Binocular Geometry and Camera Motion Directly from Normal Flows

YUAN, Ding

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

in

Automation and Computer-Aided Engineering

October 2008

Thesis/Assessment Committee

Professor LIU, Yun-hui (Chair)
Professor CHUNG, Chi-kit Ronald (Supervisor)
Professor DU, Ru-xu (Committee Member)

ABSTRACT

Active vision systems are about mobile platform equipped with one or more than one cameras. They perceive what happens in their surroundings from the image streams the cameras grab. Such systems have a few fundamental tasks to tackle – they need to determine from time to time what their motion in space is, and should they have multiple cameras, they need to know how the cameras are relatively positioned so that visual information collected by the respective cameras can be related. In the simplest form, the tasks are about finding the motion of a camera, and finding the relative geometry of every two cameras, from the image streams the cameras collect.

The relative motion between a camera and the imaged environment generally induces a flow field in the image stream captured by the camera. The flow field, which is about motion correspondences of the various image positions over the image frames, is referred to as the optical flows in the literature. If the optical flow field of every camera can be made available, the motion of a camera can be readily determined, and so can the relative geometry of two cameras. However, due to the well-known aperture problem, directly observable at any image position is generally not the full optical flow, but only the component of it that is normal to the iso-brightness contour of the intensity profile at the position. The component is widely referred to as the normal flow. It is not impossible to infer the full flow field from the normal flow field, but then it requires some specific assumptions about the imaged scene, like it is smooth almost everywhere etc.

This thesis aims at exploring how the above two fundamental tasks can be tackled by operating on the normal flow field directly. The objective is, without the full flow inferred explicitly in the process, and in turn no specific assumption made about the

imaged scene, the developed methods can be applicable to a wider set of scenes. The thesis consists of two parts. The first part is about how the inter-camera geometry of two cameras can be determined from the two monocular normal flow fields. The second part is about how a camera's ego-motion can be determined by examining only the normal flows the camera observes.

On determining the relative geometry of two cameras, there already exist a number of calibration techniques in the literature. They are based on the presence of either some specific calibration objects in the imaged scene, or a portion of the scene that is observable by both cameras. However, in active vision, because of the "active" nature of the cameras, it could happen that a camera pair do not share much or anything in common in their visual fields. In the first part of this thesis, we propose a new solution method to the problem. The method demands image data under a rigid motion of the camera pair, but unlike the existing motion correspondence-based calibration methods it does not estimate the optical flows or motion correspondences explicitly. Instead it estimates the inter-camera geometry from the monocular normal flows. Moreover, we propose a strategy on selecting optimal groups of normal flow vectors to improve the accuracy and efficiency of the estimation.

On determining the ego-motion of a camera, there have been many previous works as well. However, again, most of the works require to track distinct features in the image stream or to infer the full optical flow field from the normal flow field. Different from the traditional works, utilizing no motion correspondence nor the epipolar geometry, a new method is developed that operates again on the normal flow data directly. The method has a number of features. It can employ the use of every normal flow data, thus requiring less texture from the image scene. A novel formulation of what

the normal flow direction at an image position has to offer on the camera motion is given, and this formulation allows a locus of the possible camera motion be outlined from every data point. With enough data points or normal flows over the image domain, a simple voting scheme would allow the various loci intersect and pinpoint the camera motion.

We have tested the methods on both synthetic image data and real image sequences. Experimental results show that the developed methods are effective in determining inter-camera geometry and camera motion from normal flow fields.

摘要

主動視覺系統一般都有一個或者幾個相機構成，它們被安置在自由運動的工作平臺上。視覺系統通過相機捕捉的圖像序列感知外界的變化。這種系統有一些基本的問題需要解決——需要時時地判斷它們各自的運動情況，如果視覺系統包括多個相機，我們還需要知道相機之間的相對位置，從而可以收集到各個相機之間互相關聯的視覺信息。簡單來說，這些問題就是從相機捕捉的圖像序列中估算相繼的運動和估算相機之間的幾何參數。

相機之間的相對運動和成像的環境的變化導致了相機捕捉的圖像序列的流場。關於圖像序列中像點位置的運動匹配在文獻中叫做光流。如果可以得到每個相機的光流場，相機的運動就可以被馬上估算出來，因此各個相機之間的幾個參數也可以被估算出來。然而，由於存在著名的“小孔問題”，通常情況下，從像點可以直接觀察到的並不是光流，而只是光流的一個分量，即像點位置所在的垂直于圖像灰度圖的等亮度綫的那個法向分量。這個分量通常被稱爲法向流。從法向流場可以推斷出光流場，但是需要一些特定的假設，例如成像場景處處平滑等等的約束條件。

本篇論文致力於從法向流場入手，研究如何直接解決上述的兩個基本問題。我們的研究目標是提出一個應用更廣泛的算法，使得在計算過程中不需要估計光流場，因此不需要利用關於成像場景中的特定假設。本篇論文包括兩部分：第一部分工作是從相機的法向流場直接估算兩個相機之間的幾何參數；第二部分工作是通過相機的法向流直接計算相機的運動參數。

文獻中已經有很多關於確定兩個相機之間的幾何參數的相機標定的算法。他們或是基於成像場景中特定的標定參照物，或是基於兩個相機都能觀察到的場景的重疊部分來做計算的。然而在主動視覺系統中，由於相機的這個“主動”的特

性，一對相機的視場裏面可能沒有任何重疊部分。在本篇論文的第一部分，我們提出了新的方法來解決這個問題。我們的算法要求這對相機作剛體運動的的時候紀錄圖像信息，但是不同于現在的基於運動匹配的標定算法，我們的算法不需要估計光流或者建立運動匹配。取而代之的，我們從單目相機的法向流直接估計相機之間的幾何參數。更多的，我們提出了如何選擇法向流的組合，從而提高算法的準確性和高效性。

同樣的現在已經有很多關於計算相機運動的研究工作。然而，大部分的算法需要在圖像序列中追蹤顯著特徵點，或者需要由法向流推斷光流。和以往的傳統工作不同的是，我們的新算法直接從法向流入手，避免了建立運動匹配和內極綫幾何。我們的算法有如下幾個特點。它可以充分利用每一個法向流，因此對成像場景沒有紋理的要求。我們建立了如何從一個像點的法向流推斷相機運動的新的方法。這一方法使得每一數據點的數據都可以提供相機可能的運動的所在軌跡。如果圖像可以提供足夠的數據點或者法向流，相機的運動可以通過不同的軌跡相交的重疊區域來進行投票得到。

我們用模擬圖像數據和真實圖像數據對我們的上述算法進行測試。實驗結果證明了我們的算法可以有效的利用法向流計算相機之間的幾何參數和相機運動參數。

ACKNOWLEDGEMENTS

A number of people have contributed in various ways to helping me develop and formulate the ideas presented in my dissertation.

My deepest thanks goes to my supervisor, Professor CHUNG, Chi-kit Ronald , who introduced me to the field of computer vision, who has walked me through all the stages of the writing of this thesis, and who supported and encouraged me throughout the years.

I am also very grateful to Professor DU, Ru-xu, Professor LIU, Yun-hui, and Professor HUNG, Y.S. for their serving as my committee members and their comments in my research.

In addition, I would like thank my CVL lab members, Mr. He Yong, Dr. Wang Wei, Mr. Song Zhan, Dr. Liang Bo-dong, Mr. Chun Chun-nam, Mr. Chim Ho-Ming, Dr. Cheng Jun, Miss Dong Mei and Mr. Zhao Ming for the discussions and their help in my research work. I also wish to express my thanks to my best friends Ruo-li, Leng Jing, Xin-ping, Shen Hao, Gao Xin, He-sheng, Nian-feng, Guang-yi, Sunny, Xue-yan, Li Qi, Sun Chen-yu for friendship.

Especially, I would like to express my appreciation to my parents, my brother and my sister-in-law for their love and support throughout the years.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

Normal flow, the local gradient of the intensity information in the video, could be calculated directly from a captured image stream. Combined with additional artificial constraints, such as smoothness constraint and continuity constraint etc., normal flow is often utilized to develop algorithms to calculate camera motion parameters or inter-camera geometry of an active vision system. However, these artificial constraints are not always realistic when dealing with the image-sequence-pictured real world. Therefore, this thesis aims at exploring what could be observed in an active vision system when these restrictive assumptions are not applied as normal flow is the only source of information available. This thesis consists of two parts. First, a method to determine the inter-camera geometry of two cameras from two monocular normal flow fields is presented. Second, a novel method to estimate the camera's ego-motion by using direction information of monocular normal flows is proposed.

1.1 Background

Multi-camera system that allows relative camera movement has the advantage of dynamically configurable visual coverage. With cameras becoming more affordable these years, such active systems become more widely used and manifested as active binocular heads in the field of robotics, active camera networks in surveillance systems and others. The respective cameras in the system generally collect visual information independently. In order to build a relationship between different channels of visual information, no matter quantitatively or qualitatively, it is required that the inter-camera

geometry should be made known from time to time. This task, in the simplest form, is about determining the relative geometry of every two cameras in the multi-camera network system and is referred as inter-camera geometry or binocular geometry in this research.

A number of methods have been proposed in the literature for determining the inter-camera geometry. Some of the methods require the presence of specific objects in the scene [Malis, 2002] [Unal, 2007] [Takahashi, 1988], and some tackle the problem by exploiting the properties of vanishing lines and vanishing planes in the image data [Jaynes, 2004] [Junejo, 2006]. However, a problem of these proposed methods is that, the applicability is restricted to certain scenes. One widely adopted approach to overcome this problem is to make use of cross-camera feature correspondences[Zhang, 1996] [Bjorkman, 2002]. Another approach is to use motion correspondences in the respective cameras [Dornaika, 2001B, 2003] [Ma, 1996] [Neubert, 2002]. However, establishing cross-camera feature correspondences or motion correspondences is let alone a challenging topic due to the ill-posed nature of establishing the correspondences.

Estimating the relative motion between an observer and an object is a fundamental problem in computer vision. In this research, only the estimation of the relative motion between a moving observer and a static scene is of interest, and is also referred as camera ego-motion estimation. Ego-motion provides useful information for human-computer interaction and vehicle navigation.

There are already abundant research works on ego-motion estimation. The classical approaches attempt to determine the camera motion parameters by establishing and analyzing certain motion correspondences from the video data. Hence, the establishment of the motion correspondences ultimately remains as the key issue for

these classical approaches. There are two major categories in the literature. One is the displacement category[Horn, 1990] [Chipolla, 1993] [Armangué, 2003], which is to track the distinct features across the image frames. The other one is the gradient category [Heikkonen, 1995] [Chena, 2001] [Zhang, 2006], which is to interpolate the full optical flows from the normal flows.

Optical flow is the distribution of apparent velocities of the movement of brightness patterns in the image. The optical flow at an image point cannot be computed independently without introducing additional constraints because the velocity field at each image point has two components while the change in brightness caused by motion at a point yields only one constraint. Consider, for example, a patch of pattern where brightness varies as a function of one image coordinate but not the other. Movement of the pattern in one direction alters the brightness at a particular point, but motion in the other direction yields no change. Thus, the component of movement in the latter direction cannot be determined locally [Horn, 1981]. The phenomenon described above is called the aperture problem. As a consequence of this well-known problem, what is directly observable at any image position is generally not the full optical flow. Instead, it is the projection along the direction of the intensity gradient at this very image position. Optical flows can be inferred from normal flows usually by enforcing some artificial constraints such as smoothness constraint. However, such constraint typically assumes that the image domain is continuous or differentiable in space and time, and this is not always realistic when dealing with the real image data. It is unfortunate that although techniques for computing optical flow have been researched for decades, the current computing techniques still do not yield accurate and dense results.

Approaches based on motion correspondences establishment and optical flow estimation both require that the particular scene contains distinct features or dense texture. However, plenty of distinct features are not always presented in practice. Furthermore, in man-made scenes, dense texture is often rare for the reason that exposure to strong texture for extended period of time is not always welcomed by human eyes.

Another category of methods is the direct methods[Duric, 2000] [Sinclair, 1994] [Silva, 1996, 1997] [Fermüller, 1995A, 1995B, 1998], which attempts to recover camera motion parameters by using normal flows directly. The method proposed in this research could also be categorized into this group.

1.2 Motivation

In a novel work [Fermüller, 1995A, 1995B, 1998], Fermüller and Aloimonos proposed a method (hereafter referred to as the FA method) of determining the ego-motion of a camera directly from normal flows. They first define for any particular 3D direction (or axis that passes through the camera's optical center) a vector field for the entire image space. We shall refer to the axis as the field-inducing axis. Once a field-inducing axis and in turn the accompanying vector field is chosen, some data points in the image data could have the normal flows there parallel or anti-parallel with the field vectors induced at the image positions. Each of such data points will be labeled "+" if the normal flow has a direction the same as that of the field vector induced there, and labeled "-" if it has an opposite direction. Fermüller and Aloimonos showed that the "+"-labeled data points and the "-"-labeled data points generally span two separate regions in the image space, and the boundary in the image space that separates the two regions,

which is either a linear boundary or a quadratic boundary, actually gives a mathematical constraint on the camera motion parameters. In other words, if a sufficient number of field-inducing axes (and the accompanying vector fields) are chosen, a number of boundaries in the image space between the "+"-labeled and "-"-labeled regions can be identified, and a number of constraints can be made available to determine the camera motion precisely.

In this thesis we explore how the above single-camera mechanism can be used for a multiple-camera problem – that of determining the inter-camera geometry.

One important issue is that the FA method operates not from the available data points but from arbitrarily chosen field-inducing axes. With this, not all data points can be utilized, but only those with normal flow directions consistent with the specific vector fields defined by the chosen axes. Different sets of axis choices would thus allow different subsets of the data points to be usable, each subset with a different density of the labeled positions in the image space. Naturally, the denser the usable data points, the more precise can the boundary between the "+" labeled positions and the "-" labeled positions be localized in the image space.

In practice there is a limit on how many field-inducing axes can be used, or else the total computation time will be prohibitive. The choices of the axes are thus crucial. They determine the total number of data points usable in the method and in turn the accuracy in determining the inter-camera geometry. However, no particular scheme was ever offered for choosing the field-inducing axes. In this thesis we provide a scheme of choosing the axes, with the objective of, for any given number of axes, maximizing the number of data points usable in determining the inter-camera geometry.

Estimating the motion parameters of the individual cameras in an active vision system is yet another important issue. As mentioned above, traditional algorithms that are based on both the displacement approaches and the gradient approaches require a particular scene with distinct features or dense texture which are not always available in practice.

Algorithms which attempt to recover the camera motion parameters by using normal flows directly, are referred as direct methods. In [Duric, 2000], Z. Duric et al. proposed a method that is able to determine limited types of camera motion such as z-axis rotation, z-axis translation, lateral translation or pan without estimating the explicit parameters. C. Silva et al. presented a method [Silva, 1996, 1997] that is able to calculate the explicit camera motion parameters. This algorithm typically requires that the magnitude and direction information of the normal flows are known accurately. However, it has been pointed out [Burgi, 2004] [Chen, 2000] that the magnitude component of normal flow, in comparison with the direction component, is less tolerant in its extraction to illumination variations in the image data. Fermüller and Aloimonos proposed a method [Fermüller, 1995A, 1995B, 1998] that allows the direction and magnitude information of the normal flows to be separated for simpler determination of the camera motion parameters. However, dense texture of the imaged scene is the crucial factor which greatly affects the efficiency and precision of the estimated result.

An algorithm is proposed in this research in order to overcome the problems that the above works have encountered. In particular, the direction information of normal flows is the only required input to the proposed algorithm. Given a specific normal flow, the entire 3D space that describes the camera motion can be divided into two halves, and the direction of this normal flow indicates which half the camera motion will fall into.

Consequently, the intersection of all the half spaces that each normal flow votes for will reduce the possibilities of camera motion and eventually pinpoint it. Therefore, the density of the texture within the image domain is not crucial, as long as enough data points with detectable normal flows could be obtained.

1.3 Research Objective

This thesis presents two topics by investigating monocular normal flows. The aim of the first topic is to determine inter-camera geometry of two cameras directly from the monocular normal flows in the respective image streams, without establishing neither cross-camera correspondence nor explicit motion correspondence like optical flow. It is assumed that the intrinsic parameters of the cameras have been determined by camera self-calibration methods proposed by [Dornaika, 2001A] [Heikkila, 1996, 2000] [Maybank, 1992] [Zhang, 1996] [Zhang, 1999] [Bouguet] [Gurdjos, 2005]. The focus of this topic is the estimation of the camera-to-camera geometry. On the other hand, the aim of the second topic of this research is to estimate the camera's ego-motion by using monocular normal flows directly. In practice, a voting scheme similar to Hough Transform that transforms the image position and normal flow direction to possible camera motion in the φ-θ domain will be used. Spherical image space is also adopted in order to analyze camera motion because the spherical image space has no ambiguity in describing camera motion. In the present stage, only pure camera translation and pure camera rotation estimation are investigated and general camera motion estimation will be explored in the future.

Concisely, the objective of this research can be summarized as follows:

(1) A novel algorithm on estimating inter-camera geometry of two cameras by directly using monocular normal flows is proposed and in particular:

- It does not need to establish the motion correspondences, or epipolar geometry. Also it does not require any calibration object, particular structure of the scene, or overlaps across image pairs.
- The scheme of choosing the axes to maximize the number of data points usable in determining the inter-camera geometry is proposed.

(2) A novel algorithm on estimating camera ego-motion by directly using monocular normal flows is also proposed and in particular:

- Spherical image space is adopted to avoid the ambiguity in describing camera motion.
- A voting scheme is presented to locate the camera motion parameters in the two-dimensional φ-θ domain.

1.4 Thesis Outline

The rest of the thesis is organized as follows:

- Chapter 2: **Literature Review**

  In this chapter, previous works on binocular geometry estimation and camera ego-motion estimation are reviewed.

- Chapter 3: **Preliminaries**

  In this chapter, the concepts on optical flow and normal flow are introduced, as well as the concepts on Focus of Expansion (FoE), Focus of Contraction (FoC), Right-hand Axis of Rotation (RAoR) and Left-hand Axis of Rotation (LAoR). Also, the well-known aperture problem is explained. Furthermore, the basic

knowledge on calibrating binocular cameras is briefly introduced at the end of this chapter.

- Chapter 4: **Estimation of the Binocular Geometry from Normal Flows**

  In this chapter, the FA concept is first summarized. By utilizing the FA concept, our novel method on binocular geometry estimation is proposed. The rotational component and translational component of the inter-camera geometry will be estimated respectively. Also, a scheme on maximizing the number of data points usable in the estimation is presented. At the end, experimental results on both synthetic image data and real image sequences are presented.

- Chapter 5: **Estimation of Camera's Ego-motion from Normal Flows**

  In this chapter, the flow vector on the spherical image space is first defined. A scheme to project flow vectors from planar image space to spherical image space is introduced. Next, our strategies in estimating the pure translation and rotation of the camera are presented. Furthermore, the voting scheme for determining the motion parameters in φ-θ domain is proposed. Finally, experiments with both synthetic data and real image data show our methods provide excellent results.

- Chapter 6: **Conclusion and Future Work**

  In this chapter, the contributions and future work are summarized.

CHAPTER TWO

LITERATURE REVIEW

Estimating inter-camera geometry and estimating camera ego motion are both essential problems in the field of computer vision. Especially in the last two decades, more and more research focus on these topics, as the active vision systems have become more widely used in navigation, surveillance etc.. In this chapter, previous works on binocular geometry estimation and camera ego-motion estimation are briefly reviewed.

2.1 Literature Review on Binocular Geometry Estimation

There have been a number of methods proposed in the literature on determining the inter-camera geometry.

A great deal of research on the camera calibration problem could be dated as early as the 1970s [Sobel, 1974]. A well known method for calibrating a camera has been proposed by Tsai [Tsai, 1986]. The method is based on the knowledge of the position of some points in the world and the correspondent projections on the image. It required the camera to be pointed to a calibration grid (that must be accurately prepared). A lot of classical calibration techniques are based on surveying a 3D distribution of control points of known position [Wolf, 1983] [Weng, 1992] [Faugeras, 1993]. The control points must be positioned with extreme precision and distributed over the entire working volume to achieve a high accuracy. Some methods require the presence of specific objects in the scene, such as planar surfaces [Knight, 2000A, 2000B] [Malm, 2001] [Malis, 2002] [Unal, 2007] and cubic objects [Takahashi, 1988]. DeSouza et al. [DeSouza, 2002] utilized a calibration object of known metric structure, with only opting for self-calibration based on multi-view relationships. Such methods constitute

simpler solution mechanisms, but their operability is restricted to certain scenes or applications. Moreover, the overlap across the stereo image pairs is usually necessary for the techniques classified into this group.

Some technologies are developed by using particular camera motions. Especially planar motions are often applied in order to simplify the mathematical model. Moons et al. [Moons, 1996] describes a method based on vanishing point detection through pure translational motions of the stereo rig. One basic observation introduced by Beardsley et al. [Beardsley, 1995] and by Zisserman et al. [Zisserman, 1995] is that the projective and rigid motions of a stereo rig are conjugated. These authors investigated two types of motions: 1) planar motion and 2) general motion. In the first case, the stereo rig is allowed to move in a plane perpendicular to a unique axis of rotation and the plane at infinity is defined by a line at infinity and a point at infinity. The line and point are the same, regardless of the number of motions. In the second case, the plane at infinity can be recovered as the unique eigenvector associated with the double eigenvalue (equal to 1) of a 3D projective transformation. These authors therefore have made a major contribution since they showed for the first time that affine calibration of a stereo rig amounts to a straightforward algebraic property. The solution suggested in [Zisserman, 1995] computes both the epipolar geometry of the stereo rig and the epipolar geometry of the left camera motion. It will be shown that the latter computation is not mandatory. Inspired by Zisserman's direction, Horaud et al. proposed a method for calibrating a stereo pair of cameras using general or planar motions [Horaud, 2000]. They firstly establish the affine calibration via homography, and then metric unit is applied to obtain the explicit calibration parameters. In the recent work, Nedevschi et al. presented a method [Nedevschi, 2007] that is able to perform online estimation of the binocular

geometry by driving a car on a flat and straight road, which is parallel with the longitudinal lane marks. Same as the research based on calibration reference objects introduced above, the techniques based on planar motions or other specific motions are also limited within particular applications.

One widely adopted approach is to make use of cross-camera feature correspondences. These techniques, appearing in the last two decades, do not need a calibration reference object with known metric structure to perform the calibration. And they are more and more popular as the research works on active vision systems are deemed as the promising direction in the field of computer vision. Some early research was proposed by Faugeras et al. [Faugeras, 1994] [Maybank, 1992] and Hartley et al. [Hartley, 1994]. In [Horaud, 1998], Horaud et al. used stereo correspondence across a sequence of stereo pairs. Using different projective reconstructions that are associated with each stereo pair, they proposed an algorithm for the recovery of the camera parameters and the 3D Euclidian shape. One of the state of the art was proposed by Zhang et al. in [Zhang, 1996]. They proposed a method for calibrating a stereo rig by moving it in an environment without using any reference points. They make use of the motion and stereo correspondences across two stereo pairs (one motion of the stereo rig). The only geometric constraint between a pair of uncalibrated images is the epipolar constraint, which has been formulated from a point of view in Euclidian space. Bjorkman et al. presented their real time stereo calibration method based on epipolar constraint [Bjorkman, 2002]. Other works classified into this group include the algorithms [Knight. 2000C] [Hanning, 2004]. However, in active vision systems there is no guarantee of how much overlap is between the visual fields of the cameras, meaning that cross-camera correspondences are not always possible.

Some algorithms tackle the problem by exploiting the properties of vanishing lines and vanishing planes in the image data. In [Jaynes, 2004], they assumed a common ground plane for all cameras, and relative rotation of each camera to the ground plane is computed independently. The motion trajectories of objects tracked in each camera are then reprojected on to a plane in front of the camera frame in order to compute corresponding unwrapped trajectories. Camera-to-ground-plane rotation and plane-to-plane transform computed from the matched trajectories are then used to compute relative transform between a pair of cameras. This method assumes that all cameras are calibrated. It requires motion trajectories on objects, and each camera is considered to be stationary looking at a common ground plane. Junejo et al. proved that only one automatically computed vanishing point and a line lying on any plane orthogonal to the vertical direction are sufficient to infer the dynamic camera network configuration [Junejo, 2006]. However, certain a priori knowledge about the parallel lines (for distinguishing them among the other lines) is often necessary, not to mention the requirement that features as specific as parallel lines or planes must be present in the image data in the first place.

Another existing approach is to make use of motion correspondences in the respective cameras [Dornaika, 2001B, 2003] [Ma, 1996] [Neubert, 2002]. Ma proposed a method [Ma, 1996] that is also categorized into this group of research. However, a planar polygonal object in the scene is suggested so as to simplify the computation of focus of expansion (FoE), which is the point that the translational component of object motion is directed toward in the images when the observer is approaching. Dornaika proposed a method [Dornaika, 2001B], which shows the computation of the intrinsic and extrinsic parameters of the stereo rig can be recovered from the motion correspondences

only, i.e. the monocular fundamental matrices. Suppose that the two cameras are considered at a time. A rigid motion of the camera pair is first conducted, motion correspondences [Dornaika, 2003] in the respective image streams are then established, camera motions **A** and **B** of the two cameras are subsequently determined from the respective sets of motion correspondences, and finally the inter-camera geometry **X** is recovered from the composite transformation relation **AX=XB**. The solution was investigated starting from 1980s [Shiu, 1989] [Park, 1994] [Fassi, 2005]. One challenge of the approach is that due to the well-known aperture problem camera motions **A** and **B** can only be determined up to unknown scales from visual motion data alone, though there have been partial answers [Dornaika, 2003] proposed to tackle the challenge. Motion correspondences are established by either tracking distinct features in the video or interpolating the full optical flows (the field of dense motion correspondences) from the apparent flows in the video data. While tracking distinct features requires the presence of scene features that are distinct enough to have unique correspondences across the motion frames, interpolating the full optical flows from the apparent flow is a task that generally requires certain conditions of the imaged scene. However, a global full flow technique that is able to make realistic assumptions has not yet appeared. Due to the well-known aperture problem, directly observable in an image stream are generally not the full optical flows, but only their projections onto the directions of the local intensity gradients; such apparent flows are widely referred to in the literature as the normal flows. As has been well pointed out in the literature, interpolating the full flows from such partial observations requires the use of certain global assumptions like the scene-smoothness or flow-smoothness assumption, which are generally not applicable to everywhere in the scene. In this work, we explore if inter-camera

geometry can be determined directly from the normal flows without the full flows interpolated in the process.

2.2 Literature Review on Camera's Ego-motion Estimation

Camera's ego-motion, the problem with history in computer vision, has long been a challenge for machine vision researchers. It is still an active research topic recently, as the active vision systems become more and more popular in use.

The methods on camera's ego-motion estimation are usually classified into two major schools in the literature. One of them is the displacement school, in which the camera's ego-motion is estimated by tracking the distinct features across the image frames. The works in this group were dated as early as [Barnard, 1980] [Anandan, 1984]. The distinguished features including points, lines, or contours are firstly extracted from successive frames, then the motion correspondences are used to extract the epipolar geometry of the camera frames, whose decomposition will reveal the camera motion parameters [Lustman, 1987] [Horn, 1990] [Chipolla, 1993]. Armangué et al. reviewed the methods on ego-motion estimation by means of differential epipolar geometry in [Armangué, 2003].

Another major school is the gradient school, in which the camera's ego-motion is often estimated from the full optical flows. The works were dated as early as [Lee, 1980] [Prazdny, 1981] [Bruss, 1983] [Schunck, 1985]. Heikkonen proposed an algorithm [Heikkonen, 1995] for recovery of the 3D motion parameters from an optical flow field. The proposed approach is based on the ideas of Randomized Hough transform (RHT), i.e., the principles of random sampling of velocity vectors and accumulation of motion parameters. Chena et al. presented a robust method [Chena, 2001] to estimate the 3D ego-motion of an observer, by combining the optical flow field

observed with multiple cameras, to avoid the ambiguity of 3D motion recovery due to small field of view and small depth variation. In [Zhang, 2006], Zhang et al. presented a novel algorithm to determine optical flow field with large motion. At first, the translational direction of the observer's motion is recovered by searching a candidate over a discrete space to minimize a residual function. Once the translation has been estimated, the rotation components of the observer's motion can been resolved from the second set of equations by using the least square optimization.

Different from the above approaches categorized into the two major schools, there is also a approach, called direct method, tackling the ego-motion problem by using the image brightness informaiton, normal flows direclty. Our novel method can also be classfied in this group of reserach. Hence, the reviews will be focused on the direct methods in the rest following section.

In [Duric, 2000], Z. Duric et al. proposed a method to derive the qualitative information about the camera motion by using histograms of the normal flow vectors. The direction of normal flow vector is orthogonal to the edges, and the magnitude depends on the difference between the image intensities of the corresponding position on the two neighboring images of an image sequence. The strategy of generating the histogram is similar to "Hough Transform". One normal flow vector votes for its corresponding position in the 2D histogram. A histogram is finished after all the normal flow vectors take their votes. One histogram is drawn for every two neighboring images in an image sequence. That is, for an image sequence with n images, totally (n-1) histogram of normal flow vectors can be obtained. Different types of histograms correspond to different types of camera motion, for example, z-axis rotation, z-axis translation, lateral translation and pan. In this work, they showed that normal flow

vectors can provide qualitative information about the camera motion. However, it is difficult to interpret the complicated motions, such as general motion. Moreover, their work does not concern the analysis on quantitative motions.

In [Sinclair, 1994], D. Sinclair et al. proposed an algorithm to recover the FoE from normal flow, which is tolerant to rotational motion. Some allowance must be made for uncertainty in angular velocity. However, the rotational motion parameters can not be calculated. The efficiency of their algorithm strongly depends on the angular values.

In [Silva, 1996, 1997], C. Silva et al. presented a method for ego-motion estimation uniquely by using normal flows. Different from the works listed above, they calculated the explicit camera motion, including both camera translation and camera rotation. They made use of the image points with their normal flows pointing to specific directions to calculate the ego-motion parameters. The method includes two steps. Firstly, the image points having *circular normal flows* are selected. Circular normal flow is defined as the flow vector perpendicular to a line, which passes through its image position and the image center. This line is named as Ψ-line. The slope of Ψ-line provides a one-dimensional constraint for determining FoE. More precisely, it indicates FoE must locate on the Ψ-line. Secondly, the location of FoE and the rest two components of camera rotation are determined by searching Φ-line. Φ-line is determined by normal flow vectors pointing to specific directions. Camera rotation corresponds to the minimum variance of Φ-line. And FoE is the intersection of the Φ-line and the Ψ-line.

In [Fermüller, 1995A, 1995B, 1998], C. Fermüller and Y. Aloimonos proposed the vector field models (including both spherical image model and planar image model) to classify the normal flows into two groups, positive or negative, according to the

direction information of normal flows. The positive-negative pattern is generated by examining whether it is a positive flow vector or a negative flow vector at each image position. Camera ego-motion parameters are estimated by locating the zero-boundaries on the positive-negative patterns. Finally Fermüller and Aloimonos convert the ego-motion estimation problem to a pattern recognition problem.

CHAPTER THREE

PRELIMINARIES

This chapter aims to introduce the concepts and theories concerned in this thesis. Starting from the motion equations for a monocular observer, the concepts of optical flow (or full flow), normal flow, and the aperture problem are introduced concisely. Then the definitions of the camera intrinsic parameters, extrinsic parameters and binocular geometry will be given.

This chapter is organized as follows. Section 3.1 is an introduction of motion equations. Definitions of camera calibration parameters are given in section 3.2.

3.1 Motion Equations for Monocular Observer

The relative motion of the observer with respect to the scene gives rise to motion of the brightness patterns in the image plane. The instantaneous changes of the brightness pattern in the image plane are analyzed to derive the optical flow field, a two-dimensional vector field reflecting the image displacement.

Optical flow is the apparent motion of brightness patterns in the image. The gradient-based approach proposed by Horn and Schunck [Horn, 1981] is based on the assumption that for a given scene point the intensity E at the corresponding image point remains constant over time. If the scene point **P** projects onto image point (x, y) at time t, and onto image point $(x+\delta x, y+\delta y)$ at time $t+\delta t$, the equation should be:

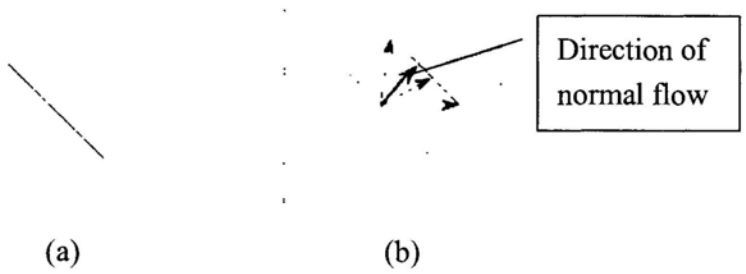
$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \tag{3.1}$$

Now we develop the right hand side of Equation (3.1) in a first order Taylor's series expansion and let $u(x, y)$, $v(x, y)$ be the velocity $(dx/dt, dy/dt)$ of the image point $(x,$

y), then the following optical flow constraint equation for the optical flow (u, v) is obtained [[Horn, 1981] :

$$E_x u + E_y v + E_t = 0 \tag{3.2}$$

From this constraint, the aperture problem [Jain, 1995] can be easily derived. The linear equation defines a line in velocity space ((u-v)-space). Thus only the vector component in the direction of gradient (E_x, E_y) can be computed. E_t is the variance in intensity at image point (x, y) between image frames taken at different time interval. Obviously, E_x, E_y, and E_t can all be computed directly from the image sequence. Thus, for each image point there is only one equation (Equation (3.2)) for solving the two unknowns, u and v, describing the movement of the image point. This is known as the aperture problem, as shown in Fig. 3.1.



(a) (b)

Figure 3. 1 Aperture problem. (a) Line feature observed through a small aperture at time *t*. (b) At time *t*+δ*t* the feature has moved to a new position.

Imagine you are watching a bar, which is moving towards a specific direction, through an aperture that is small compared to the bar at two instants of time, as shown in Fig. 3.1(a). By only watching through the aperture, it is impossible to determine where the bar is moving to. As shown in Fig. 3.1(b), each arrow represents a direction that the bar is possibly moving towards. The only information directly available from the local

measurement is the component of the velocity which is perpendicular to the bar. This component indicates the direction of normal flow.

The optical flow (u, v) can not be calculated by only using Equation (3.2) if there was no more constraints derived. Computation of optical flow is a fundamental problem in processing sequences of images, because optical flow is often a convenient and useful image motion representation. Abundant research works are concentrated on optical flow calculation. The existing numerous computational models can be classified into the following groups: *intensity-based differential methods* [Longuet-Higgins, 1980] [Horn, 1981] [Glazer, 1987A,B] [Nagel, 1989] [Uras, 1988] [Aisbett, 1989] [Tistarelli, 1990] [Schnorr, 1991,1992] [Simoncelli, 1991] [Sobey, 1991] [Bergen, 1992] [Black, 1992] [Fleet, 1995], *frequency-based filtering methods* [Fleet, 1990] [Grzywacz, 1990] [Heeger, 1988] [Watson, 1985], and *correlation-based methods* [Anandan, 1989] [Barnard, 1980] [Kalivas, 1991] [Scott, 1987] [Singh, 1990] [Sutton, 1983].

Equation (3.2) shows that only the vector component in the direction of the gradient (E_x, E_y) can be computed directly from the images, which is called normal flow $\mathbf{u}_n$. Suppose the normalized direction of the intensity gradient direction at the image point (x, y) is $\mathbf{n}$:

$$\mathbf{n} = \frac{[E_x \quad E_y]^{\mathrm{T}}}{\left\|([E_x \quad E_y]^{\mathrm{T}})\right\|} \tag{3.3}$$

Then the normal flow $\mathbf{u}_n$ is:

$$\mathbf{u}_n = ([u \quad v]^{\mathrm{T}} \cdot \mathbf{n})\frac{\mathbf{n}}{\|\mathbf{n}\|} = \left[\frac{-E_x E_t}{E_x^{\,2} + E_y^{\,2}} \quad \frac{-E_y E_t}{E_x^{\,2} + E_y^{\,2}}\right]^{T} \tag{3.4}$$

Consider the monocular imaging situation where the observer is in motion relative to the scene. Suppose the 3D relative velocity of every point $\mathbf{P}= (X, Y, Z)$ with

respect to a camera that moves with the translational velocity **t**= [*U*, *V*, *W*] and rotational velocity **ω**= [α, β, γ], is $\dot{\mathbf{P}} = (\dot{X}, \dot{Y}, \dot{Z})$, which leads to the following equation:

$$\begin{aligned} \dot{X} &= -U - \beta Z + \gamma Y \\ \dot{Y} &= -V - \lambda X + \beta Z \\ \dot{Z} &= -W - \alpha Y + \beta X \end{aligned} \tag{3.5}$$

Here we introduce two concepts, Focus of Expansion (FoE) and Focus of Contraction (FoC). When the camera undergoes a forward translation, **t** is usually referred to as *Focus of Expansion* (FoE), which is the point where all the motion trajectories intersect when they are extended. The camera's motion **t** can also be described as an intersection point of all motion trajectories, *Focus of Contraction* (FoC), if the camera takes a backward translation.

Another two definitions mentioned in the following chapters are *Right-hand Axis of Rotation* (RAoR) and *Left-hand Axis of Rotation* (LAoR). For a camera taking a pure right-hand rotation, **ω** is often represented by a point, *Right-hand Axis of Rotation* (RAoR), about which all the motion trajectories rotate about. Certainly, there is also a corresponding definition for *Left-hand Axis of Rotation* (LAoR), when the camera rotates about a left-hand axis.

Suppose that the camera undergoes a general motion. Using a camera-centered coordinate system, the equation relating the velocity (*u*, *v*) of an image point (*x*, *y*) to the 3D velocity $\dot{\mathbf{P}} = (\dot{X}, \dot{Y}, \dot{Z})$ and the depth *Z* of the corresponding scene point is:

$$\begin{aligned} u &= \frac{-Uf_x + xW}{Z} + \alpha\frac{xy}{f_x} - \beta\left(\frac{x^2}{f_x} + f_x\right) + \gamma y \\ v &= \frac{-Vf_y + yW}{Z} + \alpha\left(\frac{y^2}{f_y} + f_y\right) - \beta\frac{xy}{f_y} - \gamma x \end{aligned} \tag{3.6}$$

where **f**=(f_x, f_y) is the focal length of the camera [Fermüller,1995B].

If the direction of the intensity gradient at image point (x, y) is $\mathbf{n}=(n_x, n_y)$, and the FoE is represented by $(x_0 , y_0) = [\frac{Uf_x}{W}, \frac{Vf_y}{W}]$, then the normal flow $\mathbf{u}_n$ is:

$$\mathbf{u}_n = \frac{W}{Z}\begin{bmatrix} -x_0 + x \\ -y_0 + y \end{bmatrix} \cdot \begin{bmatrix} n_x \\ n_y \end{bmatrix} + \begin{bmatrix} \alpha\frac{xy}{f_x} - \beta(\frac{x^2}{f_x} + f_x) + \gamma y \\ \alpha(\frac{y^2}{f_y} + f_y) - \beta\frac{xy}{f_y} - \gamma x \end{bmatrix} \cdot \begin{bmatrix} n_x \\ n_y \end{bmatrix} \tag{3.7}$$

3.2 Binocular Geometry

Camera calibration is the process of relating the ideal model of the camera to the actual physical device and of determining the position and orientation of the camera with respect to a world reference system.

Depending on the model used, there are different parameters to be determined. The pinhole camera model, which is also the camera model adopted in our work, is broadly used and the parameters to be calibrated are classified in two groups. One is called intrinsic parameters, which describe the internal geometric and optical characteristics of the lenses and the imaging device. Usually intrinsic parameters include focal length, principal point, skew coefficients, and distortions of the lens. The other group is called extrinsic parameters, which actually describe the rotation and translation of the camera with respect to the world coordinate system.

Suppose we have a pair of stereo cameras, each of which has its own extrinsic parameters with respect to the world coordinate system. The binocular geometry is actually referred as the rotation and translation between the two cameras' coordinate systems, which is illustrated in Fig. 3.2.

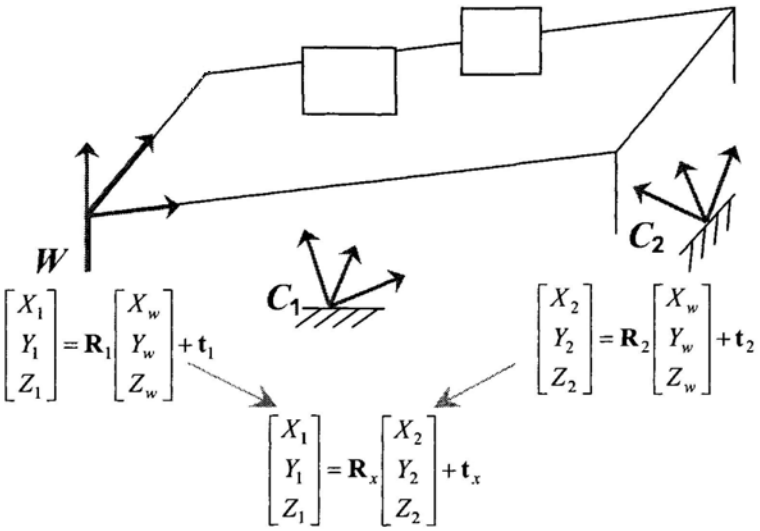


Figure 3. 2 Binocular geometry. C_1 and C_2 represent the coordinates of the two cameras' respectively. W is degfined as the world coordinate system.

As illustrated in the above Fig. 3.2, C_1 and C_2 represent the coordinates of the two cameras' respectively. W is defined as the world coordinate system. The binocular geometry is defined as the translation $\mathbf{t}_x$ and rotation $\mathbf{R}_x$. Via $\mathbf{t}_x$ and $\mathbf{R}_x$, we can describe a point, which was defined in coordinate system C_2, in the new coordinate system C_1, assuming coordinate system C_1 is the reference coordinate system. One of the state-of-art algorithms on camera calibration was proposed by Tsai [Tsai, 1986].

CHAPTER FOUR

ESTIMATION OF THE BINOCULAR GEOMETRY FROM NORMAL FLOWS

For a multi-camera system that permits relative camera movement, an important task is to determine from time to time the relative geometry of the cameras in order to relate the various channels of visual information. There are a number of proposed solutions in the literature with most of them relying upon the establishment of either cross-camera (binocular) correspondences or cross-time (motion) correspondences. However, for the case in which the cameras have little or no overlap in their visual fields, the establishment of cross-camera correspondences would become impossible. Also, the acquisition of explicit motion correspondences also demands certain conditions of the imaged scene which are not always satisfied. In this chapter, we describe a solution of determining the cameras' relative orientation and translation, which requires no overlap in the visual fields of the cameras and thereby no cross-camera correspondence. The solution requires neither optical flow nor any explicit motion correspondence to operate. Instead, the inter-camera geometry is determined from observations that are directly available in the two image streams – the monocular normal flows. Experimental results on synthetic and real image data are shown to illustrate the performance of the solution mechanism

This chapter is organized as follows. The concept of vector field is first introduced and summarized in section 4.1. Then, we propose our novel method on

binocular geometry estimation in section 4.2. Finally, Section 4.3 shows the experimental results on both synthetic image data and real image sequences.

.

4.1 Fundamental of Vector Field

Fermüller and Aloimonos proposed an algorithm that estimates ego-motion of a monocular camera from normal flows directly [Fermüller, 1995A, 1995B, 1998]. For any particular 3D axis that passes through the camera's optical center, they first define a vector field for the entire image space. The models of vector field in both spherical image space and planar image space are then proposed. The following section is a brief introduction to the models of vector field. Moreover, we summarize and unite the vector field models in spherical image space and in planar image space, in order to achieve more explicit mathematical expressions.

4.1.1 Vector Field for the Spherical Image Space

Consider the following spherical representation of the image space of any camera: a unit sphere has its center located at the optical center of the camera, its diameter toward north overlapping the optical axis of the camera, and its north hemisphere as the image space which, if unfolded, represents a plane perpendicular to the optical axis at unit distance from the sphere center. This is another representation of the infinitely large planar image space. For the camera coordinate frame, we shall use the optical axis (the diameter of the sphere toward north) as the **z**-axis, and two other axes that point out of the sphere center and are orthogonal to each other and to the **z**-axis as the **x**- and **y**- axes.

4.1.1.1 Expression of Vector Fields on the Spherical Image Space

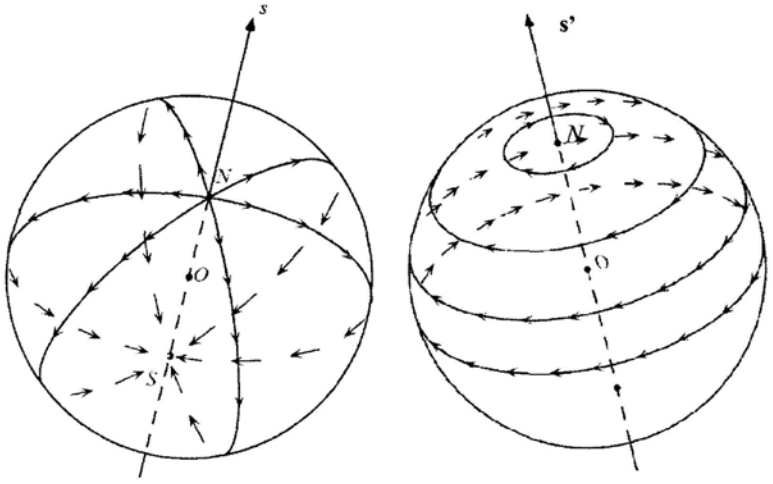
A vector pointing from the sphere center to any particular point **s** on the spherical surface represents an arbitrary direction in the 3D (**x-y-z**) space. Given such an axis, two vector fields can be naturally defined for the entire image space, as described by Fermüller and Aloimonos [Fermüller, 1995A, 1998] in their camera motion determination method. These vector fields are also utilized in our binocular geometry determination. A brief review of the vector fields is given in this section.

Assume a point **s** lies on the spherical surface. The axis through point **s** on the spherical surface defines a vector field for the entire spherical surface. This vector field corresponds to the longitudinal lines of the sphere that have the **s**-axis as the pole-to-pole reference diameter, as shown in Fig. 4.1(a). To be more precise, imagine that the camera is purely translating in the direction of the **s**-axis. Then, optical flows at various positions of the image space will be along the above longitudinal lines in directions that are away from point **s**. Such flows represent one vector field. Specifically, at any image position $\mathbf{p} = [x, y, 1]^T / \left\| [x, y, 1]^T \right\|$, the field vector induced there by **s** is in the direction of $(\mathbf{s} \times \mathbf{p}) \times \mathbf{p}$ on the spherical image space.

Consider another point **s'** which lies on the spherical surface. The axis through point **s'** on the spherical surface could also define another vector field for the entire spherical surface. This vector field corresponds to the latitudinal lines of the sphere, as shown in Fig. 4.1(b). If the camera is purely rotating about the **s'**-axis in the right-handed manner, the optical flows at various positions of the image space will be along the latitudinal lines in directions that are left-handed with respect to the **s'**-axis. Such flows represent another vector field. Specifically, at any image

position $\mathbf{p} = [x, y, 1]^T / \|[x, y, 1]^T\|$, the field vector induced there by **s'** is in the direction of $-(\mathbf{s}' \times \mathbf{p})$ on the spherical image space.

The first longitudinal vector field with respect to the **s**-axis is referred to as the *co-point field* while the second latitudinal vector field with respect to the **s'**-axis is referred as the *co-axis field* and they are shown in Fig. 4.1. It is obvious that different choices of the **s**-axis define different co-point fields and co-axis fields for the image space.



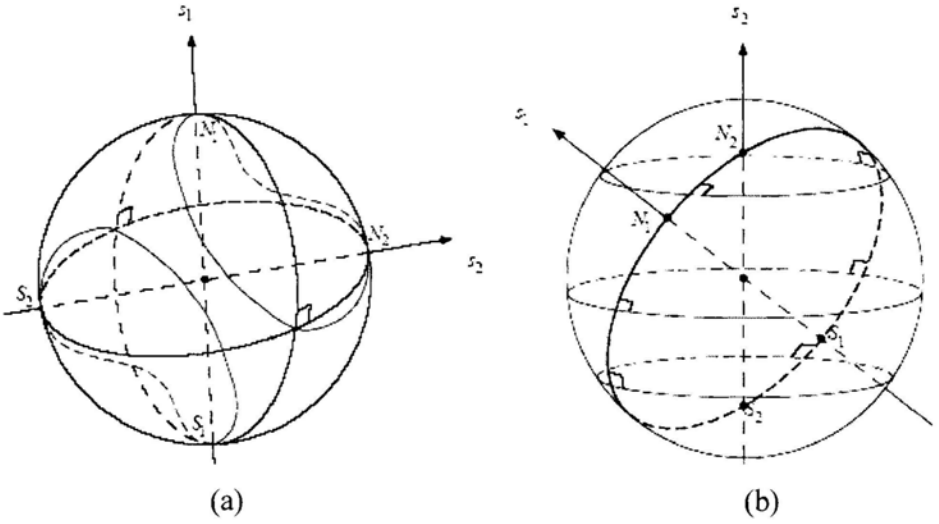
(a) co-point field (b) co-axis field

Figure 4. 1 Illustration of vector fields in the spherical image space (a) A co-point field defined by axis s. At every image point p, the co-point vector is $(\mathbf{s} \times \mathbf{p}) \times \mathbf{p}$, and the direction, that the arrow is pointing at, is the positive direction. (b) A co-axis field vector field defined by axis s'. At every image point p the co-axis is $-(\mathbf{s}' \times \mathbf{p})$, and also the direction that the arrow is pointing at is the positive direction.

4.1.1.2 Positive-negative Patterns for Motion Determination in Spherical Image Space

Now we consider the vector fields defined by two **s**-axes, $\mathbf{s}_1$ and $\mathbf{s}_2$. Each **s**-axis defines a co-point vector field and a co-axis vector field, as shown in Fig.4.2. The locus

of points on the sphere where the $\mathbf{s}_1$-co-point vectors are perpendicular to the $\mathbf{s}_2$-co-point vectors (or where the directions of $\mathbf{s}_1$-co-axis vectors are perpendicular to the directions of $\mathbf{s}_2$-co-axis vectors) constitutes two quadratic curves, as illustrated by Fig.4.2 (a). Similarly as described in Fig.4.2 (b), the $\mathbf{s}_1$-co-axis vector field and the $\mathbf{s}_2$-co-point vector field define a great circle, which is the locus of the points where the two field vectors are perpendicular to each other.



(a) (b)

Figure 4.2 Quadratic curves defined by s-axes on the spherical surface. (a) On the sphere, the green quadratic curves determined by the two co-point vector fields or the two co-axis fields are $(\mathbf{s}_1 \times \mathbf{p}) \cdot (\mathbf{s}_2 \times \mathbf{p}) = 0$ or $(\mathbf{s}_1 \cdot \mathbf{p})(\mathbf{s}_2 \cdot \mathbf{p}) = \mathbf{s}_1 \cdot \mathbf{s}_2$. (b) On the sphere, the purple great circle determined by a co-point vector field and a co-axis field is $(\mathbf{s}_1 \times \mathbf{s}_2) \cdot \mathbf{p} = 0$.

Obviously, the two quadratic curves in Fig.4.2 (a) divide the spherical surface into the positive region (where $(\mathbf{s}_1 \times \mathbf{p}) \cdot (\mathbf{s}_2 \times \mathbf{p}) > 0$) and the negative regions (where $(\mathbf{s}_1 \times \mathbf{p}) \cdot (\mathbf{s}_2 \times \mathbf{p}) < 0$). Similarly the great circle in Fig.4.2 (b) divides the spherical surface into a positive hemisphere (where $(\mathbf{s}_1 \times \mathbf{s}_2) \cdot \mathbf{p} > 0$) and a negative hemisphere (where $(\mathbf{s}_1 \times \mathbf{s}_2) \cdot \mathbf{p} < 0$). It turns out that these vector fields are related to the camera motions.

The motion of the camera can be described by a translational vector **t** and a rotational vector **ω**. Suppose the camera undergoes a pure translation in the direction of **t**, the optical flow at each point induced by this motion on the spherical image is identical to the **t**-co-point vector at its corresponding image position when the **t**-co-point vector field is drawn on the spherical surface. Then, we apply an arbitrary **s**-axis and draw its **s**-co-point vector field on the same spherical surface, and examine where the two quadratic curves $\mathbf{s}\cdot\mathbf{t}-(\mathbf{t}\cdot\mathbf{p})(\mathbf{s}\cdot\mathbf{p})=0$ (the green quadratic curves shown in Fig.4.2 (a)) locate. The two quadratic curves divide the spherical surface into three regions according to whether the expression $\mathbf{s}\cdot\mathbf{t}-(\mathbf{t}\cdot\mathbf{p})(\mathbf{s}\cdot\mathbf{p})$ at each image position is positive or negative. The image point is labeled positive if $\mathbf{s}\cdot\mathbf{t}-(\mathbf{t}\cdot\mathbf{p})(\mathbf{s}\cdot\mathbf{p})>0$, and is labeled negative if $\mathbf{s}\cdot\mathbf{t}-(\mathbf{t}\cdot\mathbf{p})(\mathbf{s}\cdot\mathbf{p})<0$. Then, all the image points with positive labels are merged together to form a positive region and similarly, all points with negative labels are merged together to form a negative region. Therefore, the positive-negative pattern for the pure camera translation is obtained. This method is illustrated in Fig.4.3 (a).

Similarly, when the camera undergoes a pure rotation about a rotational axis **ω**, the optical flow at each point induced by **ω** on the spherical image is identical to the **ω**–co-axis vector at its corresponding image point when the **ω**-co-axis vector field is drawn on the spherical surface. Again, we apply an arbitrary **s**-axis and draw its **s**-co-point vector field on the same spherical surface and examine where the great circle $(\mathbf{s}\times\boldsymbol{\omega})\cdot\mathbf{p}=0$ (the purple great circle shown in Fig.4.2 (b)) locates. The great circle divides the whole spherical surface into two regions according to whether the expression $(\mathbf{s}\times\boldsymbol{\omega})\cdot\mathbf{p}$ at every image point is positive or negative. The image point is labeled positive if $(\mathbf{s}\times\boldsymbol{\omega})\cdot\mathbf{p}>0$, and it is labeled negative if $(\mathbf{s}\times\boldsymbol{\omega})\cdot\mathbf{p}<0$. Then, all the image points with positive labels are merged together to form a positive region and similarly,

all points with negative labels are merged together to form a negative region. The positive-negative pattern for the pure camera rotation is obtained and is shown in Fig.4.3 (b).

For a general camera motion including both pure translation and rotation, the above two positive-negative patterns (Fig.4.3 (a) and Fig.4.3 (b)) are combined together to generate a new positive-negative pattern by enforcing the following rules:

Positive+Positive= Positive;

Negative+ Negative= Negative;

Positve+Negative=Don't know (depends on the structure of the scene)

Therefore, the positive-negative pattern for a camera that undergoes a general motion is shown in Fig.4.3(c).

In summary, for a camera that undergoes a general motion, the positive-negative patterns generated with an arbitrary choice of **s**-co-point vector field are shown in Fig.4.3.

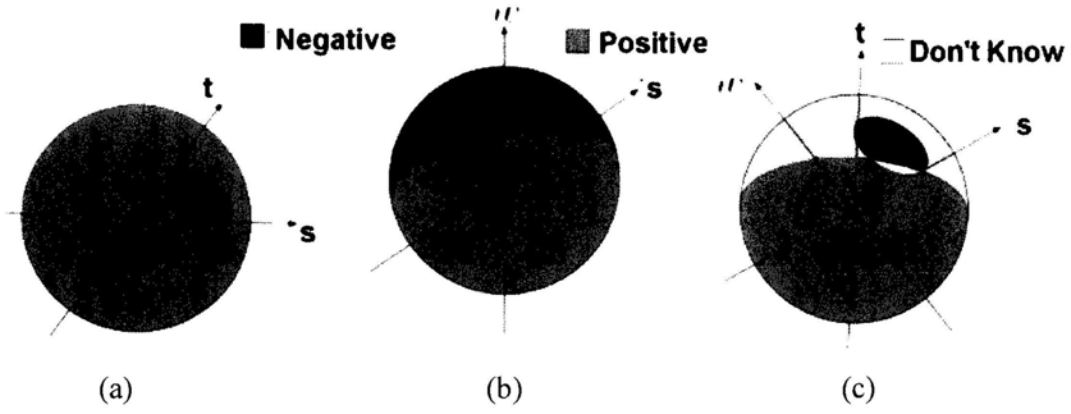


Figure 4.3 s-co-point positive-negative patterns. (a) s-co-point positive-negative pattern for a camera taking pure translation. (b) s-co-point positive-negative pattern for a camera taking pure rotation. (c) s-co-point positive-negative pattern for a camera taking a rigid general motion, including both translation and rotation.

It should be noted that the positive-negative patterns can also be determined by arbitrarily choosing the **s**-co-axis vector field instead of the **s**-co-point vector field as **s**-co-point vector field and **s**-co-axis vector field have equivalent roles in camera motion determination.

4.1.2 Vector Field for the Planar Image Space

If projected from the spherical image space to the planar image space, the co-point vector field will appear as a set of arrows emerging from the same image point – the point projected from point **s** to the planar image space, and that is how the name co-point comes about. On the other hand, the co-axis vector field will appear in the planar image space as a set of conic sections around a particular image point - again the point projected from point **s** to the planar image space through the optical center. Notice that in defining the co-point and co-axis vector fields, only vector directions are considered; vector magnitudes are ignored.

4.1.2.1 Expression of Vector Fields on the Planar Image Space

The planar image space is often used in practice. Thus we give below the algebraic expressions of the co-point and co-axis vector fields in the planar image space. For simplicity, in the subsequent discussion we shall assume that image coordinates in the planar image space have all been normalized by the focal length f of the camera. In such a case an image position $\mathbf{p}$ (a 3-vector) in the spherical image space J is equivalent to the image position $\mathbf{p}' = [\mathbf{I}_2, \mathbf{0}_{1\times 2}]\mathbf{p}/(\mathbf{p}\cdot\mathbf{k})$ (a 2-vector) in the planar image space (image plane) I. Also, a field vector $\mathbf{u}(\mathbf{p})$ (a 3-vector) at image position $\mathbf{p}$ in the image space J

is equivalent to the field vector $\Phi_J^I(\mathbf{u}(\mathbf{p})) = [\mathbf{I_2}, \mathbf{0}_{1\times 2}]\{(\mathbf{p}\times\mathbf{u}(\mathbf{p}))\times\mathbf{k}\}$ (a 2-vector) at the equivalent image position in the image plane *I*.

Consider the image plane perpendicular to the optical axis at unit distance from the optical center. Given any 3D axis $\mathbf{s} = [A, B, C]^{\mathrm{T}}$ which impacts the image plane at the image position (*A/C, B/C*), the co-point vector field of the axis on the image plane is the set of arrows emerging from the image point (*A/C, B/C*). More precisely, the field vector at image position (*x, y*) is in the direction:

$$\Phi_J^I((\mathbf{s}\times\mathbf{p})\times\mathbf{p}) = [\mathbf{I_2}, \mathbf{0}_{1\times 2}]\{(\mathbf{p}\times\{(\mathbf{s}\times\mathbf{p})\times\mathbf{p})\})\times\mathbf{k}\}$$

where $\mathbf{p} \cong [x, y, 1]^{\mathrm{T}}$ and $\mathbf{k} \cong [0,0,1]^{\mathrm{T}}$.

By simple algebraic manipulation the above expression can be simplified to:

$$\mathbf{F}_p(\mathbf{s}, x, y) =_{+} [(x - A/C), (y - B/C)]^{\mathrm{T}}$$

which is the direction of the co-point field vector defined by axis **s** at any image position (*x,y*) in the planar image space. However, in the work of Fermüller and Aloimonos [Fermüller, 1995B], for ease of algebraic manipulation the direction is rotated in the planar image space by -90°. Upon the rotation the field direction at image position (*x, y*) becomes

$$\mathbf{F}_p^{\perp}(\mathbf{s}, x, y) =_{+} [(y - B/C), (-x + A/C)]^{\mathrm{T}} \tag{4.1}$$

We refer to the above field as the *orthogonalized co-point vector field*, so as to distinguish it from the above regular co-point vector field.

The same **s**-axis can also be used to define the co-axis vector field, which is the set of conic sections centered around the point (*A/C, B/C*) on the image plane. More precisely, the field vector at image position (*x, y*) is in the direction:

$$\Phi_J^I(-(\mathbf{s}\times\mathbf{p})) = [\mathbf{I}_2, \mathbf{0}_{1\times 2}]\{(\mathbf{p}\times\{-(\mathbf{s}\times\mathbf{p})\})\times\mathbf{k}\}$$

where $\mathbf{p} \cong [x, y, 1]^T$ and $\mathbf{k} \cong [0,0,1]^T$.

By simple algebraic manipulation the above expression can be simplified to:

$$\mathbf{F}_a(\mathbf{s}, x, y) =_+ [-\frac{B}{C}(1+x^2)+\frac{A}{C}xy+y,\ \ \frac{A}{C}(1+y^2)-\frac{B}{C}xy-x]^T$$

which is the direction of the co-axis field vector defined by axis **s** at any image position (x,y) on the image plane. However, similar to the previous field, in the work of Fermüller and Aloimonos [Fermüller, 1995B], for ease of algebraic manipulation the direction is rotated in the planar image space by 90°. Upon the rotation the field direction at image position (x, y) becomes:

$$\mathbf{F}_a^{\perp}(\mathbf{s}, x, y) =_+ [-\frac{A}{C}(1+y^2)+\frac{B}{C}xy+x,\ \ -\frac{B}{C}(1+x^2)+\frac{A}{C}xy+y]^T \qquad (4.2)$$

We refer to the above field as the *orthogonalized co-axis field* induced by axis **s** at image position (x,y).

Examples of the *orthogonalized* co-point and *orthogonalized* co-axis vector fields are shown in Fig.4.4 where the arrows indicate the field directions assigned to various image positions.

In fact the *orthogonalized* co-axis and *orthogonalized* co-point vector fields have equivalent roles in camera motion determination. However, different vector fields are usually preferred to simplify the positive-negative pattern analysis.

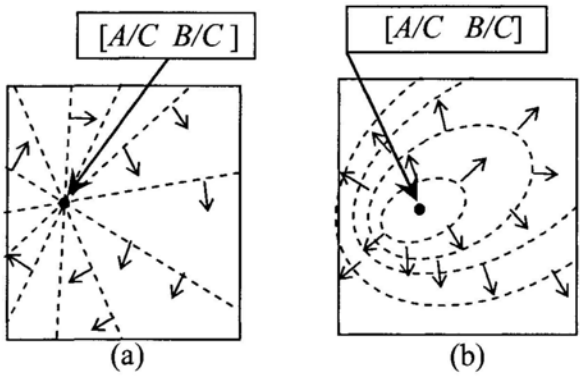


Figure 4.4 Two vector fields definable for the image space by any axis going through the optical center: (a) the orthogonalized co-point vector field, (b) the orthogonalized co-axis vector field, induced by a particular axis $\mathbf{s}=[A,B,C]^{\mathrm{T}}$.

4.1.2.2 Positive-negative Patterns for Motion Determination in Planar Image Space

In [Fermüller, 1995B, 1998] a mechanism is proposed to let the orthogonalized co-point and orthogonalized co-axis vector fields of any arbitrarily chosen **s**-axis be used to determine camera motion directly from normal flows. The mechanism to pave the background for the subsequent discussion is briefly reviewed in the following.

Suppose the camera undergoes a pure rotation, which is described by the vector $\boldsymbol{\omega}=[\alpha \quad \beta \quad \gamma]^{\mathrm{T}}$ in the rotation axis-magnitude representation. We shall refer to this $\boldsymbol{\omega}$ as the axis of rotation (AoR). The optical flows in the image plane induced by the rotation will be conic sections about the point $(\alpha/\gamma, \beta/\gamma)$ where the AoR (or $\boldsymbol{\omega}$) impacts the image plane.

If an axis **s** is chosen to define a co-point field for the image space, and this **s** happens to coincide with $\boldsymbol{\omega}$, on the spherical image space the co-point field of the **s**-axis will be exactly orthogonal to the optical flow field at every image position for the reason that one is about longitudinal lines and the other the latitudinal lines of the above spherical surface with respect to the same axis (**s** or $\boldsymbol{\omega}$) of the sphere. In the planar image

space, it will be that at all image positions the orthogonalized co-point field of axis **s** is more or less parallel to the optical flow field induced by **ω**. In other words, given a choice of the **s**-axis whose accompanying orthogonalized co-point field happens to be more or less parallel with the observed optical flow in the image plane, it is known that the rotation **ω** is in the direction of **s**, and the camera rotation is determined. Of course, in practice there are two issues: optical flow is generally not directly observable, and acquiring the above choice of the **s**-axis generally demands exhaustive search.

If the **s**-axis is chosen only arbitrarily, it generally exhibits an offset from the AoR, and at any image position (x,y) the orthogonalized co-point field direction and the optical flow are generally not parallel, as illustrated by Fig.4.5. Suppose to each image position (x,y) we assign the label "+" if the two field directions there differ by less than 90°, and the label "-" if they differ by more than 90°. Fig.4.5. shows some illustrations: image position I_1 has label "+", and image position I_2 has label "-". Notice that once **s** is chosen, the orthogonalized co-point field is defined, and every image position can be examined if a "+" label or "-" label should be given to it according to the direction of the optical flow there.

It can be shown that given any general choice of the **s**-axis, the "+"-labeled image positions and the "-" -labeled image positions again will occupy two distinct regions of the planar space, and they are separated by a second order curve, which is called *zero-boundary*. The boundary is the locus of all image positions where the orthogonalized co-point field direction makes right angle with the optical flow direction. This zero-boundary is related to the unknown **ω** and the known $\mathbf{s}=[A,B,C]^{\mathrm{T}}$ by the locus of all image positions (x,y) such that $\mathbf{F}_p^{\perp}(\mathbf{s},x,y)\cdot\mathbf{F}_a(\boldsymbol{\omega},x,y)=0$, which can be simplified to the following:

$$k(\frac{A}{C},\frac{B}{C},\alpha,\beta,\gamma,x,y)=x^2(\beta\frac{B}{C}+\gamma)+y^2(\alpha\frac{A}{C}+\gamma)-xy(\alpha\frac{B}{C}+\beta\frac{A}{C})-x(\alpha+\gamma\frac{A}{C})-y(\beta+\gamma\frac{B}{C})+(\alpha\frac{A}{C}+\beta\frac{B}{C})\ =0 \tag{4.3}$$

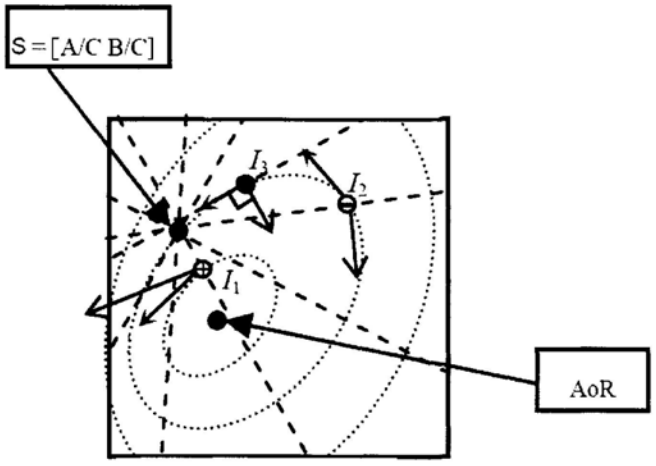


Figure 4.5 Optical flows due to camera rotation, and the orthogonalized co-point vector field of any arbitrary axis s.

As illustrated by Fig.4.6 (a), the zero-boundary is a second order curve. Notice that once an **s**-axis is arbitrarily chosen, most of the image positions can be labeled as either "+" or "-", and the zero-boundary can be identified in the image space by finding the best second order curve that separates the"+"-labeled image positions and the "-"-labeled image positions. Equation (4.3) of the boundary then provides a constraint for the determination of the AoR. In principle, if enough **s**-axes are chosen, enough constraints will be made available for the AoR, and the camera motion can be determined.

Of course, in general only normal flow not full optical flow is directly observable from image data. However, we can still assign label "+" to at least those image positions where the normal flow direction exactly coincides with that of the orthogonalized co-point field vector there. The reason is, normal flow is only the projection of full flow to certain direction (the direction of the intensity gradient), and thus the normal flow and

the full flow cannot have directions differing by more than 90°. At the above image positions, the normal flow direction and the orthogonalized co-point field direction are the same, meaning that the full flow and the orthogonalized co-point field vector must be of directions satisfying the requirement of the "+" label. For a similar reason, we can also assign label "-" to those image positions where the normal flow direction is exactly opposite to that of the orthogonalized co-point field vector. In other words, even though full flow is not observable but only the normal flow, a number of image positions can still be labeled, and the above zero-boundary can still be located though perhaps of compromised precision.

Suppose now that the camera undergoes a pure translation, which is described by the vector $\mathbf{t}=[U, V, W]^{\mathrm{T}}$. The optical flow in the image plane induced by the translation will be arrows emerging from a point called FoE (Focus of Expansion) which is where the 3D vector **t** impacts the image plane.

If an axis **s** is again chosen to define a co-point field for the image space, and this **s** happens to coincide with **t**, the co-point field of the **s**-axis will be exactly the same as the optical flow field if only vector directions are considered. In the planar image space it will be that at all image positions the orthogonalized co-point field of axis **s** is exactly orthogonal to the optical flow induced by **t**. In other words, given a choice of the **s**-axis whose orthogonalized co-point field happens to be exactly orthogonal to the observed optical flow field at all image positions, it is known that the FoE is precisely in the direction of **s**, and the direction of camera translation is determined.

If the **s**-axis is chosen only arbitrarily, it generally exhibits an offset from **t**, and at any image position (x,y) the orthogonalized co-point field direction and the optical flow are generally not exactly orthogonal. Suppose we assign the label "+" to the image

position (x,y) if the two directions differ by less than 90°, and the label "-" if they differ by more than 90°. It can be shown that given any general choice of the **s**-axis, the "+"-labeled image positions and the "-"-labeled image positions again will occupy two distinct regions of the planar space, and they are separated by a straight boundary. The boundary, which is another *zero-boundary* analogous to the previous one, is the locus of all image positions where the orthogonalized co-point field direction makes right angle with the optical flow direction. This zero-boundary is related to the unknown **t** and the known $\mathbf{s}=[A,B,C]^{\mathrm{T}}$ by the locus of all image positions (x,y) such that $\mathbf{F}_p^{\perp}(\mathbf{s},x,y)\cdot\mathbf{F}_p(\mathbf{t},x,y)=0$, which can be simplified to the following:

$$l(\frac{A}{C},\frac{B}{C},\frac{U}{W},\frac{V}{W},x,y)= \quad x(\frac{V}{W}-\frac{B}{C})-y(\frac{U}{W}-\frac{A}{C})+(\frac{U}{W}\frac{B}{C}-\frac{V}{W}\frac{A}{C})=0 \tag{4.4}$$

assuming that W is non-zero, i.e., the camera translation is not restricted to the x-y plane.

Fig. 4.6 (b) presents an example zero-boundary for camera translation, which is a straight line in the image space. Similar to the case of rotation, even though in general only normal flow not full optical flow is directly observable from image data, once an **s**-axis is arbitrarily chosen, we can still assign label "+" to those image positions where the normal flow direction exactly coincides with that of the orthogonalized co-point field vector of the **s**-axis, and label "-" to those image positions where the normal flow direction is exactly opposite to that of the orthogonalized co-point vector. In other words, a substantial number of positions of the image space can still be labeled, and the above zero-boundary can still be located. If enough **s**-axes are chosen, enough constraints will be made available for **t**, and the camera translation can be determined.

A general camera motion has both rotation component **ω** and translation component **t**. Given any choice of the **s**-axis, there are two underlying positive-negative

patterns for the image space: one induced by ($\boldsymbol{\omega}$,**s**), and the other by (**t**,**s**). The trouble is, the two underlying patterns are not individually observable, as the flow components $\mathbf{v}_\omega$ and $\mathbf{v}_t$ induced by $\boldsymbol{\omega}$ and **t** respectively to any image position are only observable as a single total sum $\mathbf{v}=\mathbf{v}_\omega+\mathbf{v}_t$. In other words, the image space can only be labeled with respect to **v** not $\mathbf{v}_\omega$ or $\mathbf{v}_t$.

However, for any arbitrarily chosen **s**-axis we still have the following. At any image position, if the labels from the ($\boldsymbol{\omega}$,**s**)-induced pattern and the (**t**,**s**)-induced pattern are both positive, the flow components $\mathbf{v}_\omega$ and $\mathbf{v}_t$ contributed by $\boldsymbol{\omega}$ and **t** must both be of directions within 90° of the orthogonalized co-point field vector there, and the same can be said about their sum **v**. In other words, given a choice of the **s**-axis, if an image position is labeled positive with respect to the full flow **v** there, it is also labeled positive in the ($\boldsymbol{\omega}$,**s**)-induced pattern and in the (**t**,**s**)-induced pattern. Similarly, if an image position is labeled negative with respect to the full flow **v** there, it is also labeled negative in the ($\boldsymbol{\omega}$,**s**)-induced pattern and the (**t**,**s**)-induced pattern. The image positions whose labels remain unknown are those where the ($\boldsymbol{\omega}$,**s**)-induced pattern and the (**t**,**s**)-induced pattern do not carry the same labels: one says positive and the other negative. This set of image positions are referred to as the "Don't know" region.

Fig. 4.6 (c) displays an image space labeled with respect to the overall flow **v** (or more correctly the normal component of the full flow) under a particular choice of the **s**-axis. There are the positive region, negative region, and the "Don't know" region. Separating the positive and negative regions are the 2nd order zero-boundary for $\boldsymbol{\omega}$ and the linear zero-boundary for **t**. If such boundaries can be located in the image space from the image positions with known labels, Equations (4.3) and (4.4) will provide constraints

for ω and **t**. With enough **s**-axes chosen, enough constraints will be made available to allow ω and **t** to be determined precisely.

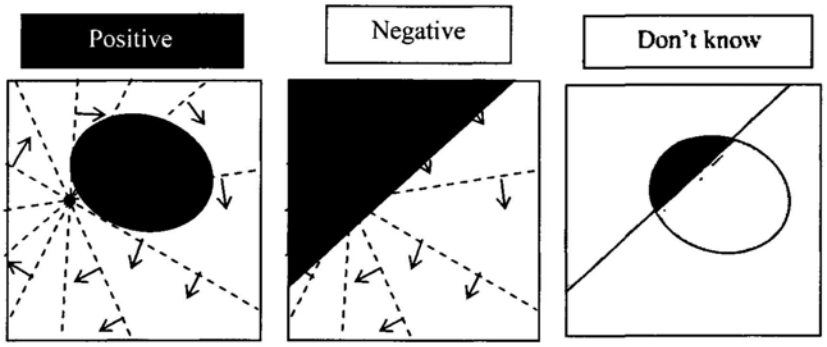


Figure 4.6 Positive-negative patterns in the image space with respect to the orthogonalized co-point field of the same s-axis. (a) The case of pure camera rotation. (b) The case of pure camera translation. (c) The case of general camera motion (with both translation and rotation).

Orthogonalized co-axis field and orthogonalized co-point field are totally equivalent on camera motion determination. Fig.4.7 illustrates the positive-negative patterns with respect to the orthogonalized co-axis vector field of an arbitrary **s**-axis.

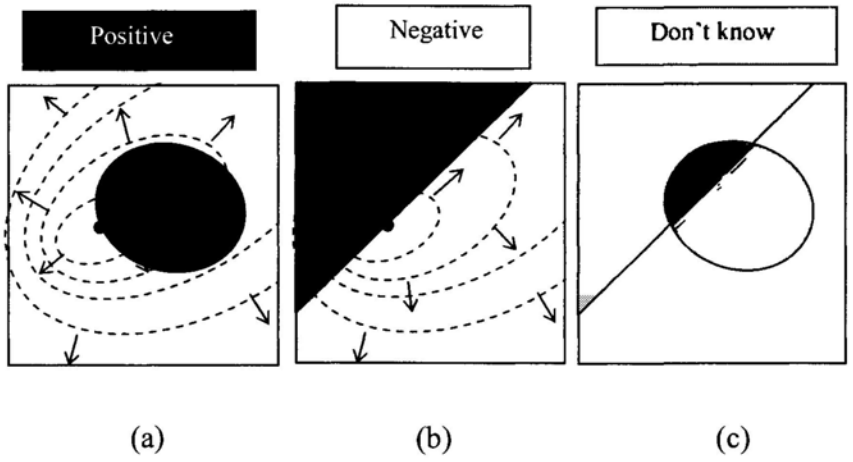


Figure 4.7 Positive-negative patterns in the image space with respect to the orthogonalized co-axis field of the same s-axis. (a) The case of pure camera translation. (b) The case of pure camera rotation. (c) The case of general camera motion (with both translation and rotation).

The essence of the described mechanism is that the original motion parameters determination problem is converted to a pattern recognition problem, namely the identification of the zero-boundaries in the image space under a number of choices of the s-axis. In this work, we address the question of how the mechanism can be borrowed to determine the relative geometry of multiple cameras.

4.2 Inter-camera Geometry Determination

In this section we present a method of determining inter-camera geometry by observing only monocular normal flows in the respective cameras. The method assumes no availability of cross-camera correspondences, meaning that even cameras with no overlap in their visual fields can still be processed.

Suppose the relative geometry of two cameras at a particular configuration of the camera system is to be determined. Our procedure consists of the following. With their relative geometry fixed, the cameras are moved rigidly in space while image streams are collected from the respective cameras, which serve as data for the determination task. Fig. 4.8 shows the relative geometry ($\mathbf{R}_x$, $\mathbf{t}_x$,) that is fixed during the rigid motion of the camera pair. ($\mathbf{R}_A$, $\mathbf{t}_A$,) and ($\mathbf{R}_B$, $\mathbf{t}_B$) represent the motions of camera A and camera B respectively.

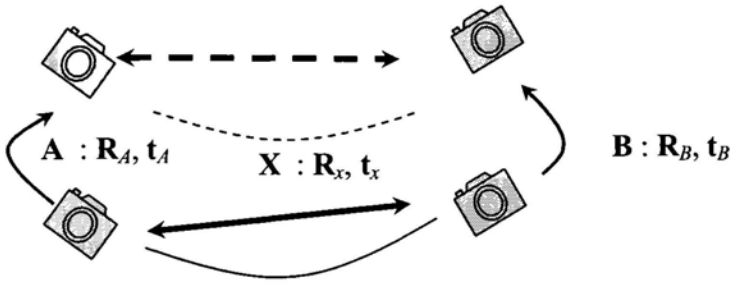


Figure 4.8 A camera pair undergoing rigid motion for the determination of the cameras' relative geometry.

If the cameras' relative geometry is expressed by a 4×4 matrix **X**, and the two respective camera motions by **A** and **B**, because of the rigidity of the overall motion of the camera pair we have the relationship **AX**=**XB**, where $\mathbf{A}=\begin{bmatrix}\mathbf{R}_A & \mathbf{t}_A \\ \mathbf{0} & 1\end{bmatrix}$, $\mathbf{B}=\begin{bmatrix}\mathbf{R}_B & \mathbf{t}_B \\ \mathbf{0} & 1\end{bmatrix}$, and $\mathbf{X}=\begin{bmatrix}\mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0} & 1\end{bmatrix}$. In details, the above equality can be rewritten into:

$$\begin{bmatrix}\mathbf{R}_A & \mathbf{t}_A \\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0} & 1\end{bmatrix}=\begin{bmatrix}\mathbf{R}_x & \mathbf{t}_x \\ \mathbf{0} & 1\end{bmatrix}\begin{bmatrix}\mathbf{R}_B & \mathbf{t}_B \\ \mathbf{0} & 1\end{bmatrix}$$

And it could be decomposed into two expressions:

$$\mathbf{R}_A\mathbf{R}_x = \mathbf{R}_x\mathbf{R}_B \tag{4.5}$$

(or $\boldsymbol{\omega}_A = \mathbf{R}_x\boldsymbol{\omega}_B$ in vector form)

$$(\mathbf{R}_A - \mathbf{I})\mathbf{t}_x = \mathbf{R}_x\mathbf{t}_B - \mathbf{t}_A \tag{4.6}$$

where $\mathbf{R}_x$, $\mathbf{R}_A$, $\mathbf{R}_B$ are the 3×3 orthonormal matrices representing the rotational components of **X**, **A**, **B** respectively, $\mathbf{t}_x$, $\mathbf{t}_A$, $\mathbf{t}_B$ are the 3-vectors representing the translational components, and $\boldsymbol{\omega}_A$, $\boldsymbol{\omega}_B$ are the rotations $\mathbf{R}_A$, $\mathbf{R}_B$ expressed in the axis-angle form.

If the camera motions **A** and **B** can be determined from normal flows by using the orthogonalized co-point vector field mechanism, Equations (4.5) and (4.6) would provide constraints for the determination of the parameters in **X**. Simple the solution scheme might appear, there are however two issues to overcome. We take the planar image space for an example to explain the two issues. First, if the process involves general motion of any particular camera (i.e., motion that involves both translation and rotation), the image space associated with that camera contains not only positively labeled and negatively labeled regions in the orthogonalized co-point vector field mechanism, but also two "Don't know" regions as illustrated by Fig. 4.6(c) (or Fig. 4.7(c) if the orthogonalized co-axis vector field is applied). The presence of the "Don't know" regions would add much challenge to the localization of the zero-boundaries. Second, with only the normal flows not the full flows accessible in the respective image streams of the two cameras, generally only a small fraction of the data points can be made usable in the image space under any random choice of the **s**-axis. This issue is the most troubling, as the localization of the zero-boundary from very sparsely labeled data points would be an almost formidable task. In this work, we specifically address these two issues.

On the first issue, we adopt specific rigid motions of the camera pair to avoid as much as possible the presence of general motion of any particular camera. On the second issue, we propose a scheme that allows the **s**-axes to be chosen not randomly, but according to how many data points they can make useful in the co-point vector field mechanism. The scheme allows each data point (an image position with detectable normal flow) to determine a family of **s**-axes that could make that particular data point useful, and to vote for such axes in the space of all possible **s**-axes. Once the votes from

all data points have been collected, the **s**-axes in the **s**-axis space that have high counts of votes are then should be used in the co-point vector field mechanism.

4.2.1 Determination of $\mathbf{R}_x$

To determine the rotation component $\mathbf{R}_x$ of the inter-camera geometry, we let the camera pair undergo a specific motion – pure translation – so as to reduce the complexity in locating the zero-boundary of the positive-negative labeled pattern in the image space.

When the camera pair exercises a rigid-body pure translation, the motion of each camera is also a pure translation, and $\mathbf{R}_A$=**I**. From Equation (4.6) we have $\mathbf{t}_A = \mathbf{R}_x\mathbf{t}_B$. By normalizing both sides of the above equality, it could be rewritten as:

$$\tilde{\mathbf{t}}_A = \mathbf{R}_x\tilde{\mathbf{t}}_B \tag{4.7}$$

where $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ correspond to the FoEs of the two cameras as unit vectors in homogeneous coordinates.

The following subsections will present how to estimate $\mathbf{R}_x$ from normal flows. We analyze the problem based on both the spherical image space and the planar image space. Since, planar image space is often used in practice, we will only test the developed algorithm with real image data in planar image space. The application of the developed algorithm in the spherical image space will not be explored and will remain as future research opportunities.

4.2.1.1 Estimating $\mathbf{R}_x$ in Spherical Image Space

Suppose the camera pair undergoes a pure translation **t** rigidly. The **s**-co-axis vector field model is adopted to simplify the positive-negative pattern analysis. For an arbitrarily chosen axis $\mathbf{s}_1$, the $\mathbf{s}_1$-co-axis vector field is generated, and then on the spherical surface it is applied to the flow vectors induced by the translation **t**. A great circle will always exist and it is determined by the arbitrarily chosen axis $\mathbf{s}_1$ and the normalized translational vector $\tilde{\mathbf{t}}$. The great circle will divide the spherical surface into two hemispheres, a positive hemisphere and a negative hemisphere. The normalized vectors representing the translations in their own coordinate systems of camera *A* and camera *B* are $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ respectively. Furthermore, we will choose the same **s**-axis for both translations, and the arbitrarily chosen axis for both translations is $\mathbf{s}_1$. Then, there would be two great circles, which are defined by $(\tilde{\mathbf{t}}_A \times \mathbf{s}_1)\cdot\mathbf{p} = 0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_1)\cdot\mathbf{p} = 0$ (where **p** represents the image point on the spherical surface) respectively, locating on the spherical surface. θ_1, the angle between the two planes in which the two great circles lie on, is related to the rotation component $\mathbf{R}_x$ of the stereo cameras and it is illustrated in Fig. 4.9:

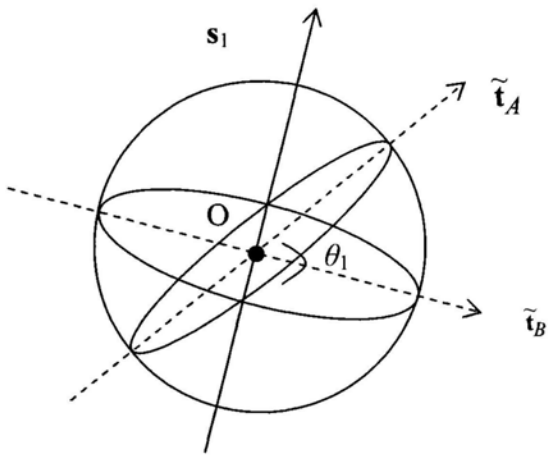


Figure 4.9 Two s_1-co-axis positive-negative patterns are overlapped on one sphere model. O is the optical center of spherical image space, and s_1 is the arbitrarily chosen axis. $\tilde{t}_A$ and $\tilde{t}_B$ are the normalized translational vectors with respect to the coordinate systems of camera A and camera B, respectively. θ_1 is the angle between the planes consisting of the two great circles.

A different choice of the **s**-co-axis vector field would result in a different angle θ. If one more axis, $\mathbf{s}_2$, is applied to the stereo cameras undergoing pure translation in the direction of **t**, two new $\mathbf{s}_2$-co-axis positive-negative patterns will be generated with another two great circles defined by $(\tilde{\mathbf{t}}_A \times \mathbf{s}_2)\cdot\mathbf{p} = 0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_2)\cdot\mathbf{p} = 0$ respectively. Then, we overlap the new positive-negative patterns on the previous $\mathbf{s}_1$-co-axis positive-negative patterns as illustrated by Fig. 4.9. Finally, the four great circles defined by $\mathbf{s}_1$, $\mathbf{s}_2$, $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ will present on the spherical surface as shown in Fig. 4.10 where θ_1 is the angle between the two planes consisting the great circles satisfying $(\tilde{\mathbf{t}}_A \times \mathbf{s}_1)\cdot\mathbf{p} = 0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_1)\cdot\mathbf{p} = 0$, and θ_2 is the angle between the other two planes consisting the great circles satisfying $(\tilde{\mathbf{t}}_A \times \mathbf{s}_2)\cdot\mathbf{p} = 0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_2)\cdot\mathbf{p} = 0$..

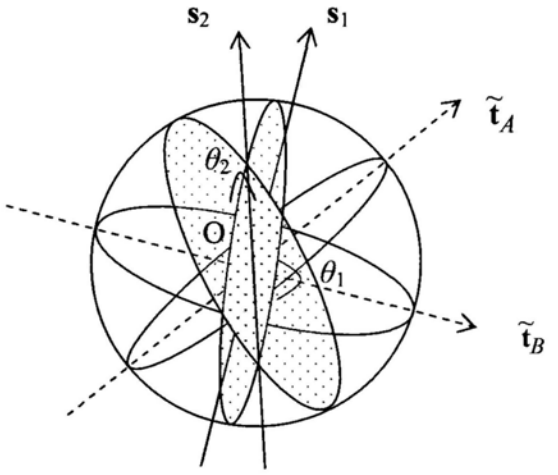


Figure 4.10 Overlap four s-co-axis positive-negative patterns in the same spherical image space. θ_1 is the angle between the two planes consisting of great circles $(\tilde{\mathbf{t}}_A \times \mathbf{s}_1)\cdot\mathbf{p}=0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_1)\cdot\mathbf{p}=0$, and θ_2 is the angle between the other two planes defined by the great circles $(\tilde{\mathbf{t}}_A \times \mathbf{s}_2)\cdot\mathbf{p}=0$ and $(\tilde{\mathbf{t}}_B \times \mathbf{s}_2)\cdot\mathbf{p}=0$.

Angle θ_1 and θ_2 depend on the selection of the **s**-axes; however, they are also related to the rotation of the vectors $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$. More precisely, we aim to investigate how to determine the rotation component of stereo cameras by using angle θ_i and the normalized translation vectors $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$.

Once axes $\mathbf{s}_1$ and $\mathbf{s}_2$ are selected, θ_1 and θ_2 are fixed. However, the angle between $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ is not fixed since the two planes $\tilde{\mathbf{t}}_A$ O $\mathbf{s}_1$ and $\mathbf{s}_1$O $\tilde{\mathbf{t}}_B$, which share the intersection $\mathbf{s}_1$ are able to rotate about $\mathbf{s}_1$ arbitrarily while keeping θ_1 constant. Similarly, the other two planes $\tilde{\mathbf{t}}_A$O$\mathbf{s}_2$ and $\mathbf{s}_2$O$\tilde{\mathbf{t}}_B$ are also able to rotate about $\mathbf{s}_2$ randomly. In other words, the angle between $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ can not be determined with the aid of angles θ_1 and θ_2 unless both $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ are known.

However, if one more axis, $\mathbf{s}_3$, is applied, the angle between the $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ is fixed and $\mathbf{R}_x$ can be estimated with the aid of the angle θ_i (i=1, 2, 3) with the condition that only one of the two normalized translation vectors $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ is known. Moreover, two independent translations of the stereo rig are necessary to obtain a unique $\mathbf{R}_x$. The mathematical expression and proof are shown in the following.

The angle θ_i (i=1, 2, 3), is defined as follow:

$$\begin{cases} \cos\theta_1 = (\tilde{\mathbf{t}}_A \times \mathbf{s}_1)\cdot(\tilde{\mathbf{t}}_B \times \mathbf{s}_1) \\ \cos\theta_2 = (\tilde{\mathbf{t}}_A \times \mathbf{s}_2)\cdot(\tilde{\mathbf{t}}_B \times \mathbf{s}_2) \\ \cos\theta_3 = (\tilde{\mathbf{t}}_A \times \mathbf{s}_3)\cdot(\tilde{\mathbf{t}}_B \times \mathbf{s}_3) \end{cases} \tag{4.8}$$

Equation (4.8) can be rewritten as:

$$\begin{cases} \cos\theta_1 = \tilde{\mathbf{t}}_B{}^{\mathrm{T}}\mathbf{R}_x{}^{\mathrm{T}}\mathbf{A}_1\tilde{\mathbf{t}}_B \\ \cos\theta_2 = \tilde{\mathbf{t}}_B{}^{\mathrm{T}}\mathbf{R}_x{}^{\mathrm{T}}\mathbf{A}_2\tilde{\mathbf{t}}_B \\ \cos\theta_3 = \tilde{\mathbf{t}}_B{}^{\mathrm{T}}\mathbf{R}_x{}^{\mathrm{T}}\mathbf{A}_3\tilde{\mathbf{t}}_B \end{cases} \tag{4.9}$$

where $\mathbf{A}_1 = [\mathbf{s}_1]_\times{}^{\mathrm{T}}[\mathbf{s}_1]$, $\mathbf{A}_2 = [\mathbf{s}_2]_\times{}^{\mathrm{T}}[\mathbf{s}_2]$, $\mathbf{A}_3 = [\mathbf{s}_3]_\times{}^{\mathrm{T}}[\mathbf{s}_3]$ with $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ being the three arbitrarily chosen and normalized **s**-axes in generating **s**-co-axis vector fields. Also, $\mathbf{R}_x$ is the rotation component of the binocular geometry. It can be seen that θ_1, θ_2 and θ_3 could be estimated by analyzing the **s**-co-axis positive-negative patterns defined by **s**-axes and the normal flows on the spherical surface.

Equation (4.9) can be rewritten as follows:

$$\underbrace{\begin{bmatrix} \cos\theta_1 \\ \cos\theta_2 \\ \cos\theta_3 \end{bmatrix}}_{3\times1} = \underbrace{\begin{bmatrix} \mathbf{B}_1^1 & \mathbf{B}_1^2 & \mathbf{B}_1^3 \\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \mathbf{B}_2^3 \\ \mathbf{B}_3^1 & \mathbf{B}_3^2 & \mathbf{B}_3^3 \end{bmatrix}}_{3\times9} \left[\underbrace{R_{11} \quad R_{12} \quad R_{13} \quad R_{21} \quad R_{22} \quad R_{23} \quad R_{31} \quad R_{32} \quad R_{33}}_{1\times9} \right]^{\mathrm{T}} \tag{4.10}$$

where $\mathbf{R}_x = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$, and $\mathbf{B}^i_j$, which is a 1×3 matrix representing the term calculated by $\mathbf{A}_j$ and $\tilde{\mathbf{t}}_B$. The index j stands for different **s**-axis.

It can be proved that $\text{rank}\left(\begin{bmatrix} \mathbf{B}^i_1 \\ \mathbf{B}^i_2 \\ \mathbf{B}^i_3 \end{bmatrix}\right) = 1$, where i=1, 2 ,3 and $\text{rank}\left(\begin{bmatrix} \mathbf{B}^p_1 & \mathbf{B}^q_1 \\ \mathbf{B}^p_2 & \mathbf{B}^q_2 \\ \mathbf{B}^p_3 & \mathbf{B}^q_3 \end{bmatrix}\right) = 2$,

where p, q=1, 2, 3 (and $p \neq q$), and $\text{rank}\left(\begin{bmatrix} \mathbf{B}^1_1 & \mathbf{B}^2_1 & \mathbf{B}^3_1 \\ \mathbf{B}^1_2 & \mathbf{B}^2_2 & \mathbf{B}^3_2 \\ \mathbf{B}^1_3 & \mathbf{B}^2_3 & \mathbf{B}^3_3 \end{bmatrix}\right) = 3$. Details of the proof are provided in the appendix.

By manipulating Equation (4.10), it can be arranged into:

$$\underbrace{\begin{bmatrix} f(\theta_1) \\ f(\theta_2) \\ f(\theta_3) \end{bmatrix}}_{3\times 1} = \underbrace{\begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 \end{bmatrix}}_{3\times 9} \begin{bmatrix} \underbrace{R_{11} \quad R_{12} \quad R_{13} \quad R_{21} \quad R_{22} \quad R_{23} \quad R_{31} \quad R_{32} \quad R_{33}}_{1\times 9} \end{bmatrix}^{\mathrm{T}} \tag{4.11}$$

where $f(\theta_i)$ is a scalar and $\mathbf{C}_i$ is a 1×3 matrix with $\text{rank}\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix}\right) = 1$.

Equation (4.11) is then further rearranged into the following form:

$$\begin{bmatrix} g(\theta_1) \\ g(\theta_2) \\ g(\theta_3) \end{bmatrix} = \mathbf{R}_x \begin{bmatrix} h(\mathbf{s}_1, \tilde{\mathbf{t}}_B) \\ h(\mathbf{s}_2, \tilde{\mathbf{t}}_B) \\ h(\mathbf{s}_3, \tilde{\mathbf{t}}_B) \end{bmatrix} \tag{4.12}$$

subject to $\mathbf{R}_x{}^{\mathrm{T}}\mathbf{R}_x = \mathbf{I}$, where $g(\theta_i)$ and $h(\mathbf{s}_i, \tilde{\mathbf{t}}_B)$, ($i$=1, 2, 3) are both scalars.

Unique $\mathbf{R}_x$ cannot be obtained from Equation (4.12) if only one unit translational vector $\tilde{\mathbf{t}}_B$ is available by locating zero-boundaries of the positive-negative patterns on the spherical surface. At least two translational motions of the stereo cameras in different directions are required to obtain from Equation (4.12) a unique solution of $\mathbf{R}_x$ [Kanatani, 1993]. Suppose the stereo rig translates towards different directions twice, then the solution for $\mathbf{R}_x$ will be:

$$\mathbf{R}_x = \mathbf{U}\mathbf{V}^{\mathrm{T}} \tag{4.13}$$

One problem that may arise here in using Equation (4.13) is that rotational matrix $\mathbf{R}_x$ obtained from Equation (4.13) can have a determinant of -1. Aiming at solving the above problem, Equation (4.13) can be modified as follows:

$$\mathbf{R}_x = \mathbf{U}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{U}\mathbf{V}^{\mathrm{T}}) \end{bmatrix}\mathbf{V}^{\mathrm{T}} \tag{4.14}$$

where $\mathbf{U}$ and $\mathbf{V}$ are the matrixes of eigenvectors obtained by SVD (singular value decomposition): $\mathbf{D} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$. More specifically, the matrixes $\mathbf{U}$ and $\mathbf{V}$ are the decomposed components of :

$$D = \frac{1}{N}\sum_{i=1}^{N}\begin{bmatrix} g_i(\theta_1) \\ g_i(\theta_2) \\ g_i(\theta_3) \end{bmatrix}\begin{bmatrix} h_i(\mathbf{s}_1, \tilde{\mathbf{t}}_B) & h_i(\mathbf{s}_2, \tilde{\mathbf{t}}_B) & h_i(\mathbf{s}_3, \tilde{\mathbf{t}}_B) \end{bmatrix} \tag{4.15}$$

where the index i represents the i^{th} translation of the stereo rig and min (N)=2.

By far the unique solution for the rotation component $\mathbf{R}_x$ of the binocular geometry is obtained, via the knowledge of the translational direction of camera B. Angle θ_i (i=1, 2, 3) (angle between the different loci on the spherical surface.) can be estimated by analyzing the **s**-co-axis positive-negative patterns.

4.2.1.2 Estimating $\mathbf{R}_x$ in Planar Image Space

As Equation (4.7) provides no more than two scalar constraints for $\mathbf{R}_x$ which contains three degrees of freedom, at least two translational motions in different directions are required to achieve from Equation (4.7) a unique solution of $\mathbf{R}_x$ [Kanatani, 1993]. In general, we can use more than two rigid translations of the camera system. With N ($N \geq 2$) rigid-body translations of the camera pair, the matrix $\mathbf{K}_t = \sum_{1}^{N} \tilde{\mathbf{t}}_A \tilde{\mathbf{t}}_B^T$ can be computed, and the least-square-error solution for $\mathbf{R}_x$ is what Equation (4.14) describes, where $\mathbf{U}_t$, $\mathbf{V}_t$ are results of SVD (singular value decomposition) of $\mathbf{K}_t$ as $\mathbf{K}_t = \mathbf{U}_t \mathbf{S}_t \mathbf{V}_t$.

On determining $\tilde{\mathbf{t}}_A$ and $\tilde{\mathbf{t}}_B$ in each rigid-body translation of the camera pair, we adopt the orthogonalized co-point vector field model (with respect to a chosen **s**-axis) to generate patterns from the normal flows in the respective image streams. Both cameras will exhibit patterns like the one shown in Fig.4.6 (b), which contains only the positive and the negative regions separated by a straight line (the zero boundary) with no "Don't know" region. Hence, the analysis of the positive-negative pattern is not difficult. In a word, the reason that the pure translation of the stereo rig is employed is to simplify the positive-negative pattern division problem.

The alternative orthogonalized co-axis vector field model (with respect to the same **s**-axis) can also adopted to generate the positive-negative pattern. If orthogonalized co-axis vector field is applied, the patterns will have positive region and negative region separated by a 2nd order curve shown in Fig.4.7 (a), still without the "Don't know" region.

4.2.2 Determination of $\mathbf{t}_x$ up to Arbitrary Scale

Here we describe how a solution scheme similar to that for determining the relative orientation can also be used to determine the separation $\mathbf{t}_x$ of the cameras. However, we must point out that the required rigid motions of the camera pair in this case are not as easy to conduct precisely as that in the case of determining the relative orientation, and thus the scheme has certain limitation on its accuracy. It should also be emphasized again that, unless with certain metric measurement about the imaged scene, due to the aperture problem $\mathbf{t}_x$ can only be determined up to arbitrary scale from visual motion data alone.

To determine the baseline $\mathbf{t}_x$ of the camera pair, we let the camera pair undergo rigid-body pure rotations while the cameras capture the image stream data. In particular, suppose the camera pair together have a pure rotation about an arbitrary axis passing through the optical center of one camera say camera *A*. Then on the motion of each camera, camera *A* only undergoes a pure rotation, while camera *B*'s motion consists of a rotation about an axis containing the optical center of camera *A*, and a translation orthogonal to the baseline link between the cameras. In this case Equation (4.6) can be rewritten as:

$$(\mathbf{R}_A - \mathbf{I})\mathbf{t}_x = \mathbf{R}_x \mathbf{t}_B \tag{4.16}$$

where Rank ($\mathbf{R}_A$- $\mathbf{I}$) = 2. We then rewrite Equation (4.16) to a homogeneous system:

$$\hat{\mathbf{A}}\tilde{\mathbf{t}}_x = \mathbf{0} \tag{4.17}$$

where $\tilde{\mathbf{t}}_x$ is the normalized vector representing the direction of the baseline, and $\hat{\mathbf{A}}$ is a 2×3 matrix calculated from $\mathbf{R}_x$, $\mathbf{R}_A$, $\tilde{\mathbf{t}}_B$ as

$$\hat{\mathbf{A}} = \begin{bmatrix} \mathbf{M}_1 N_3 - \mathbf{M}_3 N_1 \\ \mathbf{M}_2 N_3 - \mathbf{M}_3 N_1 \end{bmatrix} \quad \text{with} \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \\ \mathbf{M}_3 \end{bmatrix} = \mathbf{R}_A - \mathbf{I} \quad \text{and} \quad \begin{bmatrix} N_1 \\ N_2 \\ N_3 \end{bmatrix} = \mathbf{R}_x \mathbf{t}_B .$$

Notice that Rank ($\hat{\mathbf{A}}$) = 1. At least two rotations are needed to determine $\tilde{\mathbf{t}}_x$ uniquely. With at least two rigid-body rotations of the camera pair, like in the determination of $\mathbf{R}_x$, we apply SVD to the homogeneous linear system of equations expressed by Equation (4.17) to determine the least-square-error solution of $\mathbf{t}_x$ (up to arbitrary scale).

If the orthogonalized co-point vector field mechanism is adopted, camera A, which has only pure rotations in the process, has the positive-negative labeled patterns in the image space just like the one shown in Fig.4.6 (a), in which a 2nd order curve (the zero-boundary) separates the '+' and '-' labeled regions. We use the axis-angle expression $\boldsymbol{\omega}_A=[\alpha_A, \beta_A, \gamma_A]^T$ to represent the rotation of camera A. With first the ratios α_A /γ_A and β_A /γ_A determined from the 2nd order curve, the third component γ_A (which together with α_A/γ_A and β_A /γ_A define the magnitude of rotation) can then be determined using the method named "detranslation" as presented in [Fermüller, 1995B] [Fermüller, 1998]. As for camera B, the positive-negative labeled patterns in the image space take the form of Fig.4.6 (c), and pose more challenge because of the existence of two "Don't know" regions. There are two zero-boundaries to be determined: one a 2nd order curve, and the other a straight line. Fortunately, the rotational component $\boldsymbol{\omega}_B$ of camera B can be computed directly from Equation (4.5) once $\mathbf{R}_x$ has been determined from the previous step. With knowledge of $\boldsymbol{\omega}_B$, by applying the orthogonalized co-point vector field to the image space, the 2nd order zero-boundary dividing the positive-negative labeled patterns can be pinpointed. With this, the other straight zero-boundary defined by the FoE can be easily located despite the presence of the two "Don't know" regions.

The direction $\mathbf{t}_B$ of the translational component can thus be determined from the straight boundary. Finally the inter-camera separation $\mathbf{t}_x$ (up to arbitrary scale) can be calculated from Equation (4.17) once $\mathbf{R}_A$ and $\widetilde{\mathbf{t}}_B$ are both made available from the positive-negative patterns. In a word, the reason that the pure rotation about the optical center of one camera but not a general motion is employed, is to simplify the positive-negative pattern division problem.

A similar strategy can be used if the orthogonalized co-axis vector field is adopted to determine the direction of the baseline.

The shortcoming of the above solution scheme is that rigid rotation about the optical center of a camera cannot be conducted precisely unless the optical center of the camera is well positioned in space. For this reason we view the above solution scheme of determining $\mathbf{t}_x$, which is parallel to the solution scheme of determining $\mathbf{R}_x$, as one for obtaining only approximate knowledge about the inter-camera separation. We must point out however that the shortcoming is not present in the solution scheme of determining $\mathbf{R}_x$.

4.2.3 Optimal Selection of the **s***-Axis Set*

For any single camera, a different choice of the **s**-axis would allow a different subset of the data points (i.e., image positions with normal flow observable) to be usable in generating the positive-negative labeled pattern in the image space. Obviously a higher density of the labeled patterns is desired, as it would make the localization of the zero-boundary easier. Here we propose a scheme of selecting a set of **s**-axes that lead to higher densities of usable data points.

In the following discussion, for simplicity we only describe the scheme under the case that the camera motion is a pure translation. The scheme for the pure camera rotation is the same when the orthogonalized **s**-co-axis vector field is applied to it.

4.2.3.1 From Normal Flow Data Point to a Locus of **s**-axis

Given any choice of the **s**-axis, an orthogonalized co-point vector field is defined for the image space, and only the data points with the normal flows exactly parallel or anti-parallel with the orthogonalized co-point field vectors there are usable in the positive-negative pattern analysis process. Below we re-visit the mechanism from the opposite angle, more precisely from the angle of data point not **s**-axis. A data point (x_i, y_i) with normal flow (u^i_n, v^i_n) is usable only under the adoption of the following family of **s**-axes: **s**-axis whose equivalent position $\mathbf{S}=(s_x, s_y)$ in the image plane is located on the line l_i that contains the data point (x_i, y_i) and that is orthogonal to the normal flow (u^i_n, v^i_n). This is illustrated by Fig. 4.11. We call the line l_i the **S**-line of the data point (x_i, y_i), and it can be expressed as:

$$u^i{}_n s_x + v^i{}_n s_y - (u^i{}_n x_i + v^i{}_n y_i) = 0 \tag{4.18}$$

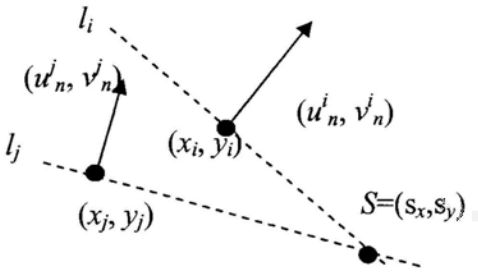


Figure 4. 11 The S-lines of data points (x_i, y_i) (with normal flow (u^i_n, v^i_n)) and (x_j, y_j) (with normal flow (u^j_n, v^j_n)).

Thus, to find the **s**-axis which can make the maximum number of data points useful, a simple scheme is to let each data point vote for the members of its **S**-line in the space of all possible **s**-axes (which is only a two-dimensional space, as an **s**-axis has only two degrees of freedom). The **s**-axes of high count of votes are then the good choices that should be used in the orthogonalized co-point vector field mechanism.

4.2.3.2 Optimum Determination

It seems that we could obtain a linear system of equations for the optimal **s**-axis (point **S** in the image space) from say n data points using Equation (4.18), and solve for the optimal **s**-axis. However, the optimal **s**-axis can not be calculated as the least-square solution of Equation (4.18). Because no matter how "good" the **s**-axis is, the normal flows that could vote for this s-axis are still a very small portion of all the detectable normal flows in the image domain, for instance 2~3 % in average. Hence estimating the solution of Equation (4.18) must be failed by the idea of eliminating the outliers.

We thus adopt a voting scheme similar to the Hough Transform. We use an accumulator array to represent the entire space of all possible **s**-axes, and to collect votes from each data point. The accumulator is a two-dimensional array whose axes correspond to the quantized values of s_x and s_y. For each data point (an image point with detectable normal flow), we determine its **S**-line, look for bins in the accumulator array that the line falls into, and put one vote in each of those bins. After we finish this with all the data points, we identify the bin with the highest count of votes in the accumulator array. An example of an accumulator array is shown in Fig. 4.12.

Figure 4. 12 Two-dimensional accumulation array that corresponds to various values of s_x and s_y. The S-line associated with each data point is determined, the array bins corresponding to the line are identified, and each of such bins has the vote count increased by one. The bin with the highest vote count is identified (and marked as a red circle in this figure), which corresponds to the optimal s-axis.

To increase computational efficiency we use a coarse-to-fine strategy in the voting process, as illustrated by Fig. 4.13.

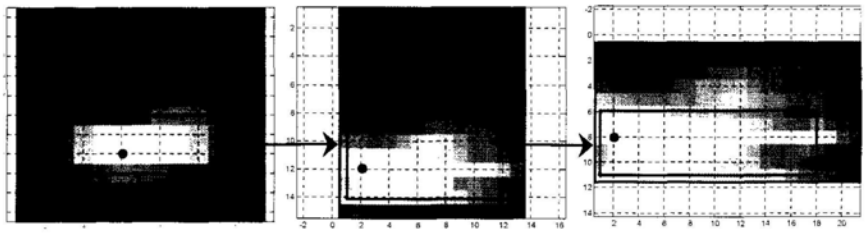


Figure 4. 13 The development of the voting process under the coarse-to-fine strategy.

Since the orthogonalized co-point vector field-based mechanism demands the use of not one but at least a few **s**-axes, in our case we use not only the optimal **s**-axis but some **s**-axes of the highest vote counts. While in synthetic data experiments the scene texture (and thus the orientation of the normal flow) is random and thus all **s**-axes have similar density of usable data points, in real image data the scene texture is generally concentrated around a few directions (and so is the normal flow), and the densities of the usable data points could be drastically different under a different choice of the **s**-axis set.

Experimental result shows that, in cases of real image data, the adoption of the optimal set of **s**-axes almost always makes great improvement to the solution quality over those under random choices. More specifically, our experiments on real image data show that the pattern generated by the optimal **s**-axis set often has 60% more data points than those generated by the average **s**-axis set.

A brief summary of the entire solution procedure on optimal **s**-axes selection is illustrated by the following flowchart.

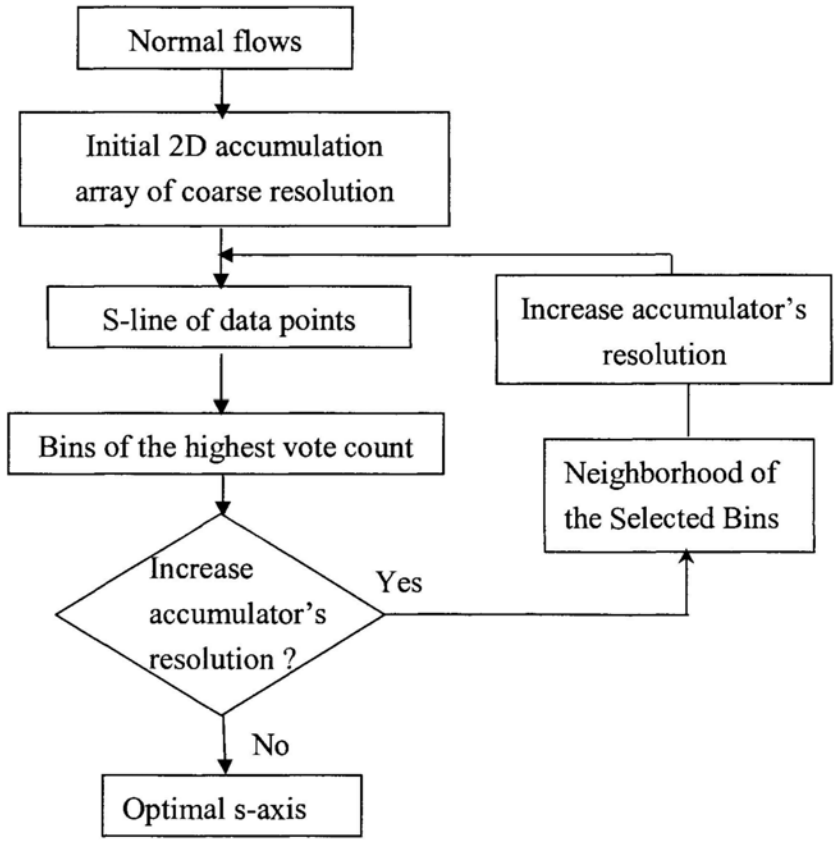


Figure 4. 14 Flow chart to illustrate the process of selecting the optimal s-axis

4.2.4 Entire Solution Procedure on Binocular Geometry Estimation

On estimating binocular geometry, we first let the stereo rig take pure translations to estimate $\mathbf{R}_x$, and then let the cameras rotate about the optical center of one camera to estimate the baseline $\mathbf{t}_x$ up to scale. In details, the steps could be summarized as follows:

Step 1. Let the stereo rig translate as a whole twice, and the two translational motions should be in different directions.

Step 2. Calculate the normal flows from the image sequences.

Step 3. Choose orthogonalized co-point vector field and apply the scheme of optimal selection of the s-axis set.

Step 4. Apply the optimal s-axes to estimate the FoEs of the two cameras

Step 5. Use Equation 4.7 to calculate $\mathbf{R}_x$.

Step 6. Let the stereo rig rotate about the optical center of one camera (for example Camera A) twice, and the two rotational axes should point to different directions.

Step 7. Calculate the normal flows from the image sequences.

Step 8. Choose orthogonalized co-axis vector field and apply the scheme of optimal selection of the s-axes set to camera A.

Step 9. Apply the optimal s-axes to estimate the AoR of camera A. And then calculate $\boldsymbol{\omega}_A$ using the method of "detranslation" [Fermüller, 1995B].

Step 10. Choose co-axis vector field to the normal flows of camera B, and estimate FoE of camera B, with the help of $\mathbf{R}_x$ and $\boldsymbol{\omega}_A$.

Step 11. Use Equation 4.17 to calculate $\mathbf{t}_x$

4.3 Experimental Results

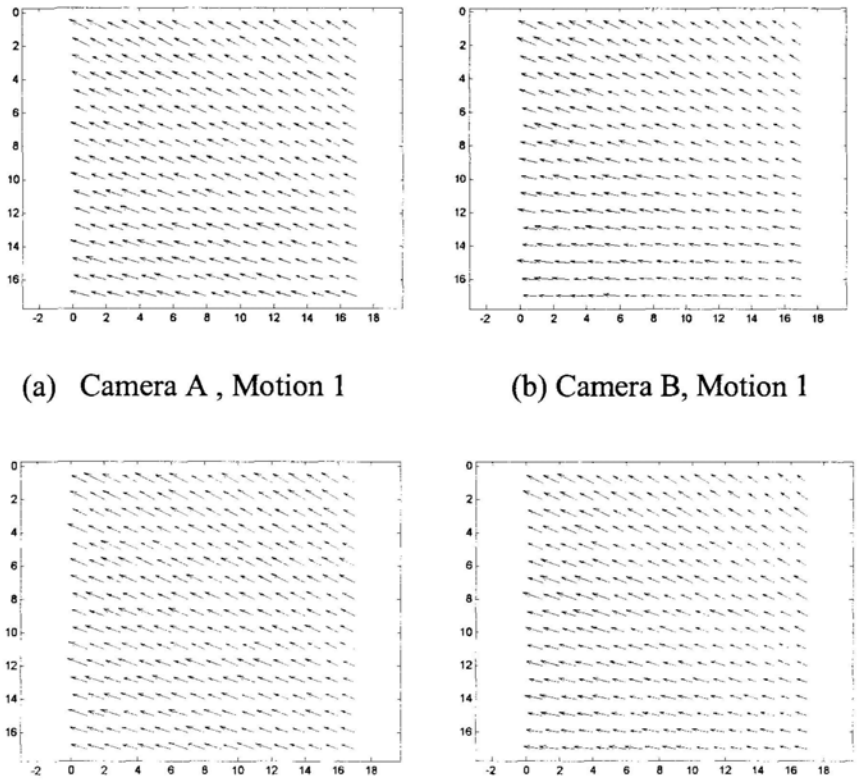
In this section, we present experimental results to illustrate how the described method performs on synthetic and real image data.

4.3.1 Synthetic Data Experiments

Synthetic data experiments are particularly useful for examining solution accuracy because they come with exact ground truth for reference. To synthesize the experimentation data, we first kept the relative geometry of two cameras locked, and then instructed the camera pair undergo the necessary rigid motion. Each of such rigid motions generated the full optical flow on each camera's image plane. We then at each image position of the image plane projected the full flow to an intensity gradient direction there that was generated randomly. In the solution process we then assumed that we did not have access to the full flow fields but only the normal flows. We used image resolution 101×101 pixels in all synthetic data. Moreover, the strategy of optimal selection of the **s**-axis set was not applied when dealing with the synthetic data, as the gradient directions were assigned to image points arbitrarily. Hence, the strategy of optimal selection of the **s**-axis set will not improve the efficiency of the algorithm obviously.

4.3.1.1 Determination of $\mathbf{R}_x$

To determine the rotation component $\mathbf{R}_x$ of the inter-camera geometry, we first conducted two rigid pure translations of the camera pair. The resultant full flow fields in the two cameras' image planes under the two pure translational rigid motions are shown in Fig. 4.15.

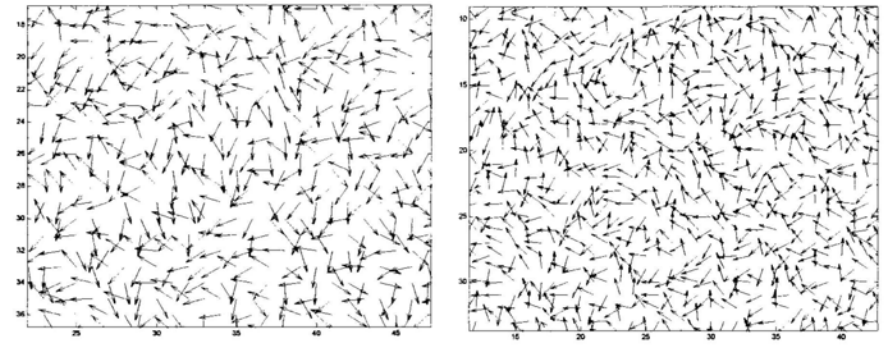


(a) Camera A , Motion 1 (b) Camera B, Motion 1

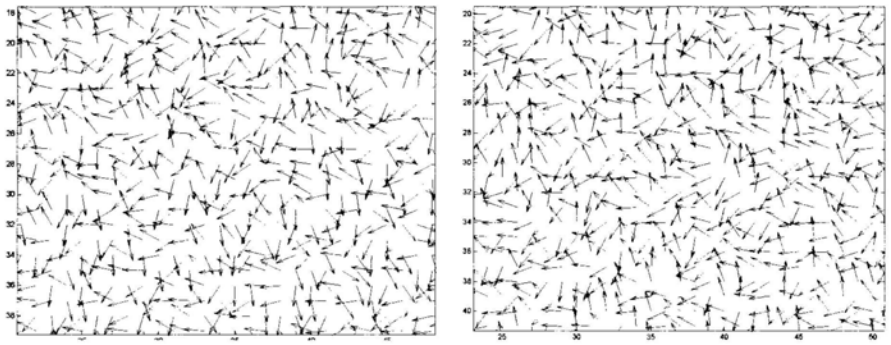
(c) Camera A, Motion 2 (d) Camera B, Motion 2.

Figure 4. 15 Full flow fields in the two cameras' image planes under two pure rigid translations of the camera pair.

We then assigned each image point an arbitrary gradient direction to project the full flow along this direction and obtained the normal flow at this image position. The resultant normal flow fields in the two cameras' image planes under the two pure translational rigid motions are shown in Fig. 4.16.



(a) Camera A, Motion 1 (b) Camera B, Motion 1



(a) Camera A, Motion 2 (b) Camera B, Motion 2

Figure 4. 16 Normal flows in the two cameras' image planes under two pure rigid translations of the camera pair.

Given the first arbitrary **s**-axis, for instance **s**=[1 0 0], we obtained the first **s**-co-axis pattern as shown in Fig.4.18. Initially, we arbitrarily chose the FoE anywhere within the image frame. After investigating the pseudo FoEs 0.25 by 0.25 pixel, as many as 1000 curves determined by these pseudo FoEs could divide the pattern into two regions well. Then, we applied a second **s**-axis to examine whether these 1,000 pseudo FoEs that performed well in the first pattern would still perform well in the new pattern. We discarded wrong FoEs that had bad performance when the second s-axis was applied and kept others to the next round of new s-axis until all the possible FoEs became relatively

small. The number of possible FoEs dramatically decreased when more **s**-axes were applied. Finally, the center of all possible FoEs' was considered as the input to compute $\mathbf{R}_x$. Table 4.1 shows the FoEs estimated by locating the zero-boundaries.

Table 4. 1 Estimation of FoEs. CA: Camera *A*; CB: Camera *B*; M1: Motion 1; M2: Motion 2

		Ground Truth	Experiment
FOE	CA,M1	[29.534 12.465]	[30.000 12.950]
	CB,M1	[-5.000 30.000]	[-5.000 29.775]
	CA,M2	[30.972 9.198]	[32.000 9.475]
	CB,M2	[-3.000 27.000]	[-3.000 27.375]

As mentioned, the number of possible FoEs dramatically decreased when more **s**-axes were applied to the normal flows. In the beginning, we applied the first **s**-axis **s**=[1 0 0] to the normal flows, generated the first positive-negative pattern, and investigated the pseudo FoEs 0.25 by 0.25 pixel within the possible location of FoE. (It is reasonable to roughly know the parameters of the camera's translation by reading the data from the platform controller in practice.) There were a total of 1,353 2^{nd} order curves defined by these 1,353 corresponding possible FoEs for camera *A* during motion one, that could divide the positive-negative pattern into two regions very well. Then we applied the second s-axis **s**=[0 1 0] to generate a new pattern and examined whether the previous pseudo 1,353 FOEs that had good performance in the first pattern would still perform well in the new pattern. Finally, 82 of the 1,353 ones were proved to still have to good performance. We discarded wrong FoEs that had bad performance when the second s-axis was applied and kept others to the next round of new s-axis until all the possible FoEs became relatively small. The FoE could be located precisely if sufficient **s**-axes were used. Fig. 4.17 shows the FoEs shrank to quite a small area (within one pixel) after a number of **s**-axe were applied. For example, 28 **s**-axes are enough to obtain

a precise location of FoE for camera B during motion 2, while 82 s-axes are enough to obtain a precise location of FoE for camera A during motion 1.

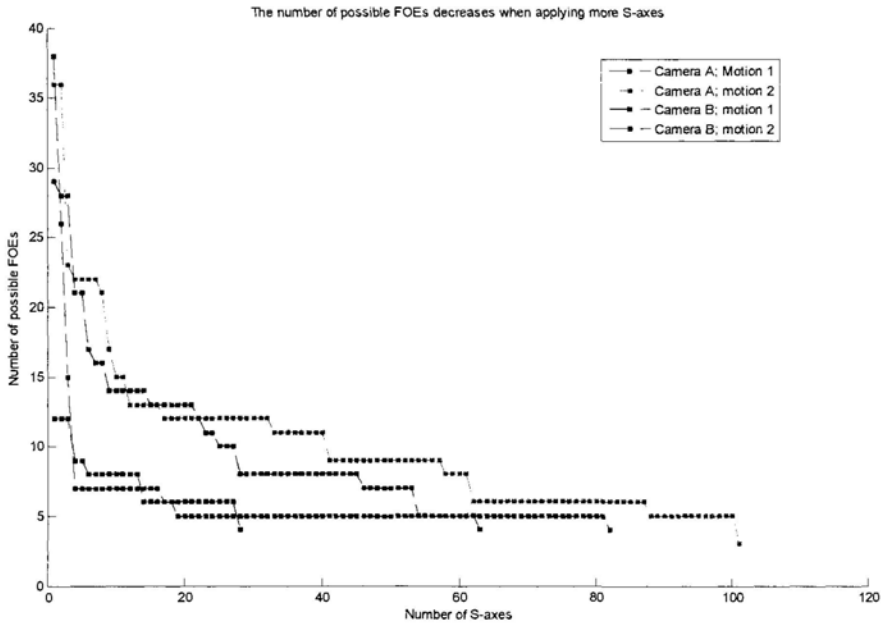
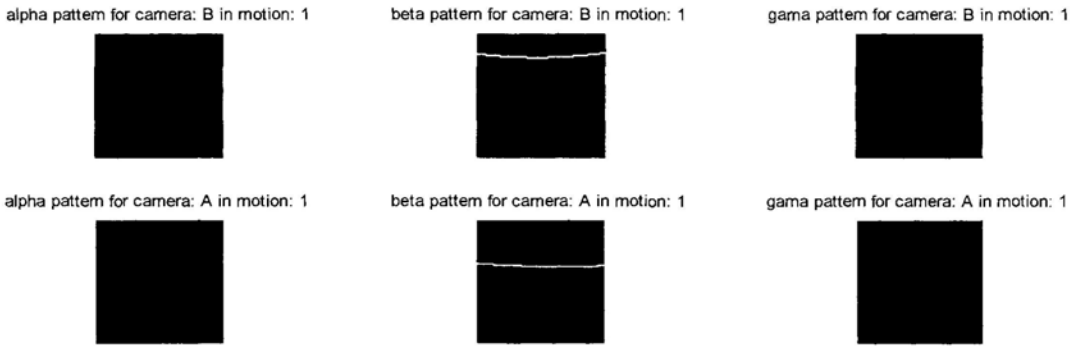


Figure 4. 17 The number of candidate FoEs decreased dramatically with the number of s-axes that were used.

Fig. 4.18 shows the zero-boundaries determined by the estimated FoEs which divided the image domain into two regions according to the positive labels and negative labels.



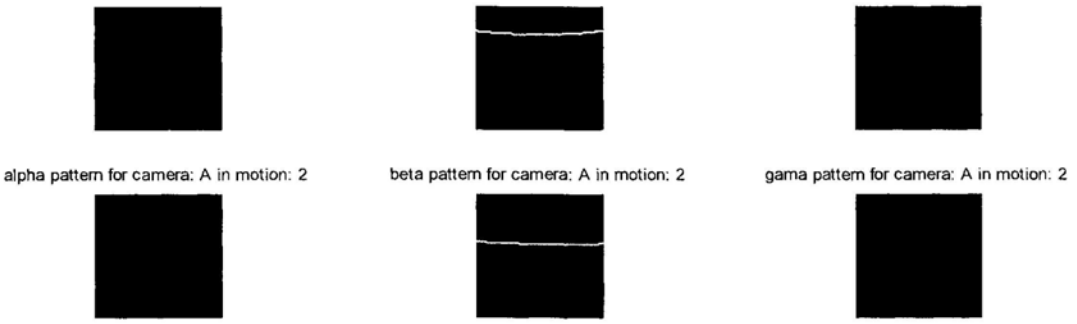


Figure 4. 18 Three patterns: alpha pattern (s=[1 0 0]) , beta pattern (s=[0 1 0]), gamma pattern (s=[0 0 1]). Green dots represent negative position; red dots represent positive position.

Finally, the rotation component ω_x of the stereo geometry was estimated from the FoEs by SVD. The estimated rotation component ω_x of the experimental binocular geometry is shown in Table 4.2. The error is 0.80° in direction and 1.26% in length.

Table 4. 2 Estimation of the rotational component of the binocular geometry.

	ω_x
Ground Truth	$[0.1000 \quad 0.2000 \quad -0.2000]^T$
Experiment	$[0.0974 \quad 0.2041 \quad -0.2029]^T$

In this synthetic data experiment, we added 45dB white Gaussian noise to the normal flow vectors to see how sensitive the result is to noise. The example of the normal flows vectors added Gaussian noise is shown in Fig. 4.19.

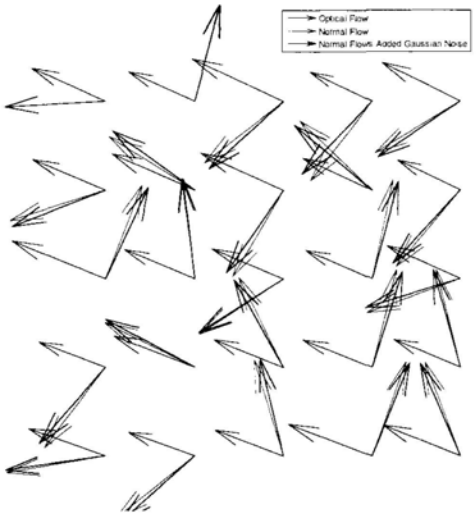
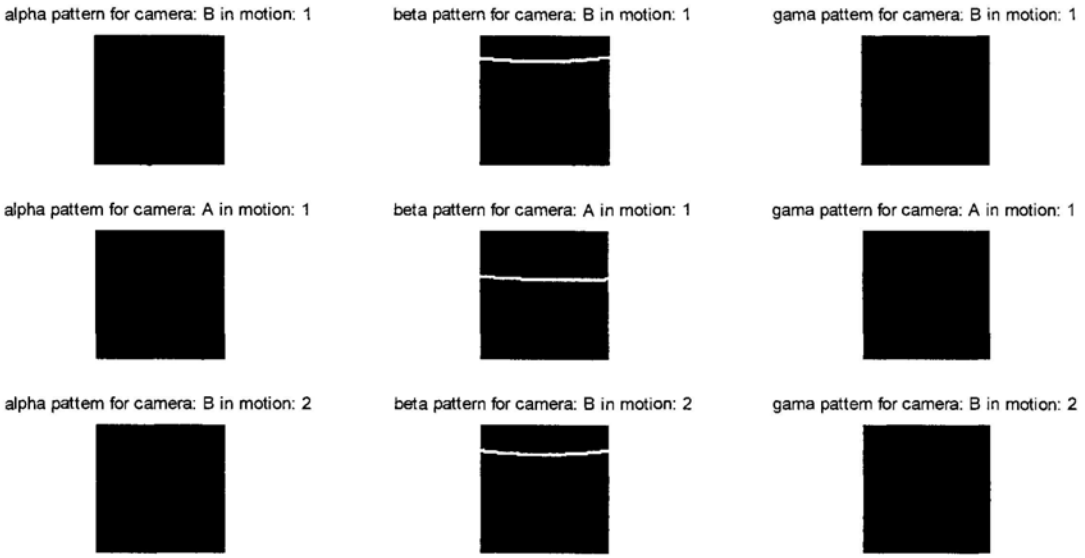


Figure 4.19 Normal flow vectors added 45dB white Gaussian noise. Vectors in blue are the optical flows; vectors in green are the original normal flows; and vectors in red are the nromal flows added Gaussian noise.

Fig. 4.20 shows the zero-boundaries determined by the estimated FoEs which divided the image domain into two regions according to the positive labels and negative labels.. We compared the result to the one that was obtained by using the noise-free normal flows, and found they are very close.



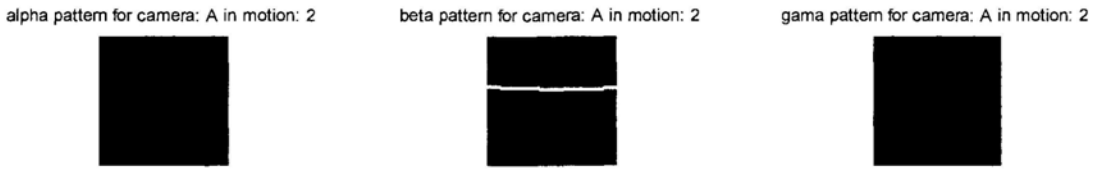


Figure 4. 20 Three patterns by using normal flow vectors added Gaussian noise: alpha pattern (s=[1 0 0]) , beta pattern (s=[0 1 0]), gamma pattern (s=[0 0 1]). Green dots represent negative position; red dots represent positive position.

We applied **s**-axes to the normal flow vectors of the two cameras during the two motions to estimate the FoEs, and set the threshold to let the estimation shrink to the same small area (within one pixel) after a number of **s**-axes were applied. We compared the number of **s**-axes totally applied in the two experiments, as shown in Tab.4.3. Obviously, more **s**-axes are needed to achieve the estimation of the same accuracy, if the normal flow vectors are disturbed by noise.

Table 4. 3 The number of s-axes applied to estimate FoEs. CA: Camera *A*; CB: Camera *B*; M1: Motion 1; M2: Motion 2

	The number of **s**-axes (Noise-free)	The number of **s**-axes (With Gaussian noise)
CA, M1	82	112
CB, M1	63	128
CA, M2	101	203
CB, M2	28	616

The rotation component ω_x of the stereo geometry was estimated from the FoEs by SVD. The estimated rotation component ω_x of the experimental binocular geometry is shown in Table 4.4.

Table 4. 4 Comparison of the estimations of the rotational component of the binocular geometry by using noise-free data and the data disturbed by Gaussian noise

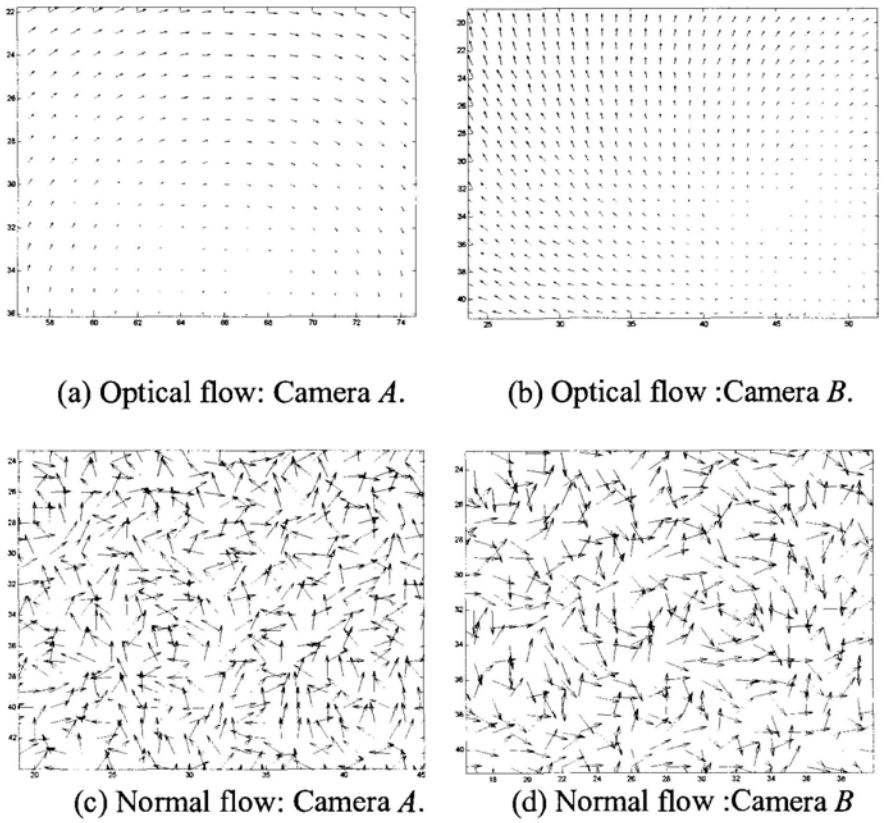
	ω_x
Noise-free	$[0.0974 \quad 0.2041 \quad -0.2029]^T$
With Gaussian noise	$[0.0890 \quad 0.1955 \quad -0.1874]^T$

Experimental results show that the algorithm is stable when dealing with the input data with limited noise. However, much more number of s-axes have to be used to help eliminate the disturbance caused by the small portion of wrong data.

4.3.1.2 Determination of $\mathbf{t}_x$ up to Scale

After $\mathbf{R}_x$ (i.e., ω_x) was determined, we permitted the camera pair to rotate about the optical center of camera *A* in two different velocities and we observed the resulted normal flow fields. We located the zero boundaries on the positive-negative labeled patterns to determine the rotation ω_A of camera *A* using an algorithm called "detranslation" [Fermüller, 1995B, 1998]. FoE of camera *B* was obtained readily from the patterns.

Again, full flow fields and normal flow fields were generated according to the two pure rotations about the optical center of camera *A* and the arbitrarily gradient direction were assigned to each image point. The samples of flow fields are shown in Fig. 4.21.



(a) Optical flow: Camera *A*. (b) Optical flow :Camera *B*.

(c) Normal flow: Camera *A*. (d) Normal flow :Camera *B*

Figure 4.21 Optical flow and normal flow fields generated with respect to one of the two pure rotations about the optical center of camera A.

For camera *A*, we first estimated its rotation component, and by using $\mathbf{R}_x$ obtained in the previous step, we analyzed the patterns generated by camera *B* to locate FoE. The results are shown in Table 4.5

Table 4. 5 Estimation of t_x by using synthetic data.

		Real Value	Result from Experiment
$\mathbf{R}_A$	Camera A, Motion 1	[0.00008 0.00008 0.0008]	[0.8023 0.8023 8.0237]e^{-004}
	Camera A, Motion 2	[0.00006 0.00004 0.0002]	[0.5920 0.3947 1.9733]e^{-004}
$\mathbf{t}_b$	Camera B, Motion 1	[0.0910 -0.5495 0.0459]	$\mathbf{t}_b$ /| $\mathbf{t}_b$ |: [1.9645 -11.2941]
	Camera B, Motion 2	[0.0216 -0.1352 0.0206]	$\mathbf{t}_b$ /| $\mathbf{t}_b$ |: [1.5864 -6.1978]

Finally, we obtained $\mathbf{t}_x$ up to an arbitrary nonzero scale using Equation (4.17). The result, shown in Table 4.6, is a unit vector describing the direction of the baseline.

Table 4. 6 Determination of $\mathbf{t}_x$ (up to scale)

	$\mathbf{t}_x$
Ground Truth	$[-700 \;\; 20 \;\; 80]^T$
Experiment	$[\; -0.9883 \;\; 0.0428 \;\; 0.1466]^T$

The result is a unit vector describing the direction of the baseline of the camera pair. The inter-vector angle between the ground truth and the result is calculated to be 2.09°.

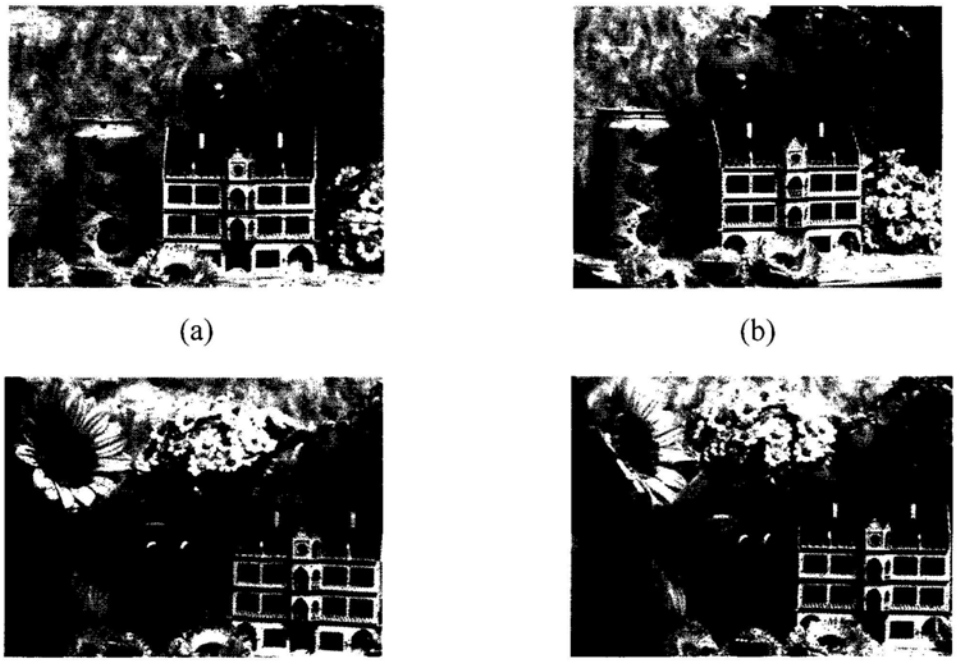
We emphasize that in the synthetic data experiments the intensity gradient direction at each image position was generated randomly. The normal flows at the various data points were thus more evenly distributed in their directions, therefore, making the intelligent selection of **s**-axes would not result in significant effect. So the experiments on the selection of **s**-axes using synthetic image data is not presented here. In this work synthetic data experiments were used to investigate accuracy, since ground truth was available. It is expected that when dealing with real image data, in which the normal flows are more concentrated in a few directions, the effect of **s**-axis selection would be significant. Therefore, experiments using real images (in which the reference solution could only be "estimated" by another method) were used primarily to examine the effect of the **s**-axis selection.

4.3.2 Real Image Experiments

Here we show results on the recovery of $\mathbf{R}_x$ (i.e., $\boldsymbol{\omega}_x$) by using real image data. We moved a camera pair on a translational platform while letting the cameras capture image sequences of the surroundings. The cameras are Dragonfly CCD cameras of resolution 640×480 pixels. We used the algorithm described in [Bouguet] to determine the intrinsic parameters of the two cameras. Normal flows were then extracted from the captured image sequences once the images were smoothed (by Gaussian Filter with n=5 and σ=1.4).

For the normal flow field of either camera under any particular translation, we first determined the optimal set of **s**-axes. With the first most optimal **s**-axis we got the first **s**-co-point positive-negative labeled pattern in the image plane. Even though the **s**-axis was chosen optimally, because of the sparseness of the usable data points under any single **s**-axis the zero-boundary could not be located very precisely on the image plane. The zero-boundary which should be a straight line could only be confined to a set of possible straight lines that well separated the "+"-labeled data points and the "-"-labeled data points. Notice that by Equation (4.4) each of such straight lines represented two possible linear constraints for the FoE of the camera under the particular translation. In other words, from the vector field of this first **s**-axis the translation of the camera could be determined up to a family of possible FoEs. Then we added the second most optimal **s**-axis to obtain a second positive-negative pattern in the image plane, determined the set of possible zero-boundaries from it, examined which of the above candidate FoEs still satisfied this second set of zero boundaries, and kept only those candidate FoEs that did. We repeated the iterations, until all candidate FoEs were located within a small enough area (one pixel in our experiment) of the image plane.

The first experiment we show here was to investigate the accuracy of the method. Different number of **s**-axes were used for different cameras as the image captured by each camera has its own texture characteristics. In any case, no more than 377 **s**-axes were used to pinpoint to 1×1-pixel accuracy the FoE of each camera under every rigid motion of the camera pair. The zero-boundaries under the determined FoEs are shown in Fig. 4.22, and they all well separated the positively labeled data points from the negatively labeled data points in the image space



(a) (b)

(c) (d)

Figure 4.22 Determination of the relative orientation of a camera pair that have substantial overlap in their visual fields. The zero-boundaries (blue lines) under the determined FoEs of the respective cameras are shown. Green dots represent negatively labeled date points; red dots represent positively leveled data points. (a) Camera A, Motion 1; (b) Camera B, Motion 1; (c) Camera A, Motion 2; (d) Camera B, Motion 2.

The computational time is 7441.825503 seconds by running the Matlab code using the PC (Pentium(R)4 CPU: 3.40GHz, RAM :1.00GB). However, the

computational time greatly depends on the accuracy of the FoE. Obviously there are two ways to speed up the computation. One is to reduce the searching accuracy of the estimation of FoE. For example we investigated the pseudo FoEs one by one pixel in this experiment, and the computational time would be greatly shortened if two by two pixels were investigated. The other way is to enlarge the tolerance of the estimation of FoE, in order to make use of less number of s-axes to analyze the positive-negative pattern.

To examine if the result was reasonable, we also calibrated the relative geometry of the camera pair using the traditional stereo calibration method [Bouguet], in which the inputs were not normal flows but manually picked corner correspondences over the chess-board pattern in the image data. This was possible because the camera pair were of such a configuration that the images taken had substantial overlap in their visual fields, rendering traditional methods possible. Table 4.7 compares the results from the proposed method and from the traditional stereo calibration method.

Table 4. 7 Results of determining ω_x of a camera pair that have substantial overlap in their visual fields.

	The Proposed Method	Traditional Stereo Calibration Method [Bouguet]
ωx	[0.0131 -0.6821 0.1005]T	[0.0270 -0.4109 -0.0100] T

The traditional stereo calibration method took much more informative input than the proposed method, and naturally its result should be more trustworthy. On comparison it can be observed that the result of the proposed method was close to such a reference. Inevitably there was error, which was mainly due to the much less informative input: the mere normal flow fields. However, the downside of the method is also its

upside. In cases where objects like checker board or distinct image features are absent from the imaged scene, or manual effort of picking feature points and establishing correspondences are inconvenient, or the two cameras have little or no overlap in their visual fields, while traditional method cannot proceed, the proposed method can still operate.

In another experiment we calibrated two cameras which had only little overlap in their fields of view. Shown in Fig. 4.25 are sample images grabbed by the respective cameras, which also display how little was the overlap between the fields of view of the cameras. The resolution of the images is 640×480.

The effect of the optimal **s**-axis selection is shown in this experiment. Lets take the image sequence captured by camera A during motion 2 for an example, which is shown in Fig. 4.25 (C). We applied our scheme of optimal determination and obtained the first optimal **s**-axis marked with red circle in Fig.4.23. Among the total 18,9887 normal flows at the image positions, the optimal **s**-axis is capable to select 5,770 normal flows to generate the positive-negative pattern. Compare to the optimal s-axis, an **s**-axis that selected the least number of normal flows marked with green circle is also shown in Fig. 4.23.

Figure 4. 23 Two-dimensional accumulation array with the optimal s-axis marked with red circle. The green circle represents the s-axis (bad axis) that selected the least number of normal flows to generate positive-nagetive pattern.

Table 4.8 shows the number of normal flows selected by the two different **s**-axes: optimal **s**-axis and bad **s**-axis.

Table 4. 8 The number of normal flows selected by the two different s-axes.

	Image domain	Optimal **s**-axis	Bad **s**-axis
Number of normal flows	18,9887	5,770	1,145
Percentage		2.933%	0.603%

The comparison of the positive-negative patterns defined by the two **s**-axes is shown in Fig.4.24. The pattern generated by the optimal **s**-axis is much denser than the one generated by the **s**-axis that only can select a small portion of the normal flows.

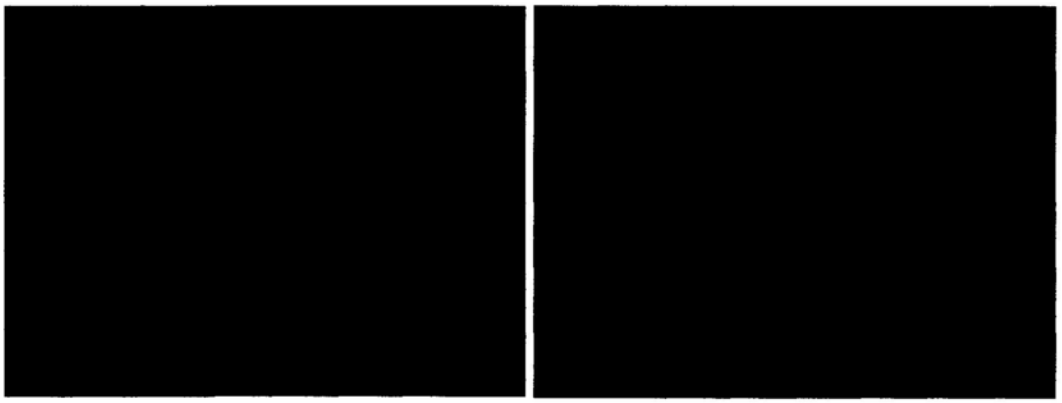
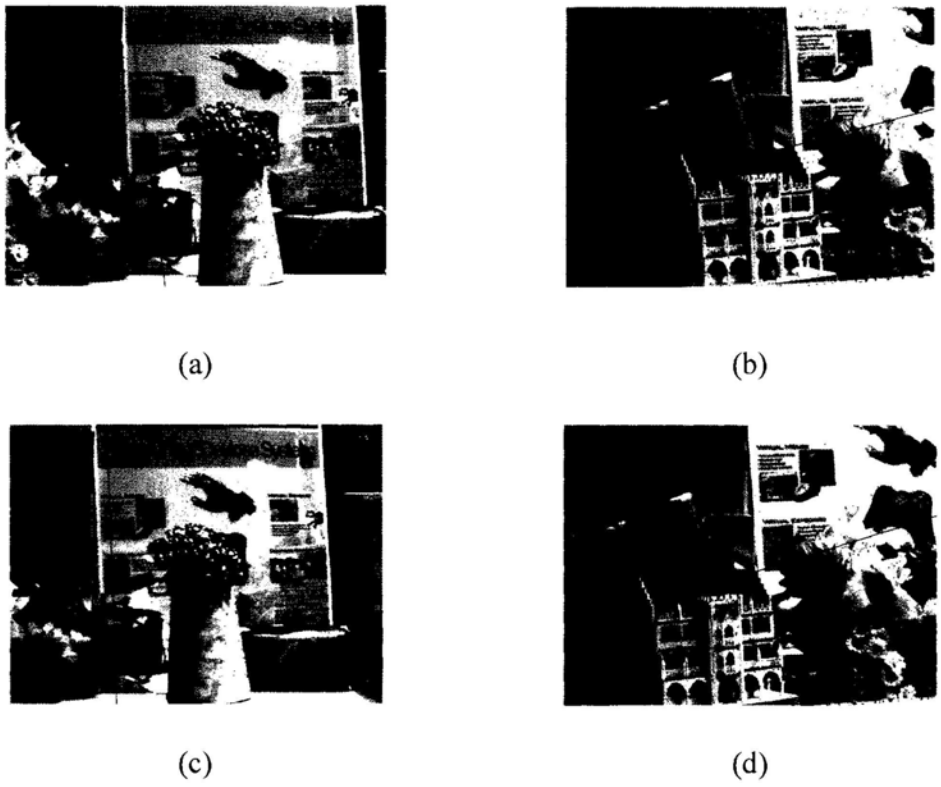


Figure 4. 24 The comparison of the positive-negative patterns defined by the two s-axes

Fig. 4.25 presents how well the zero-boundaries under the determined FoEs of the respective cameras could separate the differently labeled data points in the image space. The total computational time is 8755.714201 seconds by running the Matlab code using the PC (Pentium(R)4 CPU: 3.40GHz, RAM :1.00GB). Similar to the presented

experiment above, the computational time could be shortened greatly by reducing the accuracy of the estimation of FoE.



(a) (b)

(c) (d)

Figure 4. 25 Determination of the relative orientation of a camera pair with only little overlap in their visual fields. The zero-boundaries (blue lines) under the determined FOEs of the respective cameras are shown. Green dots represent negatively labeled date points; red dots represent positively leveled data points. (a) Camera *A*, Motion 1; (b) Camera *B*, Motion 1; (c) Camera *A*, Motion 2; (d) Camera *B*, Motion 2.

Determining the binocular geometry of such a camera pair using correspondence-based methods would be difficult since there are very few correspondences that could possibly be established across the views. However, our method demands only monocular normal flows could still operate. The result is shown in Table 4.9 and visual examination shows that the result is reasonable.

Table 4. 9 Result of determining ω_x of a camera pair that have little overlap in their visual fields.

ω_x	$[0.2157 \quad -0.2811 \quad -0.0289]^T$

4.4 Summary

In this chapter, we have presented a novel method on estimating the binocular geometry directly from monocular normal flows. Establishing motion correspondences and epipolar constraints are not required under our algorithm. Two kinds of specific motions (pure translation and pure rotation) of the stereo rig are applied to simplify the scheme of locating zero-boundaries on the positive-negative patterns. Moreover, we also proposed a scheme on optimal selection of the s-axis to improve the efficiency of the calculation. Good experimental results were obtained with both synthetic and real image data.

CHAPTER FIVE

ESTIMATION OF CAMERA'S EGO-MOTION FROM NORMAL FLOWS

In this chapter, we present a novel method on camera's ego-motion estimation for a monocular moving observer, under arbitrary translation or rotation. According to our mechanism, each normal flow will give a locus for the location of camera motion in the voting domain. The intersection of such loci determined by different normal flows will reduce the possibilities of camera motion and even pinpoint the camera motion. As the method does not track distinct features nor interpolate optical flow, it is applicable even to cases where the imaged scene is not displaying distinct features nor smooth. The method does not leave any normal flow unused in the visual data either, so it requires much less texture from the imaged scene than the traditional methods to operate. Experimental results show that the method has promising performance on office, laboratory, and urban outdoor scenes in their natural appearance that is often only sparsely textured.

This chapter is organized as follows. Section 5.1 describes how the image space and the camera motion space are represented in this work. Section 5.2 provides a new understanding of how the normal flow direction at any image position constrains the camera motion parameters. The voting scheme will be proposed in section 5.3. Finally, Experimental results on both synthetic data and real image data are presented in Section 5.4.

5.1 *Flow Vectors from Planar Image Space to Spherical Image Space*

The advantage of the spherical image space I over the planar image space I', is that the spherical image space I can represent all possible camera motion parameters $\boldsymbol{\omega}$ and $\mathbf{t}$, while the planar image itself can only encode a subset values of camera motions. Although the planar image estimation domain can be arbitrarily expanded to enlarge the subset, it still can not easily tackle the case when the camera undergoes some specific motions, translating parallel to the image plane for instance, as the FoE or FoC may be a point at infinity.

Moreover, in the planar image domain, when the camera undergoes a forward translation, $\mathbf{t}$ is usually referred to as *Focus of Expansion* (FoE). The camera motion $\mathbf{t}$ can also be described as an intersection point of all motion trajectories, *Focus of Contraction* (FoC), if the camera takes a backward translation. Provided there is merely one image point with its motion trajectory determined by the full optical flow, it is not known whether the camera undergoes a forward or a backward translation. Similarly for a camera taking a pure right-hand rotation, $\boldsymbol{\omega}$ is often represented by a point, *Right-hand Axis of Rotation* (RAoR), about which all the motion trajectories rotate. There is also a reciprocal definition of *Left-hand Axis of Rotation* (LAoR), when the camera rotates about a left-hand axis. We are facing the same problem so far when analyzing the pure camera rotation, if merely one full optical flow is provided, because we can not tell whether the camera is rotating about a right-hand axis or a left-hand axis. Fig.5.1 shows the ambiguity on camera motion analysis, supposing that only one image point with its full optical flow $\mathbf{V}'$ ($\mathbf{p}'$) is available in the planar image space.

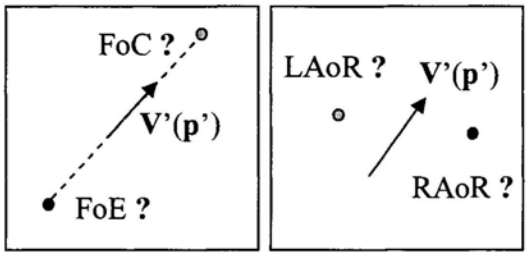


Figure 5.1 The ambiguity on camera motion analysis. The direction of camera motion can not be determined by merely one full optical flow V'(p')

However, since the spherical image space I could provide all possible camera motion parameters $\boldsymbol{\omega}$ or $\mathbf{t}$, it is not necessary for us to define whether the intersection of all motion trajectories is FoE or FoC, assuming that a camera with the spherical retina is translating toward a specific direction. More precisely, given a full optical flow $\mathbf{V}(\mathbf{p})$ on the spherical retina, the size of searching region for the possible motion will be a half as that in the planar image space I', which will be detailed in section 5.2. For the sake of convenience, the unit vectors $\widetilde{\mathbf{t}} = \mathbf{t}/\|\mathbf{t}\|$ and $\widetilde{\boldsymbol{\omega}} = \boldsymbol{\omega}/\|\boldsymbol{\omega}\|$ are adopted to represent camera's translation $\mathbf{t}$ and rotation $\boldsymbol{\omega}$ respectively, when we analyze camera's ego-motion using spherical image space. $\widetilde{\mathbf{t}}$ is the intersection point of $\mathbf{t}$ on the spherical surface, and $\widetilde{\boldsymbol{\omega}}$ is the intersection point of rotational axis $\boldsymbol{\omega}$ on the spherical surface assuming the right-hand rule is followed to describe the camera rotation.

In many places of this chapter we are concerned only with a vector's direction not its magnitude. For simplicity of presentation we shall use the notation $=_+$ to represent vector equality without regard to magnitude but only direction, i.e., it is equality up to arbitrary positive scale on its either side. On the other hand, the notation $\cong$ represents equality up to arbitrary nonzero scale as widely used in the literature; it is generally used for entities in homogeneous coordinate representation which is defined up to arbitrary nonzero scale.

In various places of the discussion we shall run into the following entities of the spherical surface S: great circle, great half-circle (one half of a great circle that is cut by a diameter of the sphere), and half-sphere (one half of the sphere that is cut by a great-circle), as locus of either $\mathbf{w}$ or $\mathbf{t}$ in various cases. Notice that such entities on S if projected through the optical center C to the planar space Γ will become line, half-line, and half-plane respectively. That will help relate solutions in S to solutions in Γ.

To recap, the input to our problem is a set of image positions where the normal flows are observable, and the desired output is two positions $\tilde{\mathbf{t}}$ and $\tilde{\boldsymbol{\omega}}$ on the full spherical surface S.

Usually only planar images, rather than spherical images , can be obtained from cameras. Consequently, the full optical flows or normal flows calculated from these planar image sequences must be projected onto the upper hemispherical retina, before the camera motion is estimated using the spherical image model. Fig.5.2 illustrates how to project flow vectors from the planar image space to the spherical image space. The upper hemispherical retina and the planar image share the same optical center C which is also the center of the spherical surface. Z-axis, perpendicular to the planar image, is defined as the optical axis. The radius CF is the focal length of camera. We use normalized image domain to simplify our mathematical model by setting CF=1. Consider any normal flow $\mathbf{v}'(\mathbf{p}')=_{+}(\cos\psi , \sin\psi)$, $\psi \in [0, \ 2\pi)$ at image position $\mathbf{p}'=(x, y)$ (a 2D-vector)on the normalized planar image. The projection of $\mathbf{p}'$ on the upper hemispherical retina is $\mathbf{p} \cong [\mathbf{p}'^{T} ,1]^{T}$ (a 3D-vector), and its corresponding normal flow $\mathbf{v}(\mathbf{p})$ at image position $\mathbf{p}$ must belong to the plane defined by optical center C and the planar normal flow $\mathbf{v}'(\mathbf{p}')$, and at the same time tangent to the hemispherical surface

(perpendicular to **p**). The normal flows' projections from the planar image space to the upper hemispherical image space can be expressed as:

$$\mathbf{v}(\mathbf{p}) =_{+} (\mathbf{p} \times \begin{bmatrix} \mathbf{v}'(\mathbf{p}') \\ 0 \end{bmatrix}) \times \mathbf{p} = (\mathbf{p} \times [\cos\psi \quad \sin\psi \quad 0] \times \mathbf{p}) \tag{5.1}$$

where $\psi \in [0, \quad 2\pi)$

Similarly, given a normal flow **v**(**p**) on the hemispherical retina I at image position **p**, its corresponding normal flows **v**'(**p**') on the planar image Γ at image position **p**' will be:

$$\mathbf{v}'(\mathbf{p}') =_{+} \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} \{[\mathbf{p} \times \mathbf{v}(\mathbf{p})] \times \mathbf{k}\}, \text{ where } \mathbf{k} = [0 \quad 0 \quad 1]^{\mathrm{T}} \tag{5.2}$$

Moreover, Equation (5.1) and (5.2) are also applicable to full optical flows' projections from the planar image space to the spherical image space and vice versa.

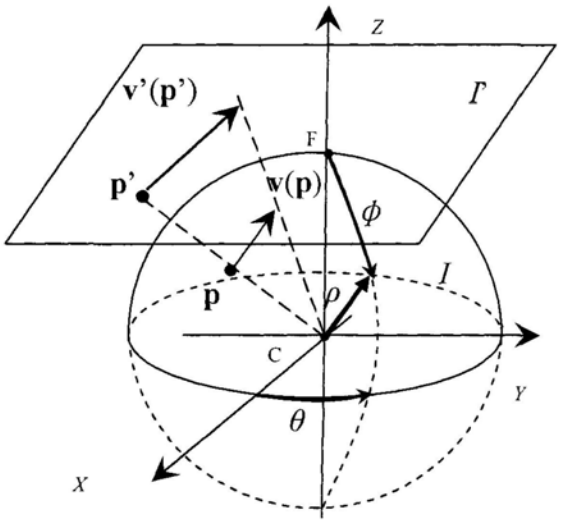


Figure 5.2 Flow vectors projected from the planar image space onto the spherical image space.

Since normal flows are the projections of full optical flows along their intensity gradient directions in the planar image space Γ, for a given normal flow **v**'(**p**')$=_{+}$ (cosψ ,

$\sin\psi)$, $\psi \in [0, \quad 2\pi)$ at image position $\mathbf{p'}=(x, y)$, the full optical flow $\mathbf{V'(p')}$ at this image position will be:

$$\mathbf{V'(p')} =_{+} (\cos(\psi+\sigma), \quad \sin(\psi+\sigma)) \tag{5.3}$$

for any $\sigma \in [-\frac{\pi}{2}, \quad \frac{\pi}{2}]$.

Consequently, according to Equation (5.1) $\mathbf{V'(p')}$'s corresponding full flow $\mathbf{V(p)}$ on the hemispherical retina at image position $\mathbf{p}$ must be:

$$\mathbf{V(p)} =_{+} (\mathbf{p} \times \begin{bmatrix} \mathbf{V'(p')} \\ 0 \end{bmatrix}) \times \mathbf{p} = (\mathbf{p} \times [\cos(\psi+\sigma) \quad \sin(\psi+\sigma) \quad 0]^{T}) \times \mathbf{p} \tag{5.4}$$

for any $\sigma \in [-\frac{\pi}{2}, \quad \frac{\pi}{2}]$.

5.2 Estimating Camera's Ego-motion Using Spherical Image Space

The spherical image space I and the planar image space I' are equivalent on investigating camera's ego-motion. The spherical image space I is adopted in this work because of its simple mathematical expression. In this section, we will also give the mathematical expression on camera's motion estimation by directly using the planar image space.

5.2.1 From Direction of Normal Flows to Direction of Pure Camera Translation

We assume a camera undergoes a pure translation in the direction of $\mathbf{t}$, as shown in Fig. 5.3. FoE is the intersection of $\mathbf{t}$ on the normalized planar image I' (focal length CF=1), since the motion shown in the figure is a forward translation. $\mathbf{V'}_{I}\mathbf{(p')}$ is the full optical flow at image position $\mathbf{p'}$ on the planar image I'. While on the upper hemispherical image I, $\mathbf{p}$ is the projection of $\mathbf{p'}$, and $\mathbf{V}_{I}\mathbf{(p)}$ is the corresponding full

optical flow at image position **p** according to Equation (5.1).In the following we will discuss the hints that a flow vector would provide for estimating the camera translation.

For the full flow vector $\mathbf{V}_t(\mathbf{p})$ on the hemispherical image I, on the spherical surface S, $\tilde{\mathbf{t}}$ (the intersection point of **t** on the spherical surface) must lie on the red great circle defined by full flow vector $\mathbf{V}_t(\mathbf{p})$ at its image position **p**, as illustrated in Fig.5.3. The great circle contains **p** and at the same time is tangent to $\mathbf{V}_t(\mathbf{p})$. Mathematically, the red great circle can be expressed as:

$$\tilde{\mathbf{t}} \cdot (\mathbf{p} \times \mathbf{V}_t(\mathbf{p})) = 0 \tag{5.5}$$

The great circle is divided into two segments by the line passing through optical center C and image position **p**. Moreover for a given $\mathbf{V}_t(\mathbf{p})$ in a specific direction, $\tilde{\mathbf{t}}$ can only locate on half-circle drawn in solid line, as shown in Fig.5.3. The great half-circle describing the locus of $\tilde{\mathbf{t}}$ is

$$\tilde{\mathbf{t}} \cdot (\mathbf{p} \times \mathbf{V}_t(\mathbf{p})) = 0 \text{ and } \tilde{\mathbf{t}} \cdot \mathbf{V}_t(\mathbf{p}) < 0 \tag{5.6}$$

Suppose that the camera with the planar image translates in the direction of **t**, which is shown in Fig.5.3. The equivalence of the above in the planar image space I' is the following. In the planar image space FoE is used to represent the camera translation **t**. Given a full optical flow $\mathbf{V}'_t(\mathbf{p}')$ at **p'**, FoE must lie on the line that contains **p'** and $\mathbf{V}'_t(\mathbf{p}')$, but only on the half of it that starts from the point **p'** and goes in the direction opposite to $\mathbf{V}'_t(\mathbf{p}')$. Mathematically, the locus of FoE is the half-line:

$$FoE = \mathbf{p}' + k\,\mathbf{V}'_t(\mathbf{p}') \text{ for all } k<0 \tag{5.7}$$

Assuming that the camera translation is a backward motion, FoC would locate on the half-line:

$$FoC = \mathbf{p}' + k\,\mathbf{V}'_t(\mathbf{p}') \text{ for all } k>0 \tag{5.8}$$

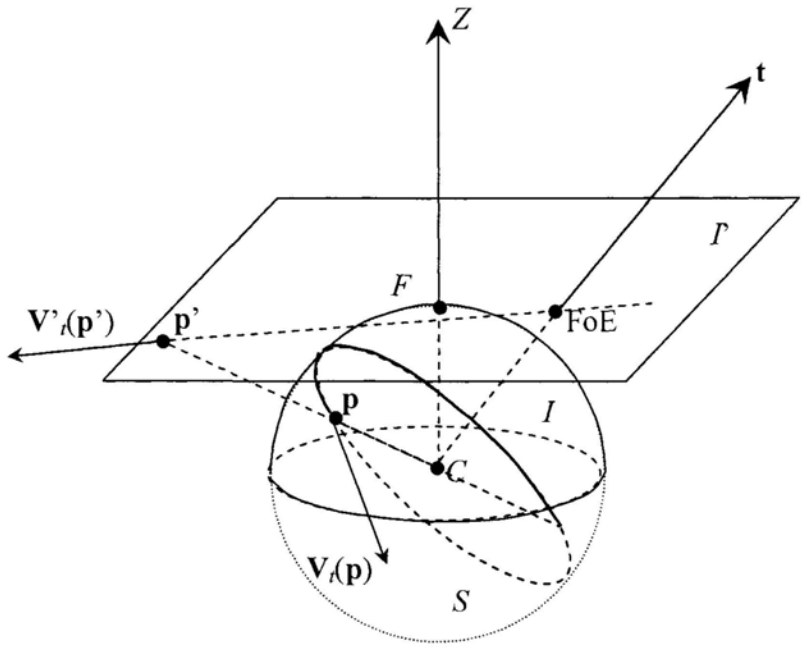


Figure 5.3 Locus of camera translation $\tilde{\mathbf{t}}$ according to a single full optical flow $\mathbf{V}_t(\mathbf{p})$ at hemispherical image position p; and the locus of FoE according to a single full optical flow $\mathbf{V}_t'(\mathbf{p}')$ at planar image position p'.

Now we consider the clue that the normal flows can provide to the estimation of camera translation. Fig.5.4 also illustrates pure camera translation in the direction of **t**. $\mathbf{v}_t'(\mathbf{p}') =_+ (\cos\psi, \sin\psi)$ is the normal flow at image position **p'** on the planar image space Γ. Then the full optical flow at **p'** will be $\mathbf{V}_t'(\mathbf{p}') =_+ (\cos(\psi+\sigma),\ \sin(\psi+\sigma))$, $\sigma \in (-\pi/2,\ \pi/2)$ according to Equation (5.3). It means the full flow on Γ will be of a direction that is $\pm\pi/2$ of the normal flow direction in the planar image space, and the two possible extreme values of the full flow are $\mathbf{V}'_t(\mathbf{p}')_{-\pi/2}$ and $\mathbf{V}'_t(\mathbf{p}')_{\pi/2}$, as shown in Fig.5.4. Therefore, on the spherical image I, its corresponding full flow at image position **p** is $\mathbf{V}_t(\mathbf{p}) =_+ ((\mathbf{p}\times[\cos(\psi+\sigma)\ \ \sin(\psi+\sigma)\ \ 0]^T)\times\mathbf{p}$, $\sigma \in (-\pi/2,\ \pi/2)$. Consequently, the above great half-circle locus of $\tilde{\mathbf{t}}$ on the spherical surface S has to

swing about the line of sight $C\mathbf{p}$ accordingly. Moreover the two possible extreme values of the full optical flow $\mathbf{V}_t(\mathbf{p})$ are $\mathbf{V}_t(\mathbf{p})_{-\pi/2}$ and $\mathbf{V}_t(\mathbf{p})_{\pi/2}$, where:

$$\mathbf{V}_t(\mathbf{p})_{-\pi/2} =_+ ((\mathbf{p}\times[\cos(\psi-\pi/2) \quad \sin(\psi-\pi/2) \quad 0]^{\mathrm{T}})\times\mathbf{p}$$

$$\mathbf{V}_t(\mathbf{p})_{\pi/2} =_+ ((\mathbf{p}\times[\cos(\psi+\pi/2) \quad \sin(\psi+\pi/2) \quad 0]^{\mathrm{T}})\times\mathbf{p}$$

At image position $\mathbf{p}$, $\mathbf{V}_t(\mathbf{p})_{-\pi/2}$ and $\mathbf{V}_t(\mathbf{p})_{\pi/2}$ define a great circle M_t on the spherical surface S, whose normal vector is:

$$\mathbf{n}(M_t) =_+ \mathbf{p}\times[\cos(\psi-\pi/2) \quad \sin(\psi-\pi/2) \quad 0]^{\mathrm{T}} \tag{5.9}$$

In Fig.5.4, the great circle M_t, which is drawn in red on the S-space, exactly divides the spherical surface S into two halves. The locus of $\tilde{\mathbf{t}}$ is the particular half-spherical surface that is constrained by $\tilde{\mathbf{t}}\cdot\mathbf{n}(M_t)<0$, or more precisely,

$$\tilde{\mathbf{t}}\cdot(\mathbf{p}\times[\cos(\psi-\pi/2) \quad \sin(\psi-\pi/2) \quad 0]^{\mathrm{T}})<0 \tag{5.10}$$

which is illustrated by the shaded half-spherical surface in Fig.5. 4.

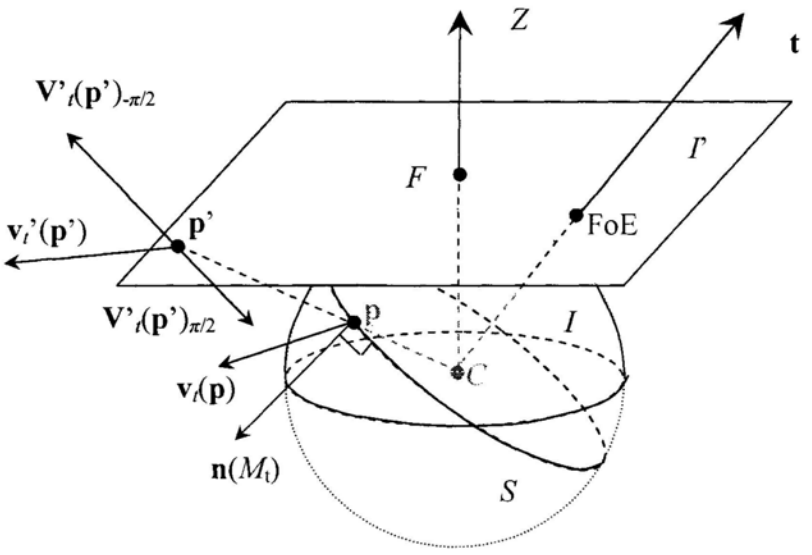


Figure 5.4 Locus of camera translation $\tilde{\mathbf{t}}$ according to a single normal flow $\mathbf{v}_t(\mathbf{p})$ at hemispherical image position p. The shaded hemispherical surface illustrates the possible location of $\tilde{\mathbf{t}}$.

The equivalence of the above in the planar image space Γ is the following. The locus of $\tilde{\mathbf{t}}$ as a half-spherical surface ($\tilde{\mathbf{t}} \cdot \mathbf{n}(M_t) < 0$) on the spherical surface S will be projected to a half-plane locus of FoE (or FoC) in the planar space. More precisely, FoE lies on the half-plane $(FoE - \mathbf{p}') \cdot \mathbf{v}_t'(\mathbf{p}') < 0$ if camera translates forward; otherwise FoC lies on the half-plane $(FoC - \mathbf{p}') \cdot \mathbf{v}_t'(\mathbf{p}') > 0$ in the case of the backward translation.

Camera translation can be determined by the mechanism described above using merely the direction information of normal flow field. Each data point gives a locus for the location of $\tilde{\mathbf{t}}$, the intersection of such loci offered by different data points will reduce the possibilities of $\tilde{\mathbf{t}}$ and even pinpoint it.

5.2.2 From Direction of Normal Flows to Axis of Pure Camera Rotation

Suppose a camera is rotating about a rotational axis $\boldsymbol{\omega}$, as shown in Fig.5.5. RAoR is the intersection of $\boldsymbol{\omega}$ on the normalized planar image Γ, as we follow the right-hand rule to describe camera rotation. $\mathbf{V}_\omega'(\mathbf{p}')$ is the full optical flow at image position $\mathbf{p}'$ on the planar image Γ. As illustrated in Fig.5.5, for a given full flow vector $\mathbf{V}_\omega(\mathbf{p})$ at image position $\mathbf{p}$ which is the projection of $\mathbf{p}'$, $\tilde{\boldsymbol{\omega}}$ must lie on the great circle passing through $\mathbf{p}$ and exactly orthogonal to $\mathbf{V}_\omega(\mathbf{p})$. This great circle, which is drawn in red on the spherical surface S, is defined by:

$$\tilde{\boldsymbol{\omega}} \cdot \mathbf{V}_\omega(\mathbf{p}) = 0 \tag{5.11}$$

However, this locus of $\tilde{\boldsymbol{\omega}}$ only guarantees that the full optical flow induced by $\boldsymbol{\omega}$ and $\mathbf{p}$ must be parallel or anti-parallel to $\mathbf{V}_\omega(\mathbf{p})$ in the hemispherical space, but not necessarily in the same direction of $\mathbf{V}_\omega(\mathbf{p})$. The line of sight $C\mathbf{p}$ divides this great circle into two haves. If the right-hand rule is followed to describe the camera rotation, the

possible locus of $\widetilde{\boldsymbol{\omega}}$ would be the half-circle drawn in red solid line on the spherical surface *S*, as shown in Fig.5.5. Mathematically, this great half-circle can be described by:

$$\widetilde{\boldsymbol{\omega}} \cdot \mathbf{V}_{\omega}(\mathbf{p}) = 0 \text{ and } \widetilde{\boldsymbol{\omega}} \cdot [\mathbf{p} \times \mathbf{V}_{\omega}(\mathbf{p})] < 0 \tag{5.12}$$

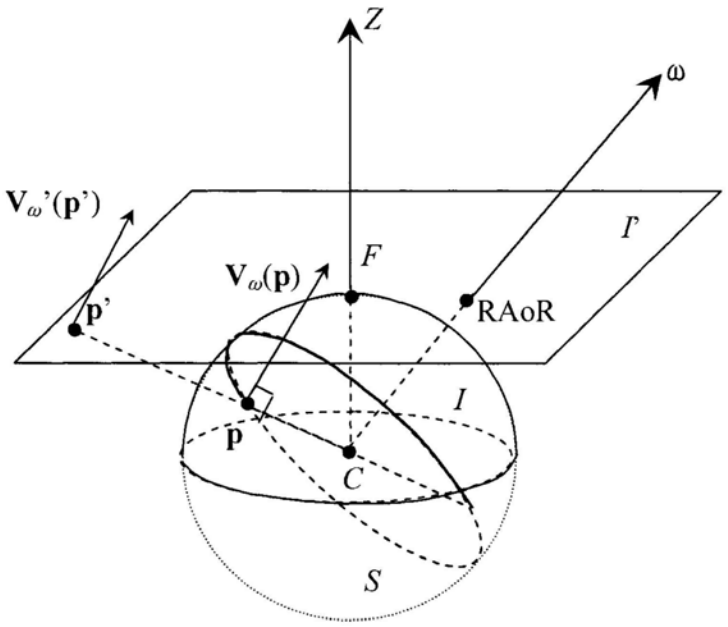


Figure 5.5 Locus of camera rotation $\widetilde{\boldsymbol{\omega}}$ according to a single full optical flow $\mathbf{V}_{\omega}(\mathbf{p})$ at hemispherical image position p.

In the planar image space, $\widetilde{\boldsymbol{\omega}}$ becomes RAoR (or LAoR), but the mechanism is not that simple. The locus of $\widetilde{\boldsymbol{\omega}}$ as a great half-circle ($\widetilde{\boldsymbol{\omega}} \cdot \mathbf{V}_{\omega}(\mathbf{p}) = 0$ and $\widetilde{\boldsymbol{\omega}} \cdot [\mathbf{p} \times \mathbf{V}_{\omega}(\mathbf{p})] < 0$) on the spherical surface *S* will be projected to a half-line locus of RAoR (or LAoR) in the planar image space. If it is an RAoR, the half-line is:

$$RAoR =_{+} \mathbf{p}' + \lambda \begin{bmatrix} \mathbf{I}_2 & \begin{matrix} 0 \\ 0 \end{matrix} \end{bmatrix} (\mathbf{V}_{\omega}(\mathbf{p}) \times \mathbf{k}) \tag{5.13}$$

where $\lambda > 0$ and $\mathbf{k} = [0 \quad 0 \quad 1]^{\mathrm{T}}$.

If $\boldsymbol{\omega}$ is an LAoR, the locus of LAoR is:

$$LAoR =_{+} \mathbf{p}' + \lambda \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} (\mathbf{V}_{\omega}(\mathbf{p}) \times \mathbf{k}) \tag{5.14}$$

where $\lambda < 0$ and $\mathbf{k} = [0 \quad 0 \quad 1]^{\mathrm{T}}$.

After investigating the locus of $\tilde{\boldsymbol{\omega}}$ offered by a specific full optical flow, we consider how to locate $\tilde{\boldsymbol{\omega}}$ by using merely normal flows' direction information. Similar to the case when we analyze the camera translation, we also start from the normal flow $\mathbf{v}_{\omega}'(\mathbf{p}') =_{+} (\cos\psi, \sin\psi)$ at image position $\mathbf{p}'$ on the planar image I', then its corresponding full optical flow at position $\mathbf{p}$ on the hemispherical image I is $\mathbf{V}_{\omega}(\mathbf{p}) =_{+} ((\mathbf{p} \times [\cos(\psi + \sigma) \quad \sin(\psi + \sigma) \quad 0]^{\mathrm{T}}) \times \mathbf{p}$, $\sigma \in (-\pi/2, \quad \pi/2)$, which is shown in Fig.5.6. The above expression indicates that the half-circle locus of $\tilde{\boldsymbol{\omega}}$ on the spherical surface S has to swing about the line of sight $C\mathbf{p}$,as the possible full optical flows at image position $\mathbf{p}$ is varying from $-\pi/2$ to $\pi/2$. The two extreme possible values of the full optical flows on the hemispherical image space are: $\mathbf{V}_{\omega}(\mathbf{p})_{-\pi/2}$ and $\mathbf{V}_{\omega}(\mathbf{p})_{\pi/2}$, which are corresponding to flows $\mathbf{V}'_{\omega}(\mathbf{p}')_{-\pi/2}$ and $\mathbf{V}'_{\omega}(\mathbf{p}')_{\pi/2}$ on the planar image I' respectively. $\mathbf{V}_{\omega}(\mathbf{p})_{-\pi/2}$ and $\mathbf{V}_{\omega}(\mathbf{p})_{\pi/2}$ at image position $\mathbf{p}$ define a great circle M_{ω} , drawn in red on the surface S in Fig.5.6, divides the spherical surface S into two half-spherical surfaces. The normal vector of the great circle M_{ω} is:

$$\mathbf{n}(M_{\omega}) =_{+} (\mathbf{p} \times [\cos(\psi - \pi/2) \quad \sin(\psi - \pi/2) \quad 0]^{\mathrm{T}}) \times \mathbf{p} \tag{5.15}$$

The locus of $\tilde{\boldsymbol{\omega}}$ is the particular half-spherical surface $\tilde{\boldsymbol{\omega}} \cdot \mathbf{n}(M_{\omega}) > 0$, and more precisely:

$$\tilde{\boldsymbol{\omega}} \cdot [(\mathbf{p} \times [\cos(\psi - \pi/2) \quad \sin(\psi - \pi/2) \quad 0]^{\mathrm{T}}) \times \mathbf{p}] > 0 \tag{5.16}$$

which is illustrated by the shaded half-spherical surface in Fig.5.6.

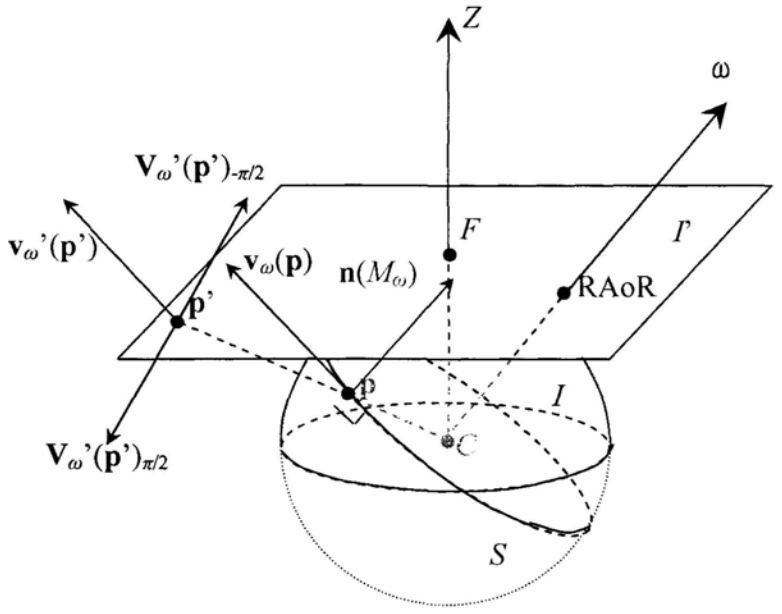


Figure 5.6 Locus of camera rotation $\tilde{\boldsymbol{\omega}}$ according to a single normal flow $\mathbf{v}_\omega(\mathbf{p})$ at hemispherical image position p. The shaded hemispherical surface illustrates the possible location of $\tilde{\boldsymbol{\omega}}$.

The equivalence of the above in the planar image space is the following. The locus of $\tilde{\boldsymbol{\omega}}$ as a half-spherical surface ($\tilde{\boldsymbol{\omega}} \cdot \mathbf{n}(M_\omega) > 0$) on the spherical surface S will be projected to a half-plane locus of $\boldsymbol{\omega}$' in the planar image space. More precisely, $\boldsymbol{\omega}$' lies on one of the half-planes separated by the line $l'(M_\omega): \mathbf{p}' + \lambda \mathbf{e}_\omega$ for all λ, where $\mathbf{e}_\omega =_+ \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} (\mathbf{n}(M_\omega) \times \mathbf{k})$ ($\mathbf{k}$=[0 0 1]$^{\mathrm{T}}$),which is the projection of the great circle M_ω onto the planar image Γ. In fact $\boldsymbol{\omega}' =_+ \mathbf{p}' + \lambda_1 \mathbf{e}_\omega + \lambda_2 \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} (\begin{bmatrix} \mathbf{e}_\omega \\ 0 \end{bmatrix} \times \mathbf{k})$ for all λ_1, and all

$\lambda_2<0$ if $\boldsymbol{\omega}$' is an RAoR, and $\boldsymbol{\omega}'=_+ \mathbf{p}'+\lambda_1\mathbf{e}_\omega+\lambda_2\begin{bmatrix}\mathbf{I}_2 & 0\\ 0\end{bmatrix}(\begin{bmatrix}\mathbf{e}_\omega\\ 0\end{bmatrix}\times\mathbf{k})$ for all λ_1, and all $\lambda_2>0$ if $\boldsymbol{\omega}$' is an LAoR.

Upon algebraic simplification, it can be shown that in the planar image space, we have:

$$\mathbf{e}_\omega =_+ \|\mathbf{p}\|^2\begin{bmatrix}\sin(\psi-\frac{\pi}{2})\\ -\cos(\psi-\frac{\pi}{2})\end{bmatrix}-[x\cos(\psi-\frac{\pi}{2})+y\sin(\psi-\frac{\pi}{2})]\begin{bmatrix}y\\ -x\end{bmatrix},$$

$$\begin{bmatrix}\mathbf{I}_2 & 0\\ 0\end{bmatrix}(\begin{bmatrix}\mathbf{e}_\omega\\ 0\end{bmatrix}\times\mathbf{k})=\|\mathbf{p}\|^2\begin{bmatrix}-\cos(\psi-\frac{\pi}{2})\\ -\sin(\psi-\frac{\pi}{2})\end{bmatrix}-[x\cos(\psi-\frac{\pi}{2})+y\sin(\psi-\frac{\pi}{2})]\begin{bmatrix}-x\\ -y\end{bmatrix}$$

Therefore, in the planar image space Γ, for a given normal flow $\mathbf{v}_\omega'(\mathbf{p}')=_+(\cos\psi,\sin\psi)$, where $\mathbf{p}'=(x, y)$, if $\boldsymbol{\omega}$' is used to represent RAoR (or LAoR), the locus of $\boldsymbol{\omega}$' is:

$$\begin{aligned}\boldsymbol{\omega}'=&\begin{bmatrix}x\\ y\end{bmatrix}+\lambda_1\{(x^2+y^2+1)\begin{bmatrix}\sin(\psi-\pi/2)\\ -\cos(\psi-\pi/2)\end{bmatrix}-[x\cos(\psi-\pi/2)+y\sin(\psi-\pi/2)]\begin{bmatrix}y\\ -x\end{bmatrix}\}\\ &+\lambda_2\{-(x^2+y^2+1)\begin{bmatrix}\cos(\psi-\pi/2)\\ \sin(\psi-\pi/2)\end{bmatrix}+[x\cos(\psi-\pi/2)+y\sin(\psi-\pi/2)]\begin{bmatrix}y\\ x\end{bmatrix}\}\end{aligned}\qquad(5.17)$$

(i) If $\boldsymbol{\omega}$' is an RAoR, the locus of RAoR in the planar image space is expressed by equation (5.17) on condition that all $\lambda_1\in R$ and all $\lambda_2<0$.

(ii) If $\boldsymbol{\omega}$' is an LAoR, the locus of LAoR in the planar image space is expressed by equation (5.17) on condition that all $\lambda_1\in R$ and all $\lambda_2>0$.

Similar to the case of camera translation, the algorithm above also provides a mechanism of determining camera rotation from the direction information of normal flow field. Each data point gives a locus for the location of $\tilde{\boldsymbol{\omega}}$, and the intersection of

such loci offered by different data points will reduce the possibilities of $\tilde{\boldsymbol{\omega}}$ and even pinpoint it. In practice, a voting scheme that transforms the image position and normal flow direction to possible locations of $\tilde{\boldsymbol{\omega}}$ in the *S*-space, similar to Hough Transform, will be used. Our voting scheme is presented at the end of this section.

5.3 Voting Scheme in the φ-θ Domain

The above analyses show that given a data point in the image steam (where the normal flow direction is available), a locus for the location of $\tilde{\mathbf{t}}$ or $\tilde{\boldsymbol{\omega}}$ can be plotted on the spherical surface *S*, and the intersection of such loci from different data points would allow the camera motion be narrowed down or even pinpointed.

In our implementation we used the spherical coordinates $\rho - \varphi - \theta$ as illustrated in Fig.5.2 to parameterize positions on the spherical surface *S* which is also the camera motion space. The spherical coordinates are related to the camera frame coordinates by:

$$\begin{cases} x = \rho \sin\phi \cos\theta \\ y = \rho \sin\phi \sin\theta \\ z = \rho \cos\phi \end{cases} \text{ where } \varphi \in [0, \pi), \theta \in [0, 2\pi]$$

Notice that for points on the camera motion space *S*, we have ρ=1. Each data point supplies a locus of $\tilde{\mathbf{t}}$ or $\tilde{\boldsymbol{\omega}}$ in the φ-θ domain, where all the votes take place domain.

Suppose that the camera undergoes a pure translation in the direction of $\tilde{\mathbf{t}}$. As pointed out in the previous analysis, given a normal flow $\mathbf{v}'_t(\mathbf{p}')=_+(\cos\psi, \sin\psi)$ (for some ψ) at image position $\mathbf{p}'=(x, y)$ on the planar image space *I'*, the locus of $\tilde{\mathbf{t}}$ as constrained by the normal flow is the half-spherical surface

$$\tilde{\mathbf{t}} \cdot \mathbf{n}(M_t) < 0$$

where $\mathbf{n}(M_t) =_+ \mathbf{p} \times [\cos(\psi - \pi/2) \quad \sin(\psi - \pi/2) \quad 0]^{\mathrm{T}}$, and $\mathbf{p} \cong [x, y, 1]^{\mathrm{T}}$ is the equivalent image position on the spherical image space I.

The above locus can be represented in the φ-θ domain in the following way.

$$\text{Let } \tilde{\mathbf{n}}(M_t) = \frac{\mathbf{n}(M_t)}{\|\mathbf{n}(M_t)\|} = [\gamma_t^x \quad \gamma_t^y \quad \gamma_t^z]^{\mathrm{T}},$$

$$\sin\alpha_t = \frac{\gamma_t^x}{\sqrt{(\gamma_t^x)^2 + (\gamma_t^y)^2}},$$

$$\sin\beta_t = \sqrt{(\gamma_t^x)^2 + (\gamma_t^y)^2}\,.$$

$$\tilde{\mathbf{t}} = [\sin\phi_t \cos\theta_t \quad \sin\phi_t \sin\theta_t \quad \cos\phi_t],$$

Then in the φ-θ domain the inequality $\tilde{\mathbf{t}} \cdot \mathbf{n}(M_t) < 0$ can be rewritten as:

$$\sin\beta_t \sin\phi_t \sin(\alpha_t + \theta_t) < -\cos\beta_t \cos\phi_t \tag{5.18}$$

$$\text{Or } \tan\phi_t < -\frac{1}{\tan\beta_t \sin(\alpha_t + \theta_t)} \qquad \theta_t \in [0, \quad 2\pi] \quad \text{for} \begin{cases} \beta_t \neq k\pi \\ \theta_t + \alpha_t \neq k\pi \\ \phi_t \neq k\pi + \pi/2 \end{cases}$$

With a number of data points available, each supplying a generally different locus for $\tilde{\mathbf{t}}$ as above, we can use a voting process similar to Hough Transform to find the best intersection of the loci. The two dimensional φ-θ domain is first discretized to become an accumulator array for receiving votes. Each data point offers a particular locus of the motion parameter in the φ-θ domain, and we just walk along the locus in the domain, casting a vote to each of the φ-θ bins we come across. Once all the votes from all the data points have been casted, the bins with the highest number of votes become the solution space of the motion parameter. How precise can the motion parameter be determined depends upon how fine is the discretization of the φ-θ domain, which can

always be increased at the expense of the computation load. Issues like this have been well addressed in the literature related to Hough Transform.

Similarly, suppose that the camera undergoes a pure rotation about the axis $\tilde{\boldsymbol{\omega}}$. As pointed out in the previous analysis, given a normal flow $\mathbf{v'}_{\omega}(\mathbf{p'})=_{+}(\cos\psi, \sin\psi)$ (for some ψ) at image position $\mathbf{p'}=(x, y)$ on the planar image space I', the locus of $\tilde{\boldsymbol{\omega}}$ as constrained by the normal flow is the half-spherical surface:

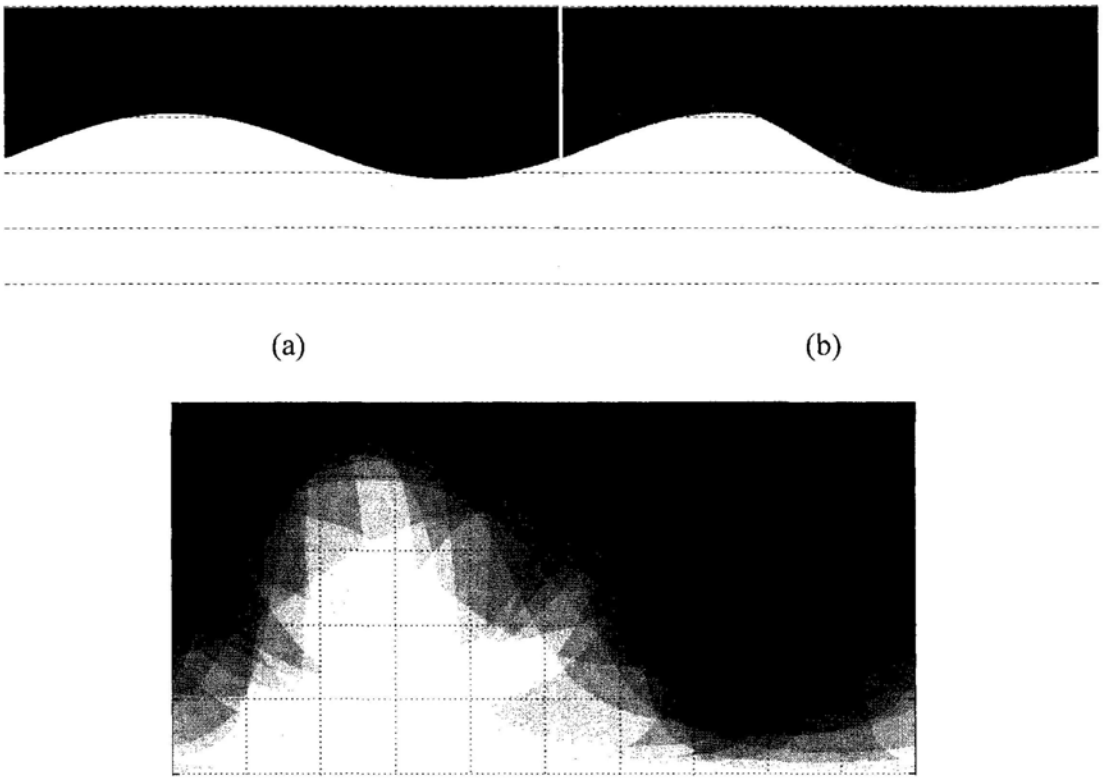
$$\sin\beta_{\omega}\sin\phi_{\omega}\sin(\alpha_{\omega}+\theta_{\omega})>-\cos\beta_{\omega}\cos\phi_{\omega} \tag{5.19}$$

where, $\sin\alpha_{\omega}=\dfrac{\gamma_{\omega}^{x}}{\sqrt{(\gamma_{\omega}^{x})^2+(\gamma_{\omega}^{y})^2}}$, $\sin\beta_{\omega}=\sqrt{(\gamma_{\omega}^{x})^2+(\gamma_{\omega}^{y})^2}$,

and $\tilde{\mathbf{n}}(M_{\omega})=\dfrac{\mathbf{n}(M_{\omega})}{\|\mathbf{n}(M_{\omega})\|}=[\gamma_{\omega}^{x}\ \gamma_{\omega}^{y}\ \gamma_{\omega}^{z}]^{\mathrm{T}}$.

In the φ-θ domain, the region that $\mathbf{v}_{\omega}(\mathbf{p})$ votes for $\tilde{\boldsymbol{\omega}}$ is a half space that lets Equation (5.19) hold.

Actually only normal flows on the planar image space are what could be obtained directly from image sequences. For each normal flow on the planar image, we firstly calculate its projection in the hemispherical image space. And then we determine which half-spherical surface this normal flow would vote for according to Equation (5.10) or Equation (5.16). Finally the half-spherical surface is projected into the φ-θ domain: the accumulator domain. An example of the φ-θ accumulator array is shown in Fig.5.7.



(a) (b)

(c)

Figure 5. 7 The voting process over the φ-θ domain for camera translation $\tilde{t}$. The green star marks the ground truth of $\tilde{t}$. (a) The accumulation array after one normal flow vector has been used. The region marked black is the half-space that the normal flow votes for. (b) The accumulation array after two normal flow vectors have been used. The darker region is the intersection that both the two normal flows vote for. (c) The accumulation array after 25 normal flow vectors have been used. The region marked red is the bins with the highest voting value, and it is the intersection that all the normal flows vote for.

5.4 Entire Solution Procedure on Camera Ego-motion Estimation

On estimating camera ego-motion, each data point with its normal flow gives a locus for the location of the camera motion, the intersection of such loci offered by different data points will reduce the possibilities of the motion and finally pinpoint it. In details, the steps could be summarized as follows:

Step 1. Calculate normal flows from the image sequences.

Step 2. Determine the great circle M_t and the particular half-spherical surface according to Equation 5.9 and Equation 5.10 (or the great circle M_ω and the particular half-spherical surface according to Equation 5.15 and Equation 5.16)for one normal flow.

Step 3. Apply voting scheme for the normal flow in the φ-θ domain.

Step 4. Check the accumulation array and look for the bin with the highest voting value, which is the solution for the camera motion.

5.5 *Experimental Results*

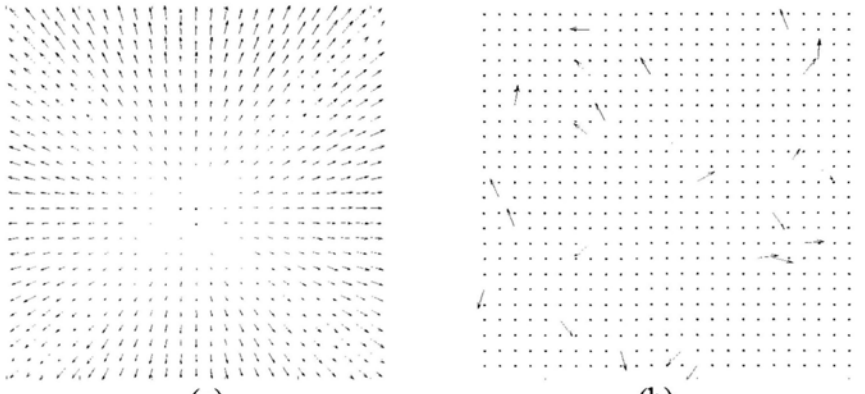
We tested our algorithm with both synthetic image data and real image sequences. The experiments with synthetic image data include the estimation of $\tilde{\mathbf{t}}$ or $\tilde{\boldsymbol{\omega}}$ when the camera undergoes pure translation and rotation respectively. The experiment with real image sequences only tested the method on estimating $\tilde{\mathbf{t}}$ when the camera took a pure translation, since the algorithm on pure camera rotation estimation is equivalent to the one on pure camera translation estimation. The φ-θ domain is of resolution 1000×2000 pixels.

5.5.1 Experiments on Synthetic Image Data

Synthetic data experiments are important because they are the ones with ground truth easily and truly accessible. We used them to examine the accuracy of the method. Normal flows are the only input, same as in the case of real image experiments.

5.5.1.1 Estimation of Camera's Pure Translation

The scene was assumed to be of a texture that caused randomly distributed intensity gradient directions in the image domain. The optical flows were induced by the assumed camera translation at each image position of the image plane, and only their components in the directions of the intensity gradients were made accessible as the normal flows. Our synthetic data consisted of images of resolution 500×500 pixels, but with 2394 (i.e., 0.9576%) of the randomly selected image positions having normal flows observable. The two flow fields are shown in Fig.5. 8.



(a) (b)

Figure 5. 8 The flow field incurred from a camera translation. (a) The full optical flow field. (b) The normal flow field under the assumption that the intensity gradient directions in the image domain were randomly distributed.

We firstly projected the normal flow vectors shown in Fig.5.8 (b) onto the hemispherical image I. For the normal flow vector at each image position, we drew its great circle M_t of the spherical surface S according to Equation (5.9), and then determine which hemispherical surface, that the particular normal flow vector votes for, would represent the locus of camera translation $\tilde{\mathbf{t}}$, according to the Inequality (5.10). Fig.5.9

gives an example of the voting scheme on the spherical surface S for a given particular normal flow **v'** (**p'**) in the planar image space.

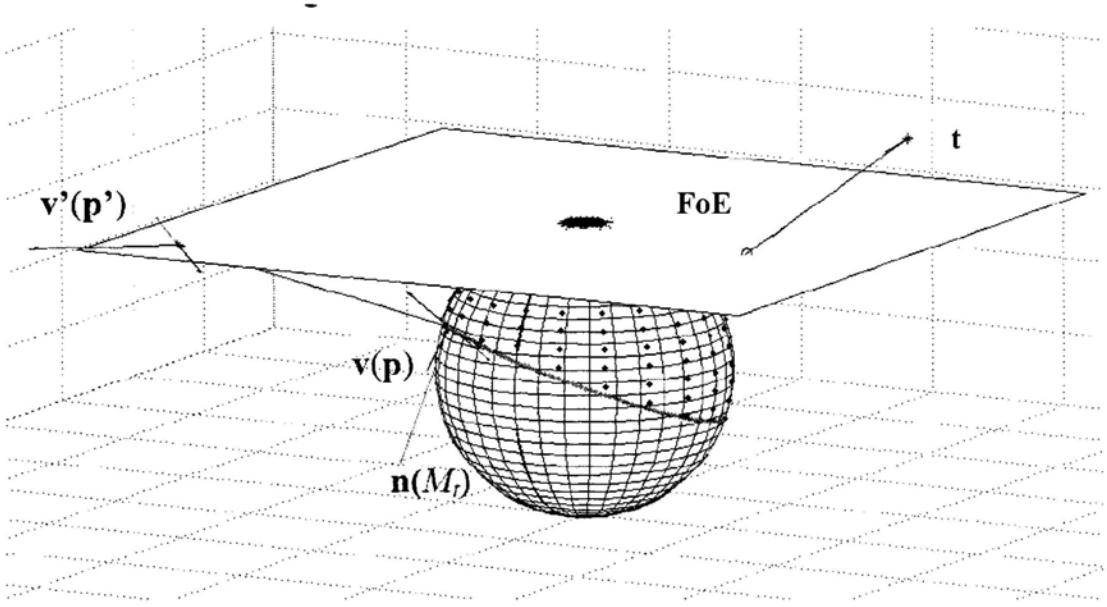


Figure 5.9 The voting scheme on the spherical surface S for a given particular normal flow assuming that camera undergoes pure translation. The normal flow v'(p') on the planar image was first projected onto the spherical surface as v(p). Then the spherical surface is divided into two hemispherical surfaces by the blue great circle M_t. The hemispherical surface marked with red dots is the region that the normal flow v'(p') votes for $\tilde{\mathbf{t}}$.

The above hemispherical surface with red dots marked can be transformed to the region marked with red dots in the φ-θ domain, as showed in Fig.5.10. The purple curve in the φ-θ domain is exactly the projection of the blue great circle M_t shown in Fig.5.9.

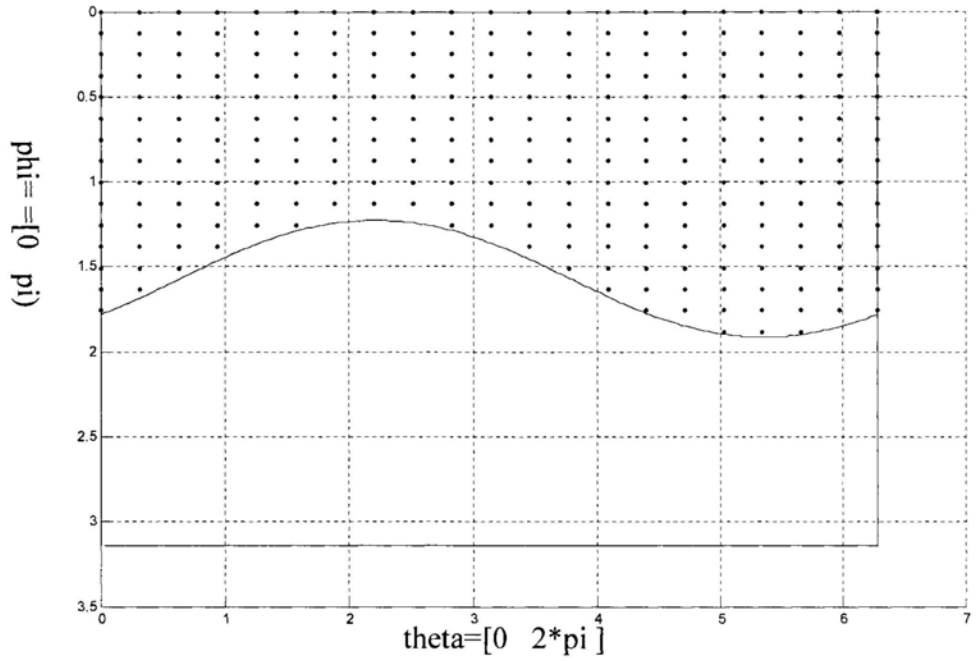


Figure 5.10 The voting scheme in the φ-θ domain for a given particular normal flow v'(p') assuming that camera undergoes pure translation. The region marked with red dots is the normal flow v'(p') voting for $\tilde{t}$.

We let the 2394 normal flows be used in the proposed method. The accumulation array in the φ-θ domain is shown in Fig.5.11. The φ-θ domain was discretized to 1000×2000 in the vote collection process.

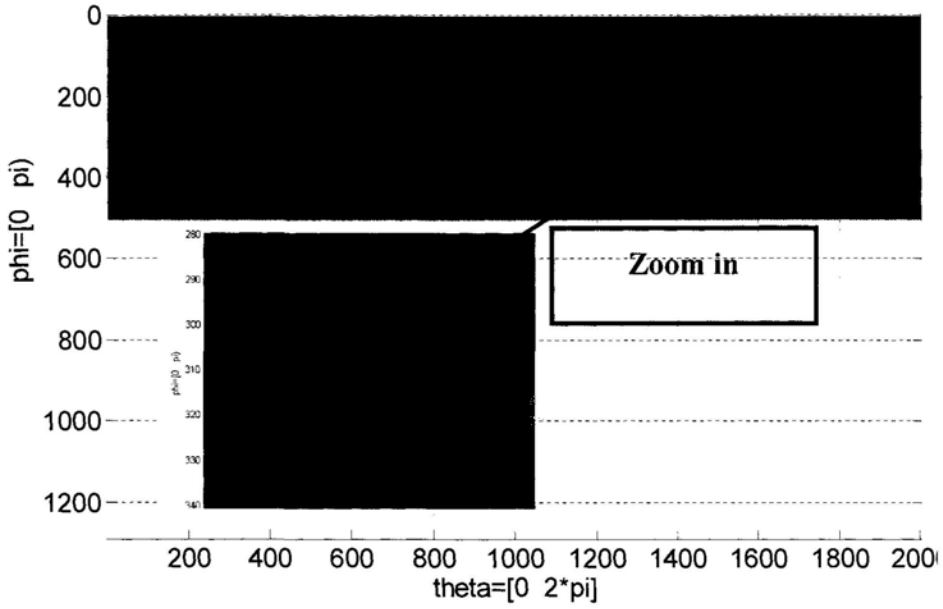


Figure 5.11 The accumulation array in the φ-θ domain for camera translation determination, with only 2394 normal flows used. The φ-θ domain was made of 1000×2000 resolution in the vote collection process. The region marked in red is the narrowed solution space for the camera translation $\tilde{t}$, which perfectly encloses the ground truth (the green star), showing that the method works as predicted.

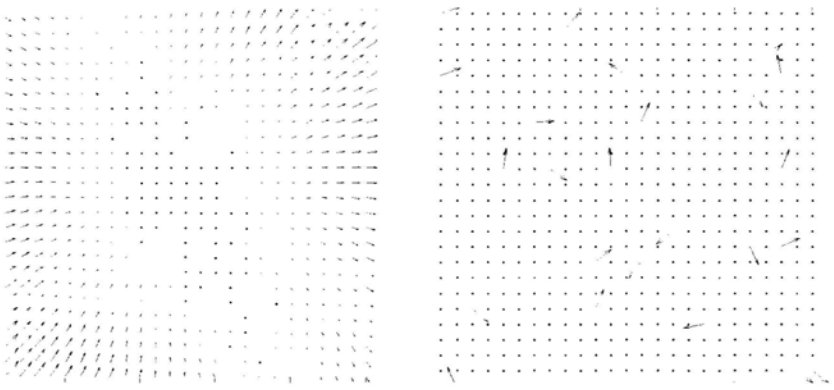
As can be seen from Fig.5.11, even with 2394 data points the method managed to narrow down the camera translation $\tilde{t}$ to a rather small solution space (the region marked in red), which perfectly encloses the ground truth of $\tilde{t}$ (the green star in Fig.5.11). The experiment indicates that not only does the method work as predicted, it also allows less texture to be present on the scene. Table 5.1 summarizes the findings from the experiment. We calculated the angle between each estimated solution and the ground truth, and used the standard deviation (STD) of these angles to describe the accuracy of the estimated result.

Table 5.1 Accuracy analysis of camera translation determination.

Ground truth of $\tilde{\mathbf{t}}$	$[-0.2673 \quad -0.8018 \quad 0.5345]^T$
STD of the determined solution zone	1.7852^o

5.5.1.2 Estimation of Camera's Pure Rotation

In this experiment we assumed that the camera was rotated about an axis $\tilde{\boldsymbol{\omega}}$ that passed through the camera's optical center. Again, an image resolution of 500×500 pixels, but with 2256 (i.e., 0.9024%) of the randomly selected image positions having normal flows observable was used, and a random distribution of the intensity gradient directions in the image domain were assumed. Fig.5.12 shows the full flow field and the normal flow field.



(a) (b)

Figure 5.12 The flow field incurred from a camera rotation. (a) The full optical flow field. (b) The normal flow field under the assumption that the intensity gradient directions in the image domain were randomly distributed.

Similarly, we firstly projected each normal flow vector onto the hemispherical image domain, and then drew its great circle M_ω of the spherical surface S according to Equation (5.15). According to the Inequality (5.16), the particular normal flow $\mathbf{v}(\mathbf{p})$ determined which hemispherical surface is the domain that it should vote for $\tilde{\boldsymbol{\omega}}$.

Fig.5.13 gives an example of the voting scheme on the spherical surface S for a given particular normal flow **v'(p')** in the planar image space.

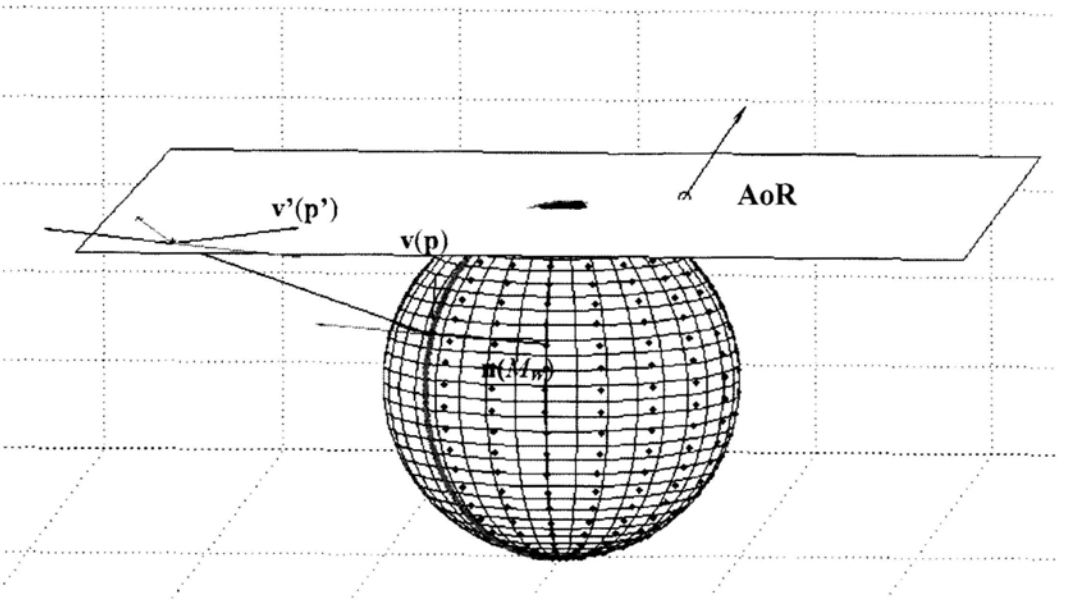


Figure 5. 13 The voting scheme on the spherical surface S for a given particular normal flow assuming that camera undergoes pure rotation. The normal flow v'(p') on the planar image was first projected onto the spherical surface as v(p). Then the spherical surface is divided into two hemispherical surfaces by the blue great circle M_ω. The hemispherical surface marked with red dots is the region that the normal flow v'(p') votes for $\widetilde{\omega}$.

The above spherical surface can be projected into the φ-θ domain, as shown in Fig.5.14. The purple curve in φ-θ domain is exactly the projection of the blue great circle M_ω shown in Fig.5.13.

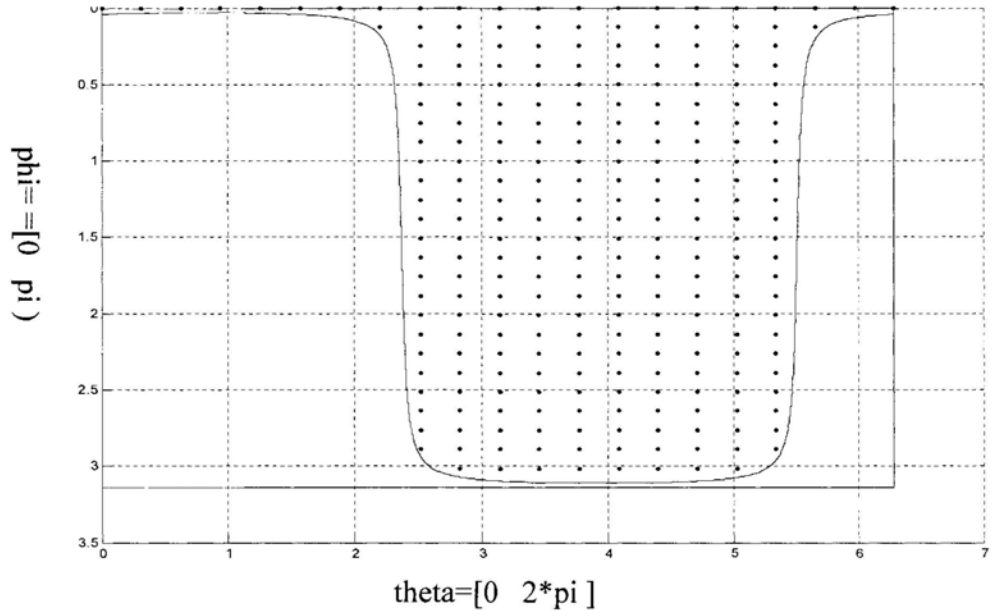


Figure 5.14 The voting scheme in the φ-θ domain for a given particular normal flow v'(p') assuming that camera undergoes pure rotation. The region marked with red dots is the possible location that normal flow v'(p') votes for $\tilde{\omega}$.

We let the 2256 data points be used in the proposed method. The accumulation array in the φ-θ domain was shown in Fig.5.15. The result again shows that the proposed method works well even with rather few data points

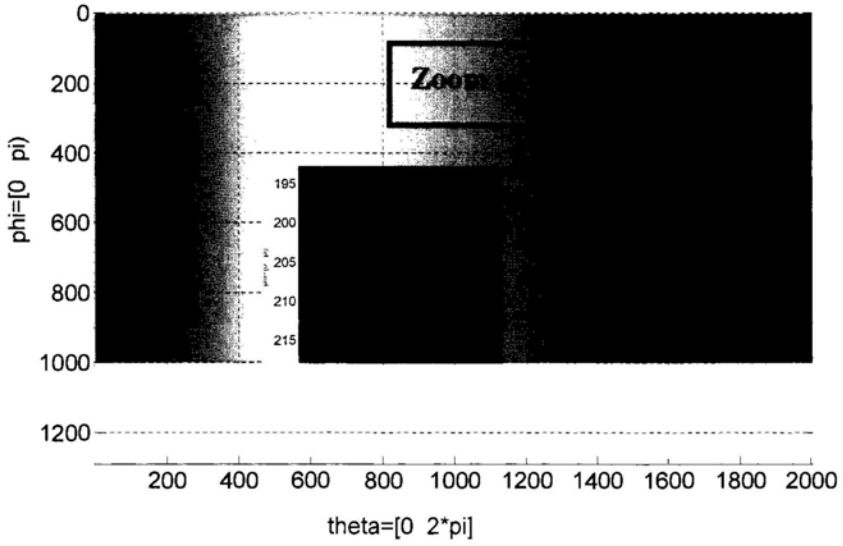


Figure 5.15 The accumulation array in the φ-θ domain for camera rotation determination, with only 2256 normal flows used. The φ-θ domain was made of 1000×2000 resolution in the vote collection process. The region marked in red is the narrowed solution space for the camera rotation $\widetilde{\omega}$, which perfectly encloses the ground truth (the green star), showing that the method works as predicted.

Tab.5.2 summarizes the findings from the experiment. We also calculated the angle between each estimated solution and the ground truth, and used the standard deviation (STD) of these angles to describe the accuracy of the estimated result.

Table 5.2 Accuracy analysis of camera rotation determination

Ground truth of $\widetilde{\omega}$	$[0.2673 \quad -0.5345 \quad 0.8018]^T$
STD of the determined solution zone	0.5035^o

Obviously, the more data points (image positions with normal flow directions observable) are used, the more precise would be the camera motion determined. The solution zone in the φ-θ domain for the motion parameters will only get smaller as more data points and in turn more loci of the motion parameters are included. Tab.5.3 shows how the precision of the solution improves as more data points came into the process. It can be seen that the improvement is much better than linear..

Table 5.3 The plot of the precision of motion determination against the number of data points used

No. of input normal flows	Number of Possible Solutions for $\widetilde{\omega}$ (under the same resolution of the voting space)	STD of the determined solution zone
25	93413	10.5226^o
90	2387	5.4537^o
225	1242	2.8902^o
380	119	1.7333^o
552	98	1.2482^o
2256	19	0.5035^o

As more input normal flows are applied, the number of possible locations of $\tilde{\omega}$ decreases dramatically, so does the STD of the estimation results.

5.5.2 Experiments on Real Image Sequences

The experiments on real image data include applying our algorithm to both highly textured images and the images without plenty of distinct features. When dealing with the highly textured image sequence, a small portion of image data is sufficient to achieve a result with the similar precision as that by using all the image data.

5.5.2.1 Experiment on the Highly Textured Images

The image sequence was captured by the Dragonfly camera when it took a pure translation on a translational platform. The sample images of resolution 640×480 pixels are shown in Fig. 5.16.

Figure 5.16 Sample images of the input image sequence which is highly textured.

The input image data was first smoothed by Gaussian filter (with n=5 and σ=1.4) before the normal flows were determined. The normal flow was computed by using

3×3 Sobel operators to estimate the spatial derivatives in the x-direction and y-direction, and by subtracting the 3×3 box-filtered values of consecutive images to estimate the temporal derivatives. Only 1% of image data with their normal flows were chosen randomly to estimate camera translation $\tilde{\mathbf{t}}$, as the texture of the scene is approximately normally distributed. The elapsed time is 167.338696 seconds by running the Matlab code using the PC (Pentium(R)4 CPU: 3.40GHz, RAM :1.00GB). For comparison, we also measured the camera translational using a traditional camera calibration method [Bouguet], in which the inputs were not optical flows, but the manually picked corner correspondences over the chess-board pattern in the consecutive images. The calibration methods by using chess-board patterns and manually picked corners are usually able to achieve the results of high accuracy. However, they have to rely on human intervention during the calibration process. Fig.5.17 shows and compares the results from the proposed method and the traditional camera calibration method. The two results are close to each other, showing that the proposed method does produce reasonable result.

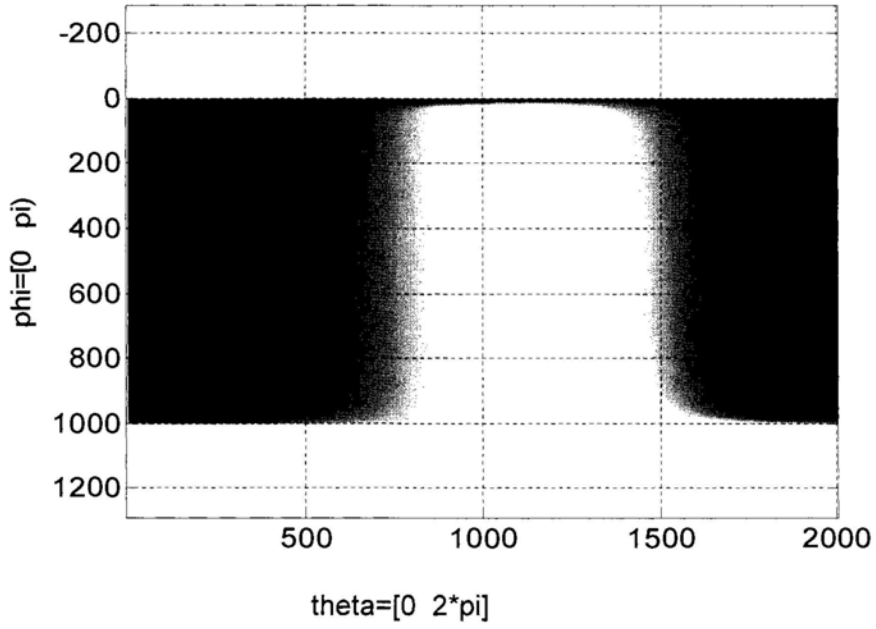


Figure 5. 17 The accumulation array in the φ-θ domain when dealing with the highly textured images. The φ-θ domain was made of 1000×2000 resolution in the vote collection process. The region marked in red is the narrowed solution space for the camera translation $\tilde{\mathbf{t}}$, and the green star is the result from an established method in the literature.

Estimation result is detailed in Tab.5.4. Two possible locations of $\tilde{\mathbf{t}}$, which we named as solution 1 & 2 respectively in Tab.5.4, were obtained after we applied 1% of the detectable normal flows (non-zero normal flow vectors) in the image domain. The estimation result by the traditional camera calibration method [Bouguet] is also listed in the following table.

Table 5.4 The experimental result on estimating camera translation using the real image data of highly textured images

Our Approach	Solution 1:	$[0.1827 \quad 0.0999 \quad 0.9781]^T$
	Solution 2:	$[0.1247 \quad 0.0655 \quad 0.9900]^T$
Traditional Camera Calibration		$[0.1190 \quad 0.0630 \quad 0.9909]^T$

We evaluated the experimental result by calculating the angles between the solutions, since $\tilde{\mathbf{t}}$ is a unit vector. The angles between these estimated vectors for $\tilde{\mathbf{t}}$ are shown in Tab.5.5. ($S1$and $S2$ represent the two solutions achieved by our algorithm, and T represents the result calculated by the traditional camera calibration method [Bouguet]. For instance, Angle <$S1$~$S2$> represents the angle between Solution 1 and Solution 2.)

Table 5.5 Evaluation of the experimental results (dealing with highly textured images) by checking the angle between the unit vectors

Angle <$S1$~$S2$>	Angle <$S1$~T>	Angle <$S2$~T>
3.9328^o	4.2623^o	0.5326^o

5.5.2.2 Experiment on the Images without Plenty of Distinct Features

We moved the camera on a translational platform against a typical office scene. The image sequences were captured by PENTAX DSLR camera at 1536×1024 resolution. Fig.5.18 shows sample images of the input image sequence. The scene was sparsely textured, which poses great difficulty to all existing direct methods (that do not require to establish explicit motion correspondences including optical flows as intermediate terms in the process).

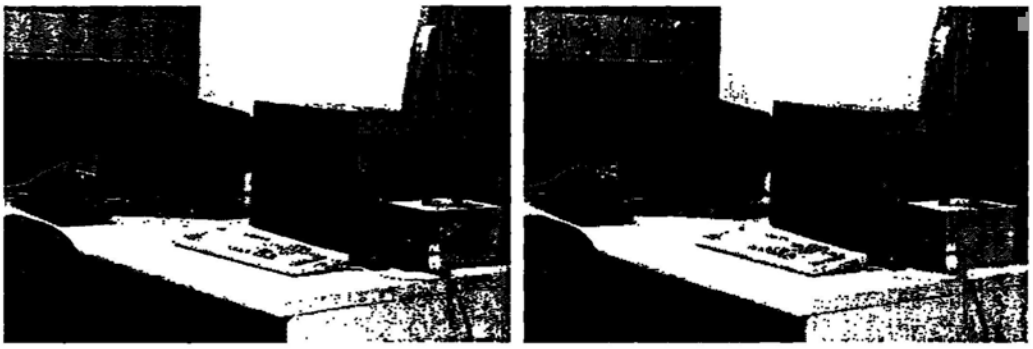


Figure 5.18 Sample images of the input image sequence without plenty of distinct features.

The input image data was smoothed by Gaussian smoothness filter (with n=5 and σ=1.4) before calculating the normal flows. Only 10,003 pixels with detectable normal flows (0.6% of all pixels in the image) were obtained due to the sparse features. All these detectable normal flows were applied to estimate camera translation $\tilde{\mathbf{t}}$. And the elapsed time is 1374.122204 seconds by running the Matlab code using the PC (Pentium(R)4 CPU: 3.40GHz, RAM :1.00GB).. We also calculated camera translational direction using the traditional camera calibration method [Bouguet]. Fig.5.19 shows and compares the results from our proposed method and from the traditional camera calibration method.

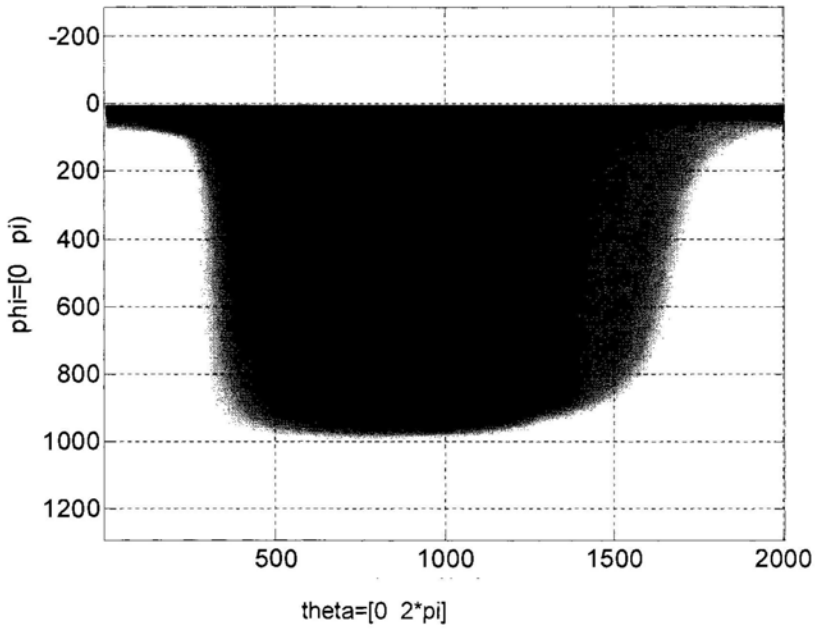


Figure 5.19 The accumulation array in the φ-θ domain when dealing with the images without plenty of distinct features. The φ-θ domain was made of 1000×2000 resolution in the vote collection process. The region marked in red is the narrowed solution space for the camera translation $\tilde{t}$, and the green star is the result from an established method in the literature.

The results from the two methods are summarized in Tab.5.6. The two possible solutions for $\tilde{t}$, referred to as solution 1 and 2 respectively in Tab.5.6, were the only ones left after we applied all the normal flow data available in the image domain. The result of the traditional camera calibration method [Bouguet] was also listed in the table.

Table 5.6 The experimental result on estimating camera translation using the real image data of images without plenty of distinct features.

Our Approach	Solution 1:	$[-0.6078\ -0.0115\ -0.7940]^T$
	Solution 2:	$[-0.5977\ -0.0169\ -0.8016]^T$
Traditional Camera Calibration		$[-0.6887\ -0.0981\ -0.7184]^T$

We compared the results also by calculating the angles between the determined translation directions. The angles $\tilde{\mathbf{t}}$ are shown in Tab5.7 ($S1$and $S2$ represent the two solutions from our method, and T represents the result from the traditional camera calibration method [Bouguet]). The angular differences are rather small, showing that the results are close.

Table 5.7 Evaluation of the experimental results (dealing with images without plenty of distinct features) by checking the angle between the unit vectors

Angle <$S1$~$S2$>	Angle <$S1$~T>	Angle <$S2$~T>
0.5910°	8.0568 °	8.4426 °

5.6 Summary

We have presented a method of determining camera motion directly from normal flows, without requiring to establish explicit motion correspondences in the process. The method is based upon a new understanding of how the normal flow direction at any image position constrains the camera motion parameters. The essence of the method includes that it allows the imaged scene not to contain distinct features that are uniquely trackable over time, nor dense texture for letting the full optical flows be interporatable. As the method does not leave any normal flow data unused, it also demands less texture from the imaged scene. Experimental results show that the method is effective in determining camera translation and camera rotation from just a small amount of normal flow data present in typical office scenes that are generally only sparsely textured. The results also show that the determined motion parameters are of reasonable accuracy.

CHAPTER SIX

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this thesis, we focused on the normal flows, the information directly obtained by the application of some simple derivative filters to the image streams, to explore what we could achieve by only making use of them directly. Our work includes two approaches. One is to estimate the inter-camera geometry of cameras directly from the monocular normal flows. The other is to calculate camera's ego-motion (pure camera translation and pure camera rotation) by directly using normal flows.

Firstly, we presented a method of determining the inter-camera geometry of cameras in chapter 4. The essence of the method is that its operation does not require presence of specific objects, parallel lines or distinct features in the imaged scene, nor does it require overlaps in the visual fields of the cameras. In a way the method is correspondence-free, as it does not depend upon correspondences across binocular views nor those across motion frames. The required normal flows are directly and locally acquirable from the image data without involving interpolation or high level processing. Experiments on synthetic data show that the solution is close to the ground truth, and experiments on real image data illustrate that the method is operable even for cameras with little overlap in their visual fields.

The method is much about locating the zero-boundary in the image space that separates the data points – the image positions with normal flows observable – that are labelled differently. The more **s**-axes are used, the more of the data points can be used in the process. We have provided a way of selecting the **s**-axes so that the same number of

the axis choices offers the maximum number of data points that are involved in the solution process. Experimental results indicate that the selection scheme does make a dramatic impact.

Secondly, in chapter 5 we proposed our direct method on estimating camera's ego-motion. Spherical image space is applied in order to avoid the ambiguity on describing camera motion. Hence, before addressing our novel method, we presented the projection scheme to project flow vectors from planar image space to spherical image space and vice versa. Our method on estimating FoE and RAoR makes use of only the direction information of normal flows. A voting scheme in the φ-θ domain is applied to simplify the 3D voting space to the 2D voting space. We tested our method using both synthetic image data and real image sequences. The experimental results showed the good performance.

6.2 Future Work

Our future works are concentrated on the camera ego-motion estimation.

In this thesis, we only proposed the framework of the solution we attempted, but there is much more works to be explored.

Firstly, the magnitude of rotational component can be determined from the magnitude information of the normal flow field without association with the scene depth. The first task of our future work is about the camera rotation magnitude estimation.

Secondly, we attempt to challenge the case of general camera motion. When the camera undergoes general motion, the normal flows are affected by both the rotational component and the translational component in unknown proportion. One thing is sure however: since the roles of rotation and translation in affecting the normal flow are

generally of the same significance and no particular order, if they can be determined, they should be determined simultaneously. We will further explore the algorithm to fulfil the task.

APPENDIX

A.1 Analysis on Equation (4.9)

In Chapter 4, Equation (4.9) can be rewritten as follows:

$$\underbrace{\begin{bmatrix}\cos\theta_1\\ \cos\theta_2\\ \cos\theta_3\end{bmatrix}}_{3\times 1}=\underbrace{\begin{bmatrix}\mathbf{B}_1^1 & \mathbf{B}_1^2 & \mathbf{B}_1^3\\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \mathbf{B}_2^3\\ \mathbf{B}_3^1 & \mathbf{B}_3^2 & \mathbf{B}_3^3\end{bmatrix}}_{3\times 9}\begin{bmatrix}\underbrace{R_{11} \quad R_{12} \quad R_{13} \quad R_{21} \quad R_{22} \quad R_{23} \quad R_{31} \quad R_{32} \quad R_{33}}_{1\times 9}\end{bmatrix}^{\mathrm{T}} \tag{6.1}$$

where $\mathbf{R}_x=\begin{bmatrix}R_{11} & R_{12} & R_{13}\\ R_{21} & R_{22} & R_{23}\\ R_{31} & R_{32} & R_{33}\end{bmatrix}$, and $\mathbf{B}_j^i$, 1×3 matrix , represents the term calculated by $\mathbf{A}_j$ and $\tilde{\mathbf{t}}_B$, and j stands for different s-axis.

Proposition 1. $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}_1^i\\ \mathbf{B}_2^i\\ \mathbf{B}_3^i\end{bmatrix}\right)=1$, where i=1, 2 ,3. $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}_1^p & \mathbf{B}_1^q\\ \mathbf{B}_2^p & \mathbf{B}_2^q\\ \mathbf{B}_3^p & \mathbf{B}_3^q\end{bmatrix}\right)=2$, where p, q=1, 2, 3, and $p\neq q$, $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}_1^1 & \mathbf{B}_1^2 & \mathbf{B}_1^3\\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \mathbf{B}_2^3\\ \mathbf{B}_3^1 & \mathbf{B}_3^2 & \mathbf{B}_3^3\end{bmatrix}\right)=3$.

Proof. In (6.1), $\mathbf{B}_j^i=\begin{bmatrix}A_j^{i1}(t_B{}^x)^2+A_j^{i2}t_B{}^xt_B{}^y+A_j^{i3}t_B{}^xt_B{}^z\\ A_j^{i1}t_B{}^xt_B{}^y+A_j^{i2}(t_B{}^y)^2+A_j^{i3}t_B{}^yt_B{}^z\\ A_j^{i1}t_B{}^xt_B{}^z+A_j^{i2}t_B{}^yt_B{}^z+A_j^{i3}(t_B{}^z)^2\end{bmatrix}^{\mathrm{T}}$ i, j=1, 2, 3, and j represents three different **s**-axes. $\tilde{\mathbf{t}}_B=[t_B^x \quad t_B^y \quad t_B^z]^{\mathrm{T}}$ is the normalized translational component with respective to the coordinates of camera B.

$$\mathbf{A}_j=[\mathbf{s}_j]_\times{}^{\mathrm{T}}[\mathbf{s}_j]=\begin{bmatrix}(s_j{}^y)^2+(s_j{}^z)^2 & -s_j{}^xs_j{}^y & -s_j{}^xs_j{}^z\\ -s_j{}^xs_j{}^y & (s_j{}^x)^2+(s_j{}^z)^2 & -s_j{}^ys_j{}^z\\ -s_j{}^xs_j{}^z & -s_j{}^ys_j{}^z & (s_j{}^x)^2+(s_j{}^y)^2\end{bmatrix}.$$

Then element $\mathbf{B}^i_j$ can be rewritten as:

$$\begin{bmatrix}\mathbf{B}^i_1\\ \mathbf{B}^i_2\\ \mathbf{B}^i_3\end{bmatrix}=\begin{bmatrix}A^{i1}_1 & A^{i2}_1 & A^{i3}_1\\ A^{i1}_2 & A^{i2}_2 & A^{i3}_2\\ A^{i1}_3 & A^{i3}_3 & A^{i3}_3\end{bmatrix}\begin{bmatrix}(t_B{}^x)^2 & t_B{}^x t_B{}^y & t_B{}^x t_B{}^z\\ t_B{}^x t_B{}^y & (t_B{}^y)^2 & t_B{}^y t_B{}^z\\ t_B{}^x t_B{}^z & t_B{}^y t_B{}^z & (t_B{}^z)^2\end{bmatrix} \quad i=1,\,2,\,3. \tag{6.2}$$

$$=\begin{bmatrix}A^{i1}_1 & A^{i2}_1 & A^{i3}_1\\ A^{i1}_2 & A^{i2}_2 & A^{i3}_2\\ A^{i1}_3 & A^{i3}_3 & A^{i3}_3\end{bmatrix}\begin{bmatrix}t_B{}^x\\ t_B{}^y\\ t_B{}^z\end{bmatrix}\begin{bmatrix}t_B{}^x & t_B{}^y & t_B{}^z\end{bmatrix}$$

Since $\mathrm{rank}(\mathbf{PQ}) \le \min\{\mathrm{rank}(\mathbf{P}), \mathrm{rank}(\mathbf{Q})\}$, we have $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}^i_1\\ \mathbf{B}^i_2\\ \mathbf{B}^i_3\end{bmatrix}\right)=1$.

Now we prove $\mathrm{rank}\left(\mathbf{B}^{pq}\right)=2$, where $\mathbf{B}^{pq}=\begin{bmatrix}\mathbf{B}^p_1 & \mathbf{B}^q_1\\ \mathbf{B}^p_2 & \mathbf{B}^q_2\\ \mathbf{B}^p_3 & \mathbf{B}^q_3\end{bmatrix}$, $p,\ q$=1, 2, 3, and $p \ne q$.

As we known, both $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}^p_1\\ \mathbf{B}^p_2\\ \mathbf{B}^p_3\end{bmatrix}\right)=1$ and $\mathrm{rank}\left(\begin{bmatrix}\mathbf{B}^q_1\\ \mathbf{B}^q_2\\ \mathbf{B}^q_3\end{bmatrix}\right)=1$. Hence the possible maximum rank of $\mathbf{B}^{pq}$ is 2. Choose two columns from $\mathbf{B}^p_j$ and $\mathbf{B}^q_j$ respectively and rewrite them into a new matrix:

$$\mathbf{B}^{pq}_{(h,k)}=\begin{bmatrix}dot([A^{h1}_1 & A^{h2}_1 & A^{h3}_1],\bar{\mathbf{t}}^n_B) & dot([A^{k1}_1 & A^{k2}_1 & A^{k3}_1],\bar{\mathbf{t}}^m_B)\\ dot([A^{h1}_2 & A^{h2}_2 & A^{h3}_2],\bar{\mathbf{t}}^n_B) & dot([A^{k1}_2 & A^{k2}_2 & A^{k3}_2],\bar{\mathbf{t}}^m_B\\ dot([A^{h1}_3 & A^{h2}_3 & A^{h3}_3],\bar{\mathbf{t}}^n_B) & dot([A^{k1}_3 & A^{k2}_3 & A^{k3}_3],\bar{\mathbf{t}}^m_B\end{bmatrix}$$

$$=\underbrace{\begin{bmatrix}A^{h1}_1 & A^{h2}_1 & A^{h3}_1 & A^{k1}_1 & A^{k2}_1 & A^{k3}_1\\ A^{h1}_2 & A^{h2}_2 & A^{h3}_2 & A^{k1}_2 & A^{k2}_2 & A^{k3}_2\\ A^{h1}_3 & A^{h2}_3 & A^{h3}_3 & A^{k1}_3 & A^{k2}_3 & A^{k3}_3\end{bmatrix}}_{\hat{\mathbf{A}}^{pq}}\underbrace{\begin{bmatrix}\bar{\mathbf{t}}^{n\,\mathrm{T}}_B & \mathbf{0}_{(3\times1)}\\ \mathbf{0}_{(3\times1)} & \bar{\mathbf{t}}^{m\,\mathrm{T}}_B\end{bmatrix}}_{\hat{\mathbf{T}}^{mn}}$$

where $h,\ k,\ m,\ n$=1, 2, 3 ($h \ne k$), and

$$\bar{\mathbf{t}}^1_B=[(t_B{}^x)^2 \quad t_B{}^x t_B{}^y \quad t_B{}^x t_B{}^z]$$

$$\bar{\mathbf{t}}_B^2 = [t_B{}^x t_B{}^y \quad (t_B{}^y)^2 \quad t_B{}^y t_B{}^z]$$

$$\bar{\mathbf{t}}_2^3 = [t_B{}^x t_B{}^z \quad t_B{}^y t_B{}^z \quad (t_B{}^z)^2].$$

Clearly, if rank($\hat{\mathbf{A}}^{pg}$)>1, rank($\mathbf{B}^{pq}_{(h,k)}$) > 1 ,because generally rank($\hat{\mathbf{T}}^{mn}$)=2. And $\hat{\mathbf{A}}^{pg}$ only depends on the 3 arbitrarily chosen **s**-axes. Hence, there always exist 3 **s**-axes such that rank($\hat{\mathbf{A}}^{pg}$)=2.

Similarly, $\operatorname{rank}\left(\begin{bmatrix} \mathbf{B}_1^1 & \mathbf{B}_1^2 & \mathbf{B}_1^3 \\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \mathbf{B}_2^3 \\ \mathbf{B}_3^1 & \mathbf{B}_3^2 & \mathbf{B}_3^3 \end{bmatrix}\right) = 3$ is also can be proved.

Since **s**-axes are chosen arbitrarily, there always exist such 3 **s**-axes to obtain Equation (6.1). By applying some elementary matrix operations, (6.1) can be arranged into:

$$\underbrace{\begin{bmatrix} f(\theta_1) \\ f(\theta_2) \\ f(\theta_3) \end{bmatrix}}_{3\times 1} = \underbrace{\begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 \end{bmatrix}}_{3\times 9} \begin{bmatrix} \underbrace{R_{11} \quad R_{12} \quad R_{13} \quad R_{21} \quad R_{22} \quad R_{23} \quad R_{31} \quad R_{32} \quad R_{33}}_{1\times 9} \end{bmatrix}^{\mathrm{T}} \tag{6.3}$$

where $f(\theta_i)$ is a scalar, and $\mathbf{C}_i$ is a 1×3 matrix with $\operatorname{rank}\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix}\right) = 1$. If equation (6.3) could be obtained, we prove $\operatorname{rank}\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix}\right) = 1$ below.

Equation (6.2) is rewritten as follows:

$$\begin{bmatrix} \mathbf{B}_1^i \\ \mathbf{B}_2^i \\ \mathbf{B}_3^i \end{bmatrix} = \begin{bmatrix} A_1^{i1} t_B{}^x + A_1^{i2} t_B{}^y + A_1^{i3} t_B{}^z \\ A_2^{i1} t_B{}^x + A_2^{i2} t_B{}^y + A_2^{i3} t_B{}^z \\ A_3^{i1} t_B{}^x + A_3^{i2} \omega t_B{}^y + A_3^{i3} t_B{}^z \end{bmatrix} \begin{bmatrix} t_B{}^x & t_B{}^y & t_B{}^z \end{bmatrix}$$

Then we further rewrite the 3×9 coefficient matrix:

$$\begin{bmatrix} \mathbf{B}_1^1 & \mathbf{B}_1^2 & \mathbf{B}_1^3 \\ \mathbf{B}_2^1 & \mathbf{B}_2^2 & \mathbf{B}_2^3 \\ \mathbf{B}_3^1 & \mathbf{B}_3^2 & \mathbf{B}_3^3 \end{bmatrix} = \underbrace{\begin{bmatrix} \begin{bmatrix} \kappa^{11} \\ \kappa^{21} \\ \kappa^{31} \end{bmatrix} \begin{bmatrix} t_B{}^x & t_B{}^y & t_B{}^z \end{bmatrix} & \begin{bmatrix} \kappa^{12} \\ \kappa^{22} \\ \kappa^{32} \end{bmatrix} \begin{bmatrix} t_B{}^x & t_B{}^y & t_B{}^z \end{bmatrix} & \begin{bmatrix} \kappa^{13} \\ \kappa^{23} \\ \kappa^{33} \end{bmatrix} \begin{bmatrix} t_B{}^x & t_B{}^y & t_B{}^z \end{bmatrix} \end{bmatrix}}_{3\times 9}$$

where $\kappa^{ji} = A_j^{i1} t_B{}^x + A_j^{i2} t_B{}^y + A_j^{i3} t_B{}^z$ is a scalar, j, i=1, 2, 3.

As presented above, the expression could be arranged in the form of (6.3) by applying some elementary matrix operations. Then the coefficient matrix of (6.3) must be:

$$\underbrace{\begin{bmatrix} \mathbf{C}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 \end{bmatrix}}_{3\times 9} = \begin{bmatrix} \begin{bmatrix} \rho_1(\kappa^{ji}, \boldsymbol{\omega}_2) \\ 0 \\ 0 \end{bmatrix} \tilde{\mathbf{t}}_B{}^{\mathrm{T}} & \begin{bmatrix} 0 \\ \rho_2(\kappa^{ji}, \tilde{\mathbf{t}}_B) \\ 0 \end{bmatrix} \tilde{\mathbf{t}}_B{}^{\mathrm{T}} & \begin{bmatrix} 0 \\ 0 \\ \rho_3(\kappa^{ji}, \tilde{\mathbf{t}}_B) \end{bmatrix} \tilde{\mathbf{t}}_B{}^{\mathrm{T}} \end{bmatrix}$$

where $\rho_1(\kappa^{ji}, \tilde{\mathbf{t}}_B)$, $\rho_2(\kappa^{ji}, \tilde{\mathbf{t}}_B)$ and $\rho_3(\kappa^{ji}, \tilde{\mathbf{t}}_B)$ are scalar functions, i,j=1, 2, 3.

Therefore, the rank $\left(\begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix} = \begin{bmatrix} \rho_1(\kappa^{ji}, \tilde{\mathbf{t}}_B) \tilde{\mathbf{t}}_B{}^{\mathrm{T}} \\ \rho_2(\kappa^{ji}, \tilde{\mathbf{t}}_B) \tilde{\mathbf{t}}_B{}^{\mathrm{T}} \\ \rho_3(\kappa^{ji}, \tilde{\mathbf{t}}_B) \tilde{\mathbf{t}}_B{}^{\mathrm{T}} \end{bmatrix} \right)$=1.

A.2 Locus of Full Flow from Normal Flow in Planar Image Space

We present here the proof of the locus of full flow from normal flow in planar image space in Chapter 5.

Consider any image position $\mathbf{p}'=(x,y)$ on the normalized image plane, where the normal flow is observed to be $\mathbf{v}'(\mathbf{p}')=_+(\cos\psi, \sin\psi)$ in the planar image space. As $\mathbf{v}'(\mathbf{p}')$ is only the projection of the true flow onto a certain direction (the direction of the local intensity gradient) in the image space, the true flow $\mathbf{V}'(\mathbf{p}')$'s direction must be in the range:

$$\mathbf{V}'(\mathbf{p}') =_+ \left(\cos(\psi+\sigma), \sin(\psi+\sigma)\right) \text{ for any } \sigma \in (-\pi/2, \pi/2)$$

The above description in the hemispherical image space is that at image position $\mathbf{p} \cong (x, y, 1)$, where the normal flow $\mathbf{v}(\mathbf{p})$ is:

$$\mathbf{v}(\mathbf{p}) =_+ [\mathbf{p} \times \begin{bmatrix} \mathbf{v}'(\mathbf{p}') \\ 0 \end{bmatrix}] \times \mathbf{p} = ([x, y, 1]^T \times [\cos\psi, \sin\psi, 0]^T) \times [x, y, 1]^T$$

(which is constructed from $\mathbf{v}'(\mathbf{p}')$ to have the property $\mathbf{v}(\mathbf{p}) \perp \mathbf{p}$), the full flow $\mathbf{V}(\mathbf{p})$ is of the following range of values:

$$\mathbf{V}(\mathbf{p}) =_+ ([x, y, 1]^T \times [\cos(\psi+\sigma), \sin(\psi+\sigma), 0]^T) \times [x, y, 1]^T \text{ for any } \sigma \in (-\pi/2, \pi/2)$$

Notice that given any flow $\mathbf{u}(\mathbf{p})$ (normal or full; a 3-vector) on the hemispherical image space I at image position $\mathbf{p} \cong [x, y, 1]^T$, the corresponding flow $\mathbf{u}'(\mathbf{p}')$ (a 2-vector) on the planar image space I' at image position $\mathbf{p}' = [x, y]^T$ 'can be determined as:

$$\mathbf{u}'(\mathbf{p}') =_+ \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} \{[\mathbf{p} \times \mathbf{u}(\mathbf{p})] \times \mathbf{k}\} \qquad \text{where } \mathbf{k} = [0, 0, 1]^T.$$

In other words, for the above full flow locus $\mathbf{V}(\mathbf{p}) =_+ ([x, y, 1]^T \times [\cos(\psi+\sigma), \sin(\psi+\sigma), 0]^T) \times [x, y, 1]^T$ for any $\sigma \in (-\pi/2, \pi/2)$ on the spherical image space, the corresponding full flow locus on the planar image space is $\mathbf{V}'(\mathbf{p}') =_+ \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} \{[\mathbf{p} \times \mathbf{V}(\mathbf{p})] \times \mathbf{k}\}$, which can be simplified to:

$$\begin{aligned} \mathbf{V}'(\mathbf{p}') &=_+ \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} \{[\mathbf{p} \times \mathbf{V}(\mathbf{p})] \times \mathbf{k}\} \\ &= \begin{bmatrix} \mathbf{I}_2 & 0 \\ & 0 \end{bmatrix} \{[\mathbf{p} \times \{(\mathbf{p} \times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix}) \times \mathbf{p}\}] \times \mathbf{k}\} \end{aligned}$$

$$= \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{[(\mathbf{p}\cdot\mathbf{p})(\mathbf{p}\times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix}) - (\mathbf{p}\cdot(\mathbf{p}\times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix}))\mathbf{p}]\times\mathbf{k}\}$$

by the identity $\mathbf{a}\times(\mathbf{b}\times\mathbf{c}) = (\mathbf{a}\cdot\mathbf{c})\mathbf{b} - (\mathbf{a}\cdot\mathbf{b})\mathbf{c}$

$$= \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{[(\mathbf{p}\cdot\mathbf{p})(\mathbf{p}\times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix}) - 0]\times\mathbf{k}\}$$

$$=_{+} \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{(\mathbf{p}\times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix})\times\mathbf{k}\}$$

because $(\mathbf{p}\cdot\mathbf{p})$ is positive

$$=_{+} \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{(\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix})\times\mathbf{k}\}$$

$$= \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{ \begin{bmatrix} -\sin(\psi+\sigma) \\ \cos(\psi+\sigma) \\ x\sin(\psi+\sigma) - y\cos(\psi+\sigma) \end{bmatrix} \times\mathbf{k}\}$$

$$= \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \{ \begin{bmatrix} -\sin(\psi+\sigma) \\ \cos(\psi+\sigma) \\ x\sin(\psi+\sigma) - y\cos(\psi+\sigma) \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\}$$

$$= \begin{bmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \\ 0 \end{bmatrix}$$

which is simply a result that is expected:

$$\mathbf{V}'(\mathbf{p}') =_{+} \begin{bmatrix} \cos(\psi+\sigma) \\ \sin(\psi+\sigma) \end{bmatrix}.$$

BIBLIOGRAPHY

[Aisbett, 1989] J. Aisbett, Optical Flow with Intensity-weighted Smoothing, *IEEE PAMI*, 11(5): 512-522, 1989.

[Anandan, 1984] P. Anandan, Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion, in SPIE Intel. Robots Comput. Vision, Vol. 521, pp. 184-194, 1984.

[Anandan, 1989] P. Anandan, A Computational Framework and an Algorithm for the Measurement of Visual Motion, *IJCV*, 2: 283-310, 1989.

[Armangué, 2003] X. Armangué, H. Araújo and J. Salvi, A Review on Egomotion by Means of Differential Epipolar Geometry Applied to the Movement of a Mobile Robot, *Patten Recognition*, 36(12): 2927 – 2944, 2003.

[Barnard, 1980] S. T. Barnard and W. B. Thompson, Disparity Analysis of Images, in *IEEE Trans. Pattern Anal. Machine Intell.*, 2(4): 333-340, July 1980.

[Beardsley, 1995] P.A. Beardsley, I.D. Reid, A. Zisserman and D.W. Murray, Active Visual Navigation Using Non-Metric Structure, in P*roc. Fifth Int"l Conf. Computer Vision*, pp. 58–64, June 1995.

[Bergen, 1992] J. R. Bergen, P. J. Burt, R. Hingorani and S. Peleg, Three-frame Algorithm for Estimating Two-component Image Motion, *IEEE PAMI*, 14(9): 886-896, 1992.

[Bjorkman, 2002] M. Bjorkman and J.O. Eklundh, Real-time epipolar geometry estimation of binocular stereo heads, in *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 24 (3), Mar. 2002.

[Black, 1992] M. J. Black, *Robust incremental optical flow*, PhD thesis, Yale University, 1992.

[Bouguet] J. Y. Bouguet, Camera Calibration Toolbox for Matlab. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc.

[Bruss, 1983] A. Bruss and B. K. P. Horn, Passive Navigation, *Computer Vision, Graphics and Image Processing*, 21: 3-20, 1983.

[Burgi, 2004] P.Y. Burgi, Motion estimation based on the direction of intensity gradient, *Image and Vision Computing*, 22 (8): 637-653, August 2004.

[Chen, 2000] H. Chen, P. Belhumeur and D. Jacobs, In search of illumination invariants, *in Proceedings of the International Conference on Computer Vision and Pattern* pp. 254–261, 2000.

[Chena, 2001] Y. S. Chena, L. G. Lioua, Y. P. Hung and C. S. Fuhb, Three-dimensional Ego-motion Estimation from Motion Fields Observed with Multiple Cameras, *Pattern Recognition*, 34(8): 1573-1583, 2001.

[Chipolla, 1993] R. Chipolla, Y. Okamoto, and Y. Kuno, Robust Structure from Motion Using Motion Parallax, *in Proc. of Int'l Conf. Computer Vision*, pp. 374–382, Berlin, May 1993.

[DeSouza, 2002] G. N. DeSouza, A.J. Jones and A.C. Kak, A World Independent Approach for the Calibration of Mobile Robotics Active Stereo Heads, in *Proc. IEEE Int'l Conf. on Robotics and Automation,* Vol. 4, pp. 3336–3341, Washington, 2002

[Dornaika, 2001A] F. Dornaika and R. Chung, An Algebraic Approach to Camera Self-calibration", *Computer Vision and Image Understanding*, Vol. .83 (3), Sept. 2001.

[Dornaika, 2001B]F. Dornaika, Self-calibration of a Stereo Rig Using Monocular Epipolar Geometry, in *Proc. of ICCV*, Vol. 2, pp. 467-472, 2001.

[Dornaika, 2003] F. Dornaika and R. Chung, Stereo geometry from 3D ego-motion streams, *IEEE Trans. On Systems, Man, and Cybernetics: Part B, Cybernetics*, Vol. 33(2): 308-323, April, 2003.

[Duric, 2000] Z. Duric, E. Rivlin and A. Rosenfeld, Qualitative Description of Camera Motion from Histograms of Normal Flow, in *Proc. of ICPR,* Vol. 3, pp. 194 -198 ,2000.

[Fassi, 2005] I. Fassi and G. Legnani, Hand to Sensor Calibration: A Geometrical Interpretation of the Matrix Equation AX=XB", *Journal of Robotic Systems,* 22(9): 497-506, July 2005.

[Faugeras, 1993] O. D. Faugeras, *Three-Dimensional Computer Visio*n, MIT Press, Cambridge, MA, 1993.

[Faugeras, 1994] O. Faugeras and T. Luong and S. Maybank, Camera self-calibration: theory and experiments, in *Proc. 3rd European Conf. Computer Vision*, Stockholm, Sweeden, 471-478, 1994.

[Fermüller,1995A] C. Fermüller and Y. Aloimonos, Direct Perception of 3D Motion from Patterns of Visual Motion, *Science*, 270, Dec. 1995, 1973-1976.

[Fermüller,1995B] C. Fermüller and Y. Aloimonos, Qualitative Egomotion, *International Journal of Computer Vision*, Vol.15, pp 7-29, 1995.

[Fermüller, 1998] C. Fermüller and Y. Aloimonos *Primates, Bees and UGV's in motion, in From Living Eyes to Seeing Machines,* S. Srinivasan (Ed.), Cambridge University Press, 1998.

[Fleet, 1990] D. J. Fleet and A. D. Jepson, Computation of Component Image Velocity from Local Phase Information, *IJCV*, 5(1): 77-104, 1990.

[Fleet, 1995] D. J. Fleet and K_ Langley, Recursive Filters for Optical Flow, *IEEE PAMI*, 17(1), 1995.

[Glazer, 1987A] F. C. Glazer, Hierarchical Gradient-based Motion Detection. In *DARPA Proc. of Image Understanding Workshop*, pp 733-748, Los Angeles, California, 1987.

[Glazer, 1987B] F. C. Glazer, Computation of Optical Flow by Multilevel Relaxation, *Technical Report COINS-TR-87-64*, Univ. of Mass., 1987.

[Grzywacz, 1990] N. M. Grzywacz and A. L. Yuille, A Model for the Estimate of Local Velocity by Cells in the Visual Cortex, *Proc. R. Soc. Lond.*, B 239: 129-161, 1990.

[Gurdjos, 2005] P. Gurdjos, P, Sturm, Methods and Geometry for Plane-Based Self-Calibration, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 491-496, 2003.

[Hanning, 2004] T. Hanning, S. Graf, and G. Pisinger, Extrinsic Calibration of a Stereo Camera System Fulfilling Generalized Epipolar Constraints, in *Proc. of Visualization, Imaging, and Image Processing*, pp. 1-5, 2004.

[Hartley, 1994] R. Hartley, An algorithm for self calibration from several views, in *Proc. Conf. Computer Vision and Pattern Recognition*, Seattle, Washington, USA, 908-912, June 1994.

[Heeger, 1988] D. J. Heeger, Optical Flow Using Spatiotemporal Filters, *IJCV*, 1:279-302, 1988.

[Heikkila,1996] J. Heikkila, O. Silven, Calibration Procedure for Short Focal Length Off-the-Shelf CCD-Cameras, *ICPR*(I: 166-170), 1996.

[Heikkila, 2000] J. Heikkila, Geometric Camera Calibration Using Circular Control Points, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22(10), 1066-1077, Oct 2000.

[Heikkonen, 1995] J. Heikkonen, Recovering 3-D Motion Parameters from Optical Flow Field Using Randomized Hough Transform, *Pattern Recognition Letters*, 16: 971-978, 1995.

[Horaud,1998] R. Horaud and G. Csurka, Self- calibration and Euclidian Reconstruction Using Motion of a Stereo Rig, in *Proc. of ICCV*, pp. 96-103, 1998.

[Horaud, 2000] R. Horaud, G. Csurka and D. Demirdijian, Stereo Calibration from Rigid Motions, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(12): 1446-1452, 2000.

[Horn, 1981] B. K. P. Horn and B. Schunck, Determining Optical Flow, *Artificial Intelligence*, Vol. 17, pp 185-203, 1981.

[Horn, 1990] B. K. P. Horn, Relative Orientation, *Int'l J. Computer Vision*, vol. 4, no. 1, pp. 58–78, June 1990.

[Jain, 1995] R. Jain, R. Kasturi, B.G. Schunck, *Machine Vision*, McGraw-Hill, Inc, 1995.

[Jaynes, 2004] C. O. Jaynes, Multi-view Calibration from Planar Motion Trajectories, *Image Vision Computing*, 22(7): 535–550, 2004.

[Junejo, 2006] I. Junejo, X. Cao, and H. Foroosh, Geometry of a Non-Overlapping Multi-Camera Network, *in Proc. of 5th IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS)*, pp. 43-43, 2006.

[Kalivas, 1991] D. S. Kalivas and A. A. Sawchuk, A Region Matching Motion Estimation Algorithm, *CVGIP*, 54(2): 275-288, 1991.

[Kanatani, 1993] K. Kanatani, *Geometric Computational for Machine Vision*, Clarendon Press, Oxford, 1993.

[Knight, 2000A] J. Knight and I. Reid, Self-calibration of a Stereo Rig in a Planar Scene by Data Combination, in *Proc. of the International Conference on Pattern Recognition*, 1411–1414, Sept. 2000.

[Knight, 2000B] J. Knight and I Reid, Binocular Self-Alignment and Calibration from Planar Scenes, in *Proc. of ECCV (II)*, pp. 462 – 476, 2000.

[Knight. 2000C] J. Knight and I. Reid, Active Visual Alignment of a Mobile Stereo Camera Platform, *in Proc.of IEEE International Conference on Robotics and Automation*, pp. 3203-3208, San Francisco, CA, April, 2000.

[Lee, 1980] D. N. Lee, The Optic Flow Field: The Foundation of Vision, *Phil. Trans. Roy. Soc. London B*, vol. 290, pp. 169-179, 1980.

[Longuet-Higgins, 1980] H. C. Longuet-Higgins and K. Prazdny, The Interpretation of A Moving Retinal Image, *Proc. R. Soc. Lond.*, B 208:385-367, 1980.

[Lustman, 1987] F. Lustman, O.D. Faugeras, and G. Toscani, Motion and Structure from Motion from Point and Line Matching, in *Proc. of First Int'l Conf. Computer Vision*, pp. 25–34, London, 1987.

[Ma, 1996] S. Ma. A Self-calibration Technique for Active Vision Systems, *IEEE Transactions on Robotics and Automation*, Vol. 12(1), pp. 114-120, February, 1996.

[Malis, 2002] E. Malis and R. Cipolla, Camera Self-calibration from Unknown Palar Structure Enforcing the Multiview Constraints between Collineations, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(9):1268-1272, 2002.

[Malm, 2001] H., Malm and A. Heyden, Stereo Head Calibration from a Planar Object, in *Proc. Conf. Computer Vision and Pattern Recognition*, 657-662, Dec. 2001.

[Maybank, 1992] S. J. Maybank and O. Faugeras, A Theory of Self-calibration of a Moving Camera, *Int' Journal of Computer Vision*, Vol. 8(2), 123-152, Aug. 1992.

[Moons, 1996] T. Moons, L. Van Gool, M. Proesmans and E. Pauwels, Affine Reconstruction from Perspective Image Pairs with a Relative Object-Camera Translation in Between, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(1): 77–83, Jan. 1996.

[Nagel, 1989] H. H. Nagel, On A Constraint Equation for the Estimation of Displacement Rates in Image Sequences, *IEEE PAMI*, 11(1): 13-30, 1989.

[Nedevschi, 2007] S. Nedevschi, C. Vancea, T. Marita and T. Graf, Online Extrinsic Parameters Calibration for Stereovision Systems Used in Far-Range Detection Vehicle Applications, in *IEEE Trans. on Intelligence Transportation Systems*, 8(4):651-660 , Dec. 2007.

[Neubert, 2002] J. Neubert and N. J. Ferrier, Robust Active Stereo Calibration, in *Proc. of ICRA*, Vol. 3, pp. 2525-2531, 2002.

[Park, 1994] F. C. Park and B. J. Martin, Robot Sensor Calibration: Solving AX=XB on the Euclidean Group, *IEEE Transactions on Robotics and Automation*, 10(5):717 – 721, Oct 1994.

[Prazdny, 1981] K. Prazdny, Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinear Moving Observer, *Computer Vision, Graphics and Image Processing*, 17(3):238-248, 1981.

[Schnorr, 1991] C. Schnorr, Determining Optical Flow for Irregular Domains by Minimizing Quadratic Functionals of A Certain Class, *IJCV*, 6(1): 25-38, 1991.

[Schnorr, 1992] C. Schnorr, Computation of Discontinuous Optical Flow by Domain Decomposition, *IEEE PAMI*, 8(2): 153-165, 1992.

[Schunck, 1985] B. G. Schunck, Image Flow: Fundamentals and Future Research, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.560-571, June 1985,

[Scott, 1987] G. L. Scott, Four-line Method of Locally Estimating Optic Flow, *Image and Vision Computing*, 5(2): 67-72, 1987.

[Shiu, 1989] Y.C. Shiu and S. Ahmad, Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form AX = XB, *IEEE Transactions on Robotics and Automation* Vol.5, pp. 16-29, 1989.

[Sinclair, 1994] D. Sinclair, A. Blake and D. Murray, Robust Estimation of Ego-motion from Normal Flow, *Int. Journal of Computer Vision*, 13(1):57-69, September 1994.

[Silva, 1996] C. Silva and J. Santos-Victor, Direct egomotion estimation, in *Proc. of ICPR*, Vol. 1, pp. 702-706, 1996.

[Silva, 1997] C. Silva and J. Santos-Victor, Robust egomotion estimation from the normal flow using search subspaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9): 1026-1034, Sept. 1997.

[Simoncelli, 1991] E. P. Simoncelli, E. H. Adelson and D. J. Heeger, Probability Distributions of Optical Flow, In *IEEE Proc. of CVPR*, pp 310-315, 1991.

[Singh, 1990] A. Singh, An Estimation-theoretic Framework for Image Flow Computation, *In Proc. of ICCV*, pp 168-177, Osaka_ Japan, 1990.

[Sobel, 1974] I. Sobel, On Calibrating Computer Controlled Cameras for Perceiving 3D Scenes, *Artificial Intelligence*, Vol.5, pp 185-198, 1974.

[Sobey, 1991] P. Sobey and M. V. Srinivasan, Measurement of Optical Flow by A Generalized Gradient Scheme, *J. Opt. Soc. Am.*, A 8(9): 1488-1498, 1991.

[Sutton, 1983] M. A. Sutton, W. J. Walters, W. H. Peters, W. F. Ranson and S. R. McNeil, Determination of Displacement Using an Improved Digital Correlation Method, *Image and Vision Computing*, 1(3): 133-139, 1983.

[Takahashi, 1988] H. Takahashi and F. Tomita, Self-calibration Of Stereo Cameras, in *Proc. 2nd Int'l Conference on Computer Vision*, 123-128, 1988.

[Tikhonov, 1977] A. Tikhonov and V. Arsenin, *Solution of Ill-posed Problems*, Wash. DC: Winston, 1977.

[Tistarelli, 1990] M. Tistarelli and G. Sandini, Estimation of Depth from Motion Using Anthropomorphic Visual Sensor, *Image and Vision Computing*, 8:271-278,1990.

[Tsai, 1986] R. Tsai, An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision, in *Proc. of CVPR'86*. 1986.

[Unal, 2007] G. Unal and S. Soatto, A Variational Approach to Problems in Callibration of Multiple Cameras, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(8):1322-1338, 2007.

[Uras, 1988] S. Uras, F. Girosi, A. Verri, and V. Torre, A Computational Approach to Motion Perception, *Biological Cybernetics*, 60:79-87,1988.

[Watson, 1985] A. B. Watson and A. J. Ahumada Jr., Model of Human Visual Motion Sensing, *J. Opt. Soc. Am.*, A 2(2): 322-341, 1985.

[Weng, 1992] J. Weng, P. Cohen and P.M. Henriou, Camera Calibration with Distortion Models and Accuracy Evaluation, in *IEEE Trans. Pattern Anal. Machine Intell.* 14 (10): 965-979, 1992.

[Wolf, 1983] P.R. Wolf, *Elements of Photogrammetry*, McGraw-Hill,New York, 1983.

[Zhang, 1996] Z. Zhang, Q.-T. Luong, and O. Faugeras, Motion of an Uncalibrated Stereo Rig: Self-calibration and Metric Reconstruction, in *IEEE Trans. on Robotics and Automation*, Vol. 12 (1), 103-113, February 1996.

[Zhang, 1999] Z. Zhang, Flexible Camera Calibration by Viewing a Plane from Unknown Orientations, in *Proc. ICCV*, pp. 666-673, Corfu, Greece, Sept. 1999.

[Zhang, 2006] Z. Zhang, P. Cui and H. Cui, Recovery of Egomotion from Optical Flow with Large Motion Based on Subspace Method, in *Proc. of ROBIO*, pp. 555-560, 2006.

[Zisserman, 1995] A. Zisserman, P.A. Beardsley and I.D. Reid, Metric Calibration of a Stereo Rig, in *Proc. IEEE Workshop Representation of Visual Scenes*, pp. 93–100, June 1995.