 1979.

[44] W. I. Grosky and L. A. Tamburino, "A unified approach to the linear camera calibration problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 7, pp. 663-671, 1990.

[45] T.-C. Lee, C.-Y. Tsai and K.-T. Song, "Fast parking control of mobile robots: a motion planning approach with experimental validation," *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 5, pp. 661-676, 2004.

[46] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[47] A. Habed and B. Boufama, "Camera self-calibration: a new approach for solving the modulus constraint," in *Proceedings of IEEE International Conference on Pattern Recognition*, Cambridge, UK, 2004, pp. 116-119.

[48] W. J. Rugh and J. S. Shamma, "Survey paper: Research on gain scheduling," *Automatica*, Vol. 36, No. 10, pp. 1401-1425, 2000.

[49] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[50] J. D. Schutter, J. D. Geeter, T. Lefebvre and H. Bruyninckx, "Kalman filters: a tutorial," *Journal A*, Vol. 40, No. 4, pp. 52-59, 1999.

[51] O. V. Korniyenko, M. S. Sharawi and D. N. Aloi, "Neural network based approach for

tuning Kalman filter," in *Proceedings of IEEE International Conference on Electro/Information Technology*, Lincoln, Nebraska, 2005, pp. 1-5.

[52] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1989.

[53] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image postprocessing using local color ratios," *IEEE Transactions on Circuits and Systems Video Technol.*, Vol. 14, No. 6, pp.914-920, 2004.

[54] M. Mahy, E. Van Eyckden, and O. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV," *Color Res. Applicat.*, Vol. 19, No. 2, pp. 105-121, 1994.

[55] J. F. Hamilton and J. E. Adams Jr., "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent 5 629 734, May 1997.

[56] The Experimental Raw Photo Editor. [Online]. Available:http://www.rawtherapee.com/

[57] The experiment video website. [Online]. Available: http://isci.cn.nctu.edu.tw/video/ChiYi_Thesis/

[58] K. Liu, R. E. Skelton and K. Grigoriadis, "Optimal controllers for finite wordlength implementation," *IEEE Trans. on Automatic Control*, Vol. 37, No. 9, pp. 1294-1304, 1992.

[59] H.-J. Ko and W.-S. Yu, "Guaranteed robust stability of the closed-loop systems for digital controller implementations via orthogonal Hermitian transform," *IEEE Trans. on System, Man, and Cybernetics-Part B: Cybernetics*, Vol. 34, No. 4, pp. 1923-1932, 2004.

[60] K.-T. Song, J.-S. Hu, C.-Y. Tsai, C.-M. Chou, C.-C. Cheng, W.-H. Liu, and C.-H. Yang, "Speaker attention system for mobile robots using microphone array and face tracking," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando,

Florida, 2006, pp. 3624-3629.

[61] G. R. Bradski and S. Clara, “Computer vision face tracking for use in a perceptual user interface,” *Intel Technology Journal*, Vol. 2, No. 2, pp. 1-15, 1998.

Vita

姓名: 蔡奇謚

性別: 男

生日: 中華民國 67 年 3 月 25 日

籍貫: 台灣省高雄縣

論文題目:中文: 影像感測器之色彩濾波陣列補插與輪式機器人之視覺追蹤控制設計

英文: Color Filter Array Interpolation for an Image Sensor and Visual Tracking Control Design of a Wheeled Mobile Robot

學/經歷:

1. 民國 85 年 6 月　高雄市立高雄高工電機科畢業

2. 民國 89 年 6 月　國立雲林科技大學電機工程學系畢業

3. 民國 91 年 6 月　國立雲林科技大學電機工程研究所畢業

4. 民國 91 年 9 月　國立交通大學電機與控制工程研究所博士班

5. 民國 95 年 10 月至民國 96 年 4 月　比利時魯汶大學機械工程系短期研究員

Publication List

Journal paper

[1] Ti-Chung Lee, Chi-Yi Tsai and Kai-Tai Song, "Fast Parking Control of Mobile Robots: A Motion Planning Approach with Experimental Validation," *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 5, pp. 661-676, 2004. [2 點].

[2] Chi-Yi Tsai and Kai-Tai Song, "Heterogeneity-Projection Hard-Decision Color Interpolation Using Spectral-Spatial Correlation," *IEEE Transactions on Image Processing*, Vol. 16, No. 1, pp. 78-91, 2007. [4 點].

[3] Chi-Yi Tsai and Kai-Tai Song, "A New Edge-Adaptive Demosaicing Algorithm for Color Filter Arrays," *Journal of Image and Vision Computing*, Vol. 25, No. 9, pp. 1495-1508, 2007. [2 點].

[4] Chi-Yi Tsai and Kai-Tai Song, "Visual Tracking Control of a Wheeled Mobile Robot with System Model and Velocity Quantization Robustness," *IEEE Transactions on Control Systems Technology*, May 15, 2008 (Accepted). [Brief Paper, 2 點].

[5] Chi-Yi Tsai and Kai-Tai Song, "Visual Tracking Control of a Nonholonomic Mobile Robot with Image Noise and Occlusion Robustness," *Journal of Image and Vision Computing*, December 14, 2007 (Revised).

[6] Chi-Yi Tsai, Kai-Tai Song, Xavier Dutoit, Hendrik Van Brussel and Marnix Nuttin, "Robust Visual Tracking Control System of a Mobile Robot Based on Dual-Jacobian Model," *Journal of Robotics and Autonomous Systems*, May 14, 2008 (Revised)

[7] Chi-Yi Tsai, Xavier Dutoit, Kai-Tai Song, Hendrik Van Brussel and Marnix Nuttin, "Robust Face Tracking Control of a Mobile Robot Using Self-Tuning Kalman Filter and Echo State Network," *Submitted to Asian Journal of Control*, March, 2008. (Invited paper)

Patent

[1] Chi-Yi Tsai and Kai-Tai Song, "Heterogeneity-Projection Hard-Decision (HPHD) Color Interpolation Method for Digital Cameras," Taiwan Patent No. I265726. [1 點].

[2] Chi-Yi Tsai and Kai-Tai Song, "Heterogeneity-Projection Hard-Decision Interpolation Method for Color Reproduction," USA Patent US20080062479 (Published).

[3] Chi-Yi Tsai, Wei-Hsiang Chen, Kai-Tai Song and Sheng-Fun Lin, “Edge-Enhanced JPEG image compression method,” Taiwan Patent 200823804 (Published).

[4] Chi-Yi Tsai, Fu-Sheng Huang, Chen-Yang Lin, Zhi-Sheng Lin, Chun-Wei Chen and Kai-Tai Song, “Robot of living aid,” Taiwan and USA Patents 2007 (Submitted).

[5] Chi-Yi Tsai, Fu-Sheng Huang, Chen-Yang Lin, Zhi-Sheng Lin, Chun-Wei Chen and Kai-Tai Song, “Robot of living aid,” USA Patent 2008 (Submitted).

Conference paper

[1] Ti-Chung Lee, Chi-Yi Tsai and Kai-Tai Song, “A Motion Planning Approach to Fast Parking Control of Mobile Robots,” *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 2003, Vol. 1, pp. 14-19.

[2] Chi-Yi Tsai and Kai-Tai Song, “Visual Tracking Control of a Mobile Robot Using a New Model in Image Plane,” *IEEE International Conference on Advanced Robotics*, Seattle, WA, USA, July 2005, pp. 540-545.

[3] Chi-Yi Tsai and Kai-Tai Song, “Robust Visual Tracking Control of Mobile Robots Based on an Error Model in Image Plane,” *IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Canada, July 2005, pp. 1218-1223.

[4] Chi-Yi Tsai and Kai-Tai Song, “A New Demosaicing Algorithm for Effectively Reducing Color Artifacts,” *18th IPPR Conference on Computer Vision, Graphics and Image Processing*, Taipei, Taiwan, August 2005, pp. 14-19.

[5] Chi-Yi Tsai and Kai-Tai Song, “Demosaicing: Heterogeneity-Projection Hard-Decision Adaptive Interpolation Using Spectral-Spatial Correlation,” *IS&T/SPIE 18th Annual Symposium Electronic Imaging Science and Technology*, San Jose, California USA, January 2006, Vol. 6069, pp. 1-10.

[6] Kai-Tai Song, Jwu-Sheng Hu, Chi-Yi Tsai, Chung-Min Chou, Chieh-Cheng Cheng, Wei-Han Liu, and Chia-Hsing Yang, “Speaker Attention System for Mobile Robots Using Microphone Array and Face Tracking,” *IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006, pp. 3624-3629.

[7] Chi-Yi Tsai and Kai-Tai Song, “Face Tracking Interaction Control of a Nonholonomic Mobile Robot,” *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006, pp. 3319-3324.

[8] Chi-Yi Tsai and Kai-Tai Song, “Robust Visual Tracking Control of a Nonholonomic Mobile Robot with a Tilt Camera Platform,” *Proceedings of Automatic Control*

Conference, Tamsui, Taiwan, November 2006, pp. 36-41.

[9] Chi-Yi Tsai, Kai-Tai Song, Xavier Dutoit, Hendrik Van Brussel and Marnix Nuttin, “Robust Mobile Robot Visual Tracking Control System Using Self-Tuning Kalman Filter,” *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, Florida, June 2007, pp. 161-166.

[10] Chi-Yi Tsai and Kai-Tai Song, “Dynamic Visual Tracking Control of a Mobile Robot with Occlusion Robustness,” *Proceedings of Automatic Control Conference*, Taichung, Taiwan, November 2007, ID: O108.

[11] Kai-Tai Song, Chi-Yi Tsai, Fu-Sheng Huang, Jung-Wei Hong, Chen-Yang Lin, Chun-Wei Chen and Zhi-Sheng Lin, “Development of a Robot to Help Elderly People at Home,” *Proceedings of Automatic Control Conference*, Taichung, Taiwan, November 2007, ID: 5190.

[12] Chi-Yi Tsai, Xavier Dutoit, Kai-Tai Song, Hendrik Van Brussel and Marnix Nuttin, “Visual State Estimation Using Self-Tuning Kalman Filter and Echo State Network,” *IEEE International Conference on Robotics and Automation*, Pasadena, California, May 19-23, 2008, pp. 917-922.

[13] Kai-Tai Song, Chi-Yi Tsai, Fu-Sheng Huang, Jung-Wei Hong, Chen-Yang Lin, Chun-Wei Chen and Zhi-Sheng Lin, “Development of the Robot of Living Aid: RoLA,” *IEEE International Conference on Automation and Logistics*, Qingdao, Chian, September 1-3, 2008. (Submitted)

[14] Kai-Tai Song, Chi-Yi Tsai, and Cheng-Hsien Chiu Huang, “Multi-Robot Cooperative Sensing and Localization,” *IEEE International Conference on Automation and Logistics*, Qingdao, Chian, September 1-3, 2008. (Submitted)

Honors

[1] 許晉淮, 蔡奇謚, 陳俊瑋, “即時人臉追蹤系統,” 2006德州儀器亞洲區DSP應用競賽台灣區分區競賽, 系統應用組-特優獎.