Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM-Algorithm

Wallace James MacLean

A thesis submitted in conformity with the requirements for the degree of
Doctor of Philosophy,
Graduate Department of Electrical & Computer Engineering,
in the
University of Toronto

Recovery of Egomotion and Segmentation of Independent Object Motion Using the EM-Algorithm

Ph.D. 1996

Wallace James MacLean

Department of Electrical & Computer Engineering

University of Toronto

Abstract

Motion in image sequences can result from the motion of the observer (*egomotion*) and from the presence of independently moving objects (IMOs) within the field of view of the observer. Any vision system intended for an observer capable of motion needs the ability to distinguish between these two possibilities in order to successfully perform navigation and collision avoidance tasks.

One approach to motion segmentation is to perform a statistical clustering on a set of local constraints on 3-D motion in the image. This thesis proposes two new methods, based on the EM algorithm, to perform robust motion segmentation on image sequences that contain IMOs. The first method uses statistical clustering of linear and bilinear constraints (derived from computed optical flow using subspace methods) on 3-D translation and rotation. The problems of outlier detection, and determining number of processes and their initial parameters for the EM algorithm are considered. Also, analysis of the effects of IMO boundaries on linear constraints, as well as a derivation for the removal of bias inherent in translation estimates from linear constraints, are presented. Effects of fixation on detection of IMOs are considered. A framework for hypothesizing about motions underlying a set of constraint clusters is detailed.

There exist situations in which 3-D motion constraints are not sufficient to perform segmentation. The second method tracks depth-structure over time and evaluates

rigidity allowing IMOs to be identified as outliers.

Results obtained from four image sequences are presented. The first sequence is synthetic optic flow generated from a depth map and contains one IMO. The second sequence was captured from a robot moving in an industrial environment. The third sequence is similar to the second, except the flow has been generated using a regularly spaced grid which assumes no prior segmentation of the image. The fourth sequence illustrates a case in which the 3-D constraints are insufficient to perform the segmentation, and the estimation of depth structure is needed to solve the problem.

Finally, directions for future research into this problem are presented.

Acknowledgements

A Ph.D. dissertation doesn't just happen in isolation, and to suggest that this one did would be ludicrous. I would like to thank Allan Jepson for being patient and helping me see things I otherwise would have missed. He has been both a mentor and friend, and has demonstrated a high standard that I can only aspire to achieve. I am also deeply grateful to Richard Frecker, my friend and confidant. His gentle wisdom inspires me greatly. I must also thank (in no particular order) Michael Black, Sven Dickinson, Richard Mann and David Wilkes for all the intriguing and stimulating conversations, as well as their comradeship. Additional thanks to David for helping to capture the forklift sequence images. Michael Black provided the dense optic flow for the JQ sequence, and Dan Huttenlocher provided the raw images. Gene Amdur and Brian Down also deserve a vote of thanks for cheerfully answering the many questions I managed to ask them.

As in all of my endeavours, my parents have been a constant source of support and encouragement. I am deeply indebted to them. I would also like to thank Nancy Noldy for her cheerful tolerance throughout all phases of this work.

Finally, I would like to acknowledge the financial support from NSERC and OGS that allowed me the freedom to work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What is vision? David Marr opened his seminal book [61] on computational vision with this seemingly simple question. Vision is something we all do effortlessly and very well. Yet the question "What is vision?" has proven increasingly difficult under the continuing scrutiny of many vision researchers. It is probably accurate to say that vision is the most powerful of sensory modalities, and it is hardly surprising that of all the senses it occupies the most cortical space. The physics of vision is relatively well understood: properties of light and lenses and photoreceptors in the retina have been studied intensively in the last century. Even early stages of image representation in terms of neural signals leaving the retina have plausible theories which allow some insight into the information collected by the eyes. However, once higher level representations are considered the situation becomes less clear. How is visual information coded? How is attention focussed? How does contextual (*a priori*) information affect the raw information coming to the brain from the retina? How do we recognize objects? These are all active areas of vision research.

One important area of vision research concerns our ability to understand the time-varying nature of the images which reach the retina. Because creatures with eyes are typically capable of motion,[1] and because many things in the real world move for one reason or another, any vision system which can only deal with static imagery

[1] The author knows of no creatures for which this is not true.

would be inadequate for general tasks. The question of motion perception has been studied systematically since the turn of the century—Helmholtz [37] is responsible for linking structure recovery to motion, and the notion that stereopsis and motion are interchangeable concepts [61]. James Gibson [30] pioneered work into the perception of visual motion, and introduced the concept of optic flow. In terms of research aimed at using vision systems in industry, the ability to interpret images that change in time has great importance in robotics and other dynamic systems.

1.1 Why is Motion Important to Vision?

It could be argued that if nothing ever moved one would not need vision. For example, if no motion were possible, it would be pointless to identify objects (for example, coffee cups on a table) since one could never interact with them. If nothing moved, then creatures with vision would look eternally at the same image. Just as deep-sea creatures that exist in total darkness may have no need for vision[2] because of the lack of anything to see, it may be that plants have no vision because of their stationary nature. Creatures with vision use it to search for food and to avoid hazards whilst moving in their environment, including avoiding other creatures that might consider them as food. It allows them to identify other creatures of their kind to allow for social interaction. Motion appears to be an essential element relating to the need for vision.

1.2 What is an Independently Moving Object?

As an observer moves about, the images it collects representing the world change, even if everything in the field of view is static, *i.e.*, nothing *else* is moving. From this sequence of images the observer can infer information relating to its own motion, as well as the depth structure of the environment in which it is moving. This is useful

[2]Some deep sea creatures do not have eyes, and those that do have eyes may have the ability to generate their own light [31].

because it aids navigation by allowing the observer to avoid obstacles and aids in the planning of a path.

There may be objects in the field of view that are not static. Such objects are referred to as "moving independently." The image of an independently moving object (IMO) as viewed by a moving observer might seem difficult to interpret. After all, it is no longer obvious whether a region in the image is changing because of the observer's motion or because of an object that is moving in the environment. The human visual system appears to deal very well with this kind of information. Consider a baseball player running to catch a fly ball, or attempting to beat the throw to second base. In evolutionary terms, a predator had to be able to identify and track its prey during pursuit in order to be successful at hunting.

1.3 Why is it Important?

Given the importance of motion to vision, any attempt to construct a comprehensive vision system needs to address the question of interpreting visual motion. Because such a system would also need to be able to operate in a general environment it need concern itself with IMOs. The task of identifying IMOs by an observer that is moving is of tremendous importance. In industrial applications any robot that is to be capable of independent navigation would need to be able to avoid collisions with vehicles, workers, and other robots, while at the same time making sense of the static environment. This task is done effortlessly by humans, but designing algorithms to do it has proven difficult for vision researchers. This thesis deals with the problem of identifying IMOs in monocular image sequences collected by a moving observer.

1.4 Purpose & Outline of Thesis

This thesis presents two new methods for identifying IMOs in monocular image sequences captured by a moving observer. The first method relies on a statistical clustering of constraints on 3-D motion which are derived from an estimation of the

2-D image motion. The clustering algorithm, which is based on the EM-algorithm, attempts to simultaneously estimate 3-D motion parameters as well as assign a probability that a given constraint arises from a particular motion. The egomotion parameters of the observer (translation and rotation) are recovered. It is assumed that the number of IMOs will not be known in advance, and that the algorithm must estimate this as well. It will be shown that when the image motion from an IMO is locally similar to the image motion due to *egomotion* it may not be possible to identify IMO's using the motion constraints alone. In these cases the second method, based on recovering relative depth information, is proposed for detecting image points whose depth information is inconsistent with that of the static environment.

An overview of the thesis follows. Chapter 2 introduces the imaging coordinate system used throughout the thesis, and discusses perspective projection and the concept of a *motion field*. This will be related to *optic flow* and some of the issues regarding the measurement of 2-D image motion will be presented. Work on 2-D motion-based segmentation of images is also discussed in this chapter. Chapter 3 presents previous work on the problem of identifying IMOs in image sequences. First, work relating to the recovery of egomotion parameters and depth structure from image sequences is presented. This forms the background for work on detecting independent objects. Previous methods for segmentation based on 3-D motion are categorized and discussed. A discussion of the limitations of these methods is given. Chapter 4 reviews the generation of constraints on 3-D motion through the use of "subspace methods." There are two different types of constraints. "Linear subspace constraints" provide partial constraints on translation direction, and "bilinear subspace constraints" provide partial constraints on both translational direction and on rotation. The underlying assumptions of these methods are presented, with the intent of preparing for the case of multiple object motion. The relationship of subspace methods to other important methods for recovering motion parameters is discussed. An analysis of the effect of IMO boundaries on the generation of linear constraints is presented. By considering this case it is possible to better understand the structure of the linear constraints. Also, a new method is presented for eliminating the bias

inherent in translation estimates derived from linear constraints. The cause of the bias is explained, leading to an elegant method for its removal. Chapter 5 presents an overview of mixture models and their importance to the issue of data clustering. The EM algorithm, which can be used to simultaneously solve for distribution parameters and perform clustering is also discussed. Chapter 6 deals with the issues surrounding clustering of linear and bilinear constraints. These issues include choosing a form for the distributions which underly the mixture models, generating initial guesses, deciding on the number of motion processes and dealing with noise in the constraints. Methods for recovering relative depth structure are presented. Chapter 7 considers the interpretation of constraint clusters in relation to recovering the underlying motion processes. Chapters 8, 9, 10 and 11 present results from the application of these methods to one synthetic and three real image sequences. The fourth sequence presents a case in which subspace constraints are insufficient to identify IMOs, and demonstrates the use of robust statistics to identify them according to the evolution of depth structure. This is the second new method proposed for IMO detection. Finally, a summary of the contributions made by this thesis, as well as directions for future research, are presented in Chapter 12.

Chapter 2

Some Preliminaries—
Terminology & Concepts

Before reviewing previous work on motion segmentation, or starting on new methods of motion segmentation, it is useful to spend time discussing basic terminology and assumptions. Specifically, this chapter will define the coordinate systems underlying equations in this thesis, present the equations which define perspective projection, and discuss the difference between the motion field and optic flow. As my methods for motion segmentation start with optic flow as data, it is necessary to understand some of the problems inherent in the recovery and use of an optic flow field. In Chapter 8 the generation of synthetic flow, which is in fact just a computed motion field, is described. In Chapter 9 the methods used to generate flow estimates from real image sequences which are under analysis are outlined.

2.1 Imaging Coordinate Systems (Planar Receptor)

The study of low-level vision starts with the imaging system. Horn [39] defines an image as a "two-dimensional pattern of brightness." An imaging system captures images of the world by focusing incident light onto a receptor surface in order to

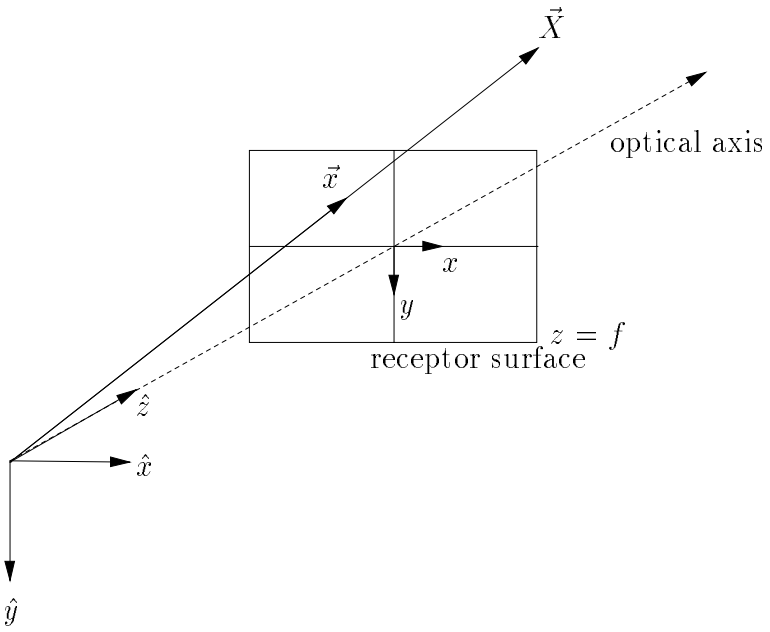


Figure 2.1: A right-handed coordinate system is attached to the imaging system. The origin coincides with the nodal point of the imaging system, and the $\hat{z}$-axis with the optical axis. The planar receptor surface lies in the $z = f$ plane. A point $\vec{X}$ in the 3-D world is imaged to a point $\vec{x}$ in the image plane. Under perspective projection the relation is $\vec{x} = \frac{f}{\vec{X}^T \hat{z}} \vec{X}$.

create these patterns of brightness. The receptor surface transduces incident light into signals that can be analyzed. These signals may be neural impulses sent to the brain, as in the human visual system, or electrical impulses such as those generated by a video camera and sent to a computer for analysis. By attaching a coordinate system to the imaging system it is possible to describe mathematically the imaging process.

Figure 2.1 shows the coordinate system underlying the equations in this thesis. A right-hand system is used with the $\hat{z}$-axis aligned with the optical axis of the imaging system. The origin coincides with the nodal point of the imaging system—this means that under perspective projection (see below) all rays imaged on the receptor surface will pass through the origin. Assume a planar receptor surface described by the plane $z = f$, where f is the *focal-length* of the imaging system. This receptor is placed in front of the nodal point in order to avoid having to reflect coordinates in the equations.

While a planar receptor has been chosen for the development of motion segmentation techniques outlined in this thesis, a spherical receptor surface could also have been used. Such a surface is defined as having a radius f and is centred on the origin. While a spherical receptor would simplify somewhat the analysis of the motion field, it is a less-realistic model than a planar receptor when dealing with images captured by a standard video camera.[1] It should be noted that a spherical receptor could be used as an approximation to a number of cameras mounted together in a fixed relative orientation, with nodal points that coincide. Such an arrangement could be useful when trying to capture images over a wide angular extent while keeping lens distortions to a minimum.

2.2 Perspective Projection

A point in the 3-D world, $\vec{X} \equiv [X_1\ X_2\ X_3]^T$ will image onto a point $\vec{x}$ in the image plane. Under perspective projection the relation between these two points is

$$\vec{x} = \frac{f}{X_3}\vec{X}\ . \tag{2.1}$$

The first two components of $\vec{x}$ describe the position of the point in the image plane. In the following analysis it is useful to continue to think of $\vec{x}$ as a 3×1 vector, even though its third component is the constant f.

In this thesis the case of perspective projection is considered. In most cases perspective projection is a good model for image formation onto a planar receptor surface. Another possibility is orthographic projection, where the relationship between $\vec{X}$ and $\vec{x}$ is given by

$$\vec{x} = [\ X_1 \quad X_2 \quad f\ ]^T\ .$$

Horn [39] points out that, for images of small angular extent where the scene points are far away, orthographic projection is a good approximation of the imaging process.

[1] An example of a spherical receptor would be a retina in a human eye. This receptor does not, however, define an entire sphere. Also, the retina, while not planar, is not perfectly spherical either.

Perspective and orthographic projection carry different information about the world. For example, when trying to recover scene structure from motion, it is necessary that the motion has a translational component if the imaging involves perspective projection, as will be seen from the perspective-projection motion field equations to be presented below. A purely rotational motion will not allow for recovery of scene structure. However, if orthographic projection is used, a purely translational motion of the nodal point gives no information about scene structure: a rotational component is required. Perspective projection is a more general model and will be used throughout this thesis.

2.3 The Motion Field & Optic Flow

In the study of vision, images are seldom static in time. As objects in the scene move, so do their associated images. Analysis of this "image motion" to recover information about scene motion is not a simple task, since information is lost in the perspective projection from a 3-D world onto a 2-D image plane. In this section the relation between scene motion and its associated image motion is studied. Attempts to measure image motion and the difficulties encountered when trying to relate measured image motion to scene motion will then be presented.

2.3.1 The Motion Field

In Figure 2.2 the relationship between the motion of a point $\vec{X}(t)$ in the scene and the motion of its image point, $\vec{x}(t)$ is shown. The velocity of $\vec{X}(t)$ is defined by

$$\vec{V}(t) = \frac{d\vec{X}(t)}{dt} .$$

In order to find the velocity of $\vec{x}(t)$ differentiate

$$\vec{u} = \frac{d\vec{x}}{dt} = \frac{d\vec{x}}{d\vec{X}} \frac{d\vec{X}}{dt} = \frac{d\vec{x}}{d\vec{X}} \vec{V} .$$

It is a simple matter to show that

$$\frac{d\vec{x}}{d\vec{X}} = \frac{f}{X_3}\begin{bmatrix} 1 & 0 & -x_1/f \\ 0 & 1 & -x_2/f \\ 0 & 0 & 0 \end{bmatrix} = \frac{f}{X_3}R(\vec{x}) \ .$$

Here $\vec{u} = R(\vec{x})\vec{V}$ is called *the motion field* [39, 24, 86, 44] of point $\vec{X}$. It is worthwhile at this time to introduce the concept of *depth-scaled* projected velocity $\vec{v}$, defined by

$$\vec{v} = P(\vec{x})\vec{V} \ ,$$

where $P(\vec{x}) = I - \vec{x}\vec{x}^T/\|\vec{x}\|^2$ is a projection onto the plane perpendicular to $\vec{x}$. It can be shown that $\vec{u} = R(\vec{x})\vec{v}$ and $\vec{v} = P(\vec{x})\vec{u}$. This will be useful in Chapter 6 in the discussion on depth recovery. Figure 2.2 shows the relative geometry of $\vec{u}$ and $\vec{V}$. The motion field vector $\vec{u}$ and the depth-scaled projected velocity $\vec{v}$ are shown in the inset of Figure 2.2.

A general form for the velocity of a scene point is $\vec{V} = \vec{T} + \vec{\Omega} \times \vec{X}$ where $\vec{\Omega}$ is a rotation about an axis passing through the origin. The equation for the motion field becomes

$$\vec{u}(\vec{x}) = \begin{bmatrix} 1 & 0 & -x_1/f \\ 0 & 1 & -x_2/f \\ 0 & 0 & 0 \end{bmatrix} \left(\frac{f}{X_3}\vec{T} + \vec{\Omega} \times \vec{x} \right) \ . \tag{2.2}$$

There is a translational component

$$\vec{u}_T(\vec{x}) = R(\vec{x})\frac{f}{X_3}\vec{T}$$

and a rotational component

$$\vec{u}_\Omega(\vec{x}) = R(\vec{x}) \left(\vec{\Omega} \times \vec{x} \right) \ .$$

Only the translational component depends on the depth of the scene point. Longuet-Higgins [58] noted this structure, and showed that it was possible to recover $\vec{T}$, $\vec{\Omega}$ and

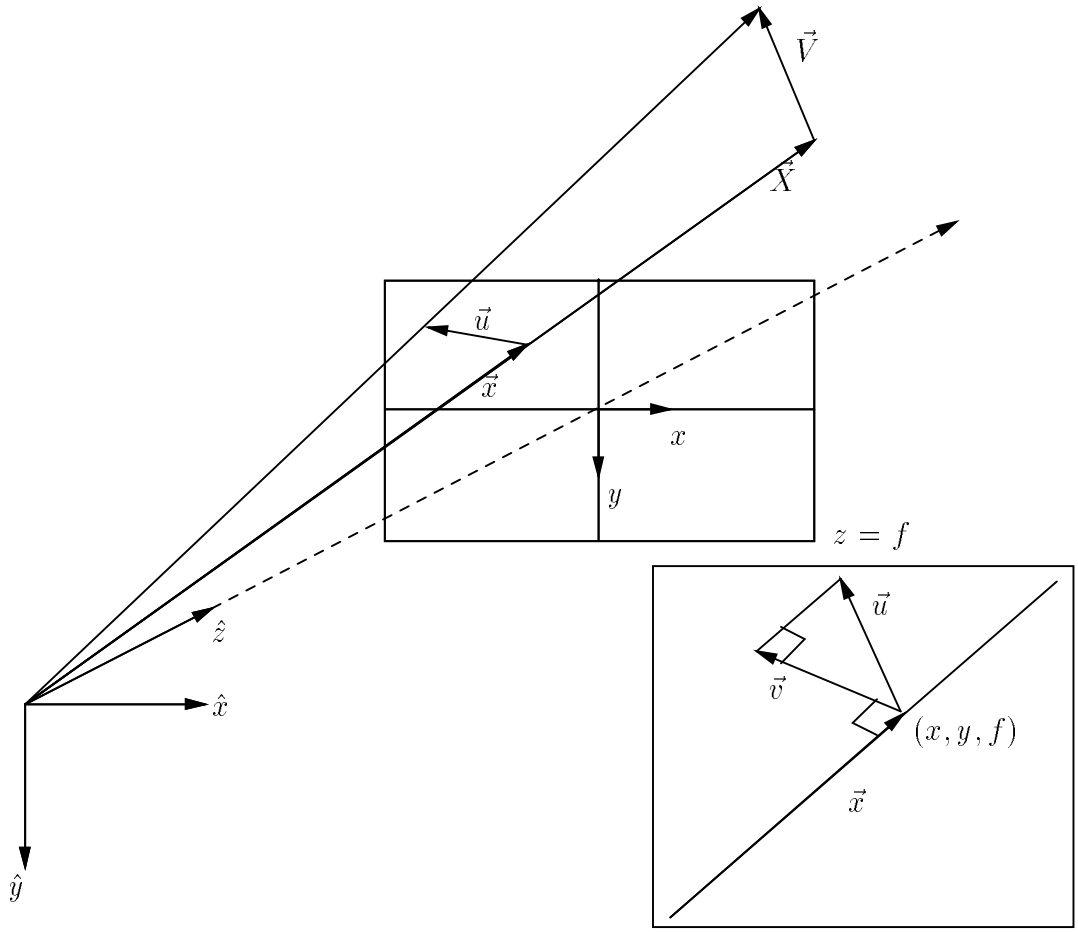


Figure 2.2: The *motion field* is defined as the velocity of image point $\vec{x}$ when its associated scene point $\vec{X}$ is moving with a velocity $\vec{V}$. Label the velocity of the image point (in the image plane) $\vec{u}(\vec{x})$. The motion field vector $\vec{u}$ lies in the image plane. *Inset:* The relationship between $\vec{x}$, $\vec{u}$ (the motion field) and $\vec{v}$ (depth-scaled projected velocity) is shown.

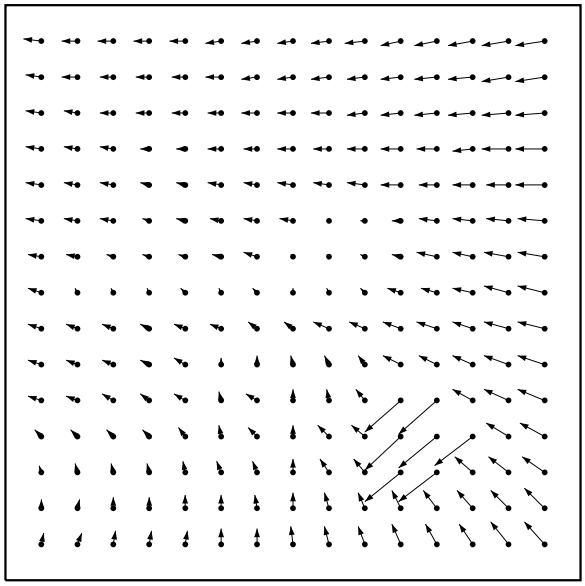


Figure 2.3: A graphical representation of a dense motion field calculated from a computer-generated image. The motion has both a translational and rotational component. An independently moving object can be seen in the lower-right corner.

depth-structure ($1/X_3$). The recovered depth structure is relative, *i.e.* only recovered up to a scale factor, since one can only recover the direction, and not magnitude, of $\vec{T}$. Motion parallax, the apparent relative motion of points in the same visual direction but at different depths, is necessary to recover the *focus-of-expansion* (FOE). Once the focus-of-expansion is known, it is possible to recover $\vec{\Omega}$ and then image depth structure. Gibson was probably the first to experiment with depth recovery from motion parallax [30, 29]. A number of other researchers [50, 39, 57] have considered the recovery of depth-structure from motion, as will be shown in Chapter 4.

The motion field is a purely geometric concept [39]. It is a useful tool for developing the theory of image motion: It is exact[2] and it assigns a velocity vector to each point in the image [39, 86]. An example of a dense motion field (calculated from a computer generated image) is shown in Figure 2.3.

As an example of how the concept of a motion field may be used, Nelson [68] presents an interesting analysis of the motion field as imaged onto a spherical receptor surface. He describes in qualitative terms the form of the motion field generated

[2]Assuming that our projection equations are in fact correct—in reality the image generated by most imaging systems will deviate from these equations through "distortion" effects.

when $\vec{V}$ is a pure translation or a pure rotation. In considering the problem of recovering $\vec{V}$ from a given motion field, he notes that while the motion field can be locally ambiguous[3] as to whether rotation or translation is the underlying motion, consideration of the global motion field, *i.e.* the motion field over the entire sphere, removes this ambiguity.

2.3.2 Optic Flow

While optic flow is not the focus of this thesis, methods of 3-D motion recovery often start with the assumption that either optic flow or point correspondences are available. Also, many of the problems of integration of constraints to achieve good optic flow estimates are similar to those of 3-D motion recovery. Therefore, it is necessary to be aware of the issues surrounding the measurement and use of optic flow. The term *optic flow* was coined by James Gibson.[4] Horn describes optic flow as "the apparent motion of brightness patterns" in an image [39, 86]. This motion is encoded as a field of 2-D vectors, each of which indicates the 'velocity' of the associated image position. In this respect the optic flow field is similar to the motion field. While the concept of optic flow is intuitive and commonly accepted and understood by computer vision researchers, there does not seem to exist any commonly accepted definition.[5] For the purposes of the present discussion, Horn's description should serve us well, despite its lack of mathematical formality.

Ideally, the optic flow corresponds to the motion field, but this is not always the case [39, 86, 24]. For example, consider a sphere with a uniform (unmarked) surface rotating [39, 24]. The image of the sphere does not change, leading to a zero optic flow, whereas the motion field is obviously non-zero. Assume now that the sphere is still, but a light source behind the observer is moving. The shading and specular reflections induced by the light source on the surface of the sphere will move, causing a

[3] This ambiguity arises more because of error and limited precision in measuring the motion field than in the motion field itself.

[4] This is according to Horn [39].

[5] Often individual researchers have definitions which are tied to the methods they use to estimate image motion.

change in the image's brightness patterns even though the motion field is clearly zero. Fleet [24] observes that image brightness is a function of several variables, of which scene structure and camera motion are the ones that interest us. Varying position and intensity of light sources, shadows, and reflectance properties of materials in the scene also play a role. Verri states that "motion field and optical flow are exactly the same only for Lambertian[6] objects which translate under uniform, fixed illumination" [86]. This statement is true if one neglects the possibility of cast shadows which may also move. Verri points out that the assumption that viewing surfaces are Lambertian is not sufficient for the optic flow and the motion field to be identical, as in the case of the rotating sphere. It is usually sufficient, however, that there is texture[7] on the viewed surface [39, 86].

Verri suggests a method by which an optic flow field can be compared with the corresponding motion field: "an optical flow field can be thought of as *close* to the true motion field, if the topological description of the two vector fields (in terms of the theory of dynamical systems) is the same at any fixed time."[86] Specifically, one would compare the number, *kind*[8], and positions of the singular points of the two vector fields. If the first two quantities are the same, and the positions are close, then the vector fields can be called similar. This approach is similar to one by Koenderink and van Doorn [50] in which differential invariants of the optic flow field are determined, and used to hypothesize about the underlying motion field. It should be noted that this approach runs into severe difficulties at places where the fields are discontinuous [86], as it often is due to the presence of occlusion boundaries.

2.3.3 Differential Methods for Estimating Optic Flow

In this thesis it is typically assumed that there is a correspondence between the motion field and optic flow. The methods used to recover optic flow described in Chapter 9

[6] "An *ideal Lambertian surface* is one that appears equally bright from all viewing directions and reflects all incident light, absorbing none." [39]

[7] Texture could be defined, for our purposes, as a large variation in contrast and/or colour over a surface relative to shading variations.

[8] By *kind* one refers to nodes (stable and unstable), saddle points, vortices, *etc.*

are quite robust [40, 43]. The question then arises "How does one measure optic flow?" Fleet breaks down the measurement of optic flow into three steps:

1. *Prefiltering of images.* This removes unwanted noise and prepares for the measurement step.

2. *Extraction of constraints (measurement).* Constraints on the optic flow are generated, and may be used to estimate the flow field.

3. *Integration of constraints.* The constraints from the previous step are combined with assumptions about the world in order to achieve estimates of the optic flow field.

It is possible to broadly categorize techniques for optic flow into *correlation methods* and *differential methods*. While correlation methods will not be discussed here, they do deserve comment. These methods are based on the idea of comparing a region in one image with neighbouring regions in the subsequent image(s). The indication of how well two regions match is usually based on a correlation measure, hence the name "correlation methods". These methods do not easily lend themselves to clustering as they do not generate local constraints which can be used to estimate the flow field. The choice of size of image regions used in matching is not simple, as any image region may contain more than a single motion, or contain insufficient structure to unambiguously determine image velocity. Also, while these methods provide an estimate of local image velocity, they do so by searching for the best estimate instead of explicitly calculating it. They do not show how to improve this estimate as is the case in a method such as gradient-descent minimization. In the remainder of this section aspects of the differential methods are considered. The second and third steps above are exclusive to the domain of differential methods for estimating optic flow, and they will now be considered in more detail.

2.3.4 Generation of Constraints on Optic Flow

If the "apparent motion of brightness patterns" really is a good definition for optic flow it should be possible to formulate a method for measuring flow based on the

intensity in an image. Assume that $\vec{x}(t)$ is the image of a scene point $\vec{X}(t)$ and that $I(\vec{x}(t), t)$ is a scalar-valued function that represents the "brightness" at image location $\vec{x}$ at time t. Further assume that the image of this point has a constant brightness as it moves, *i.e.* $I(\vec{x}(t), t) =$ constant. Determining the image motion at $\vec{x}(t)$ can be formulated as the problem of tracking image points on the basis that their intensity is constant. Horn [39] suggested tracking contours of constant intensity via the *brightness constancy constraint*,

$$\nabla_{\vec{x}} I(\vec{x}, t)^T \vec{u}(\vec{x}) + \frac{\partial I}{\partial t}(\vec{x}, t) = 0 \ , \tag{2.3}$$

which is derived by differentiating $I(\vec{x}(t), t) =$ constant and defining $\vec{u} = d\vec{x}/dt$. This constraint assumes that the only reason that $I(\vec{x}(t), t)$ changes is due to motion in the image. Although this constraint is often violated in practice [7, 43, 39, 66] it is nonetheless a useful start to measuring optic flow.

For a given image location $\vec{x}$, note that Eqn. 2.3 provides a single linear constraint on image motion at that point. An important feature of this equation is that one can only recover the component of the image velocity that lies in the direction of the spatial gradient. If the local image intensity structure is due, say, to the contrast at an edge, then $\nabla_{\vec{x}} I(\vec{x}(t), t)$ will point in the direction perpendicular to the edge. The recovered component of the image velocity is named a *component velocity*.

It is not, therefore, possible to make a local measurement of image velocity. At best it is possible to constrain the image velocity to lie along a line in image velocity space, as in Figure 2.4: this is referred to as the *aperture problem*. If one calculates a constraint at a nearby image point $\vec{x}_2$ the intensity gradient may be in a different direction, thus proving a second constraint line on the image velocity. One expects $\vec{u}$ to lie at the intersection of these two constraint lines. With a number of such constraints it should be possible to compute the image velocity. This will be discussed further in the next section. A modification to Eqn. 2.3 that allows brightness to change in a linear fashion has been proposed [66], although it is shown that it constrains image velocity in an even weaker fashion.

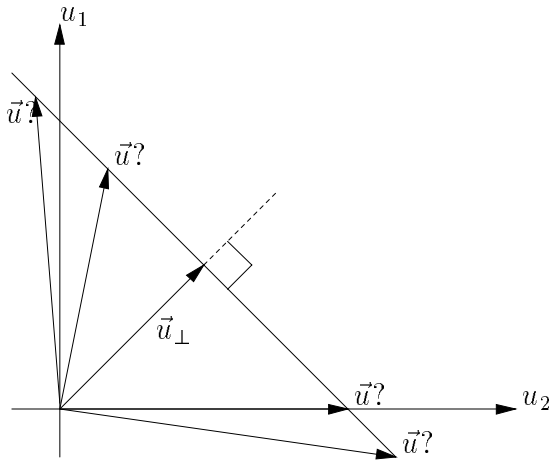


Figure 2.4: Image velocity is constrained to lie along a line, since only the component of image velocity in the direction of the spatial intensity gradient can be measured. Here $\vec{u}_\perp$ is the measured component velocity, and (u_1, u_2) are the coordinates in velocity-space. Any of the $\vec{u}$'s shown here are possible.

While the brightness constancy constraint is widely used, it is not the only method for computing constraints on the flow. Fleet [24] and Fleet & Jepson [25] suggested the use of a *phase constraint* for the recovery of component velocity. During the prefiltering stage it is possible to use filters that are orientation specific and have complex responses to the input images. This response can be written as

$$R(\vec{x}, t) = \rho(\vec{x}, t) e^{i\varphi(\vec{x}, t)}$$

where $\varphi(\vec{x}, t)$ is the phase response of the filter. By differentiating $\varphi(\vec{x}, t) =$ constant a constraint is derived:

$$\nabla_{\vec{x}} \varphi(\vec{x}, t)^T \vec{u}(\vec{x}) + \frac{\partial \varphi}{\partial t}(\vec{x}, t) = 0 \ . \tag{2.4}$$

This equation has the same form as Eqn. 2.3; it tracks contours of constant phase. The aperture problem is also apparent here. It is argued that phase information is more stable than intensity with respect to illumination changes, contrast variation and geometric deformations encountered with perspective projection [24, 25]. An analysis in frequency space is used to support this claim. Furthermore, it is possible

to determine when the phase information is unreliable [26] by using a method of detecting phase singularities, giving an estimate of confidence in the data. The phase gradient $\nabla_{\vec{x}}\varphi(\vec{x},t)$ can be measured given the filter response and the gradient of the filter response. In a comparison of methods, the phase-based flow [25] has been found to be more accurate than comparable intensity based methods [5]. It is also possible to use scale- and orientation-specific filters with the intensity based approach [89, 35]. This has the advantage that several constraints can be recovered at a single image point. Heeger [35] and Fleet [24] note that, in the frequency domain, the power spectrum of a translating image pattern is given by a plane in frequency space. This plane passes through the origin and its orientation is uniquely determined by the image velocity in the time domain. The use of a set of scale- and orientation-specific filters allows for the determination of the tilt of this plane.

It is possible to design filters that are orientation-specific, steerable, and separable [27]. A *steerable* filter is one for which the orientation can be changed using weighted combinations of the outputs of basis filters. *Separable* implies that the 2-D convolution used in generating the filter response can be replaced by a small number of 1-D convolutions. This greatly speeds up the computation of the filter response. These filters can be used for both intensity and phase tracking.

Two types of constraints are shown that can be used to recover image velocity. The information from each constraint is incomplete, as each recovers only the component of the image velocity which is in the direction of the spatial gradient of the quantity being tracked. If it is possible to combine information from several constraints, an estimate of the image velocity can then be made. However, the question of how to integrate constraints is far from simple.

Integration of Constraints

The recovery of optic flow from an image sequence is *ill-posed*[9] [39, 7]. It is, in general, an under-constrained problem, and therefore does not give rise to a unique solution.

[9] A *well-posed* problem is one for which a solution exists, is unique, and varies continuously with the data.

The usual method of dealing with this is to add some assumptions about the flow field in an effort to constrain the solution further, and attempt to choose a flow field that minimizes some cost function that involves the constraints and the extra assumption (regularization). Horn [39] suggests the addition of a requirement that the flow field be smooth. This seems sensible since one would expect that a smooth surface in the scene would give rise to a smooth flow field [49]. The assumption of rigidity will further constrain the solution [39, 49]. However, the smoothness constraint will be violated at occluding boundaries in the scene [39, 49, 7, 9], as can be seen from Eqn. 2.2. Flow discontinuities will also exist at the boundary of an IMO since these boundaries will form occluding boundaries with respect to the background. It therefore becomes necessary to identify points in the image at which the image depth changes discontinuously. If the location of depth discontinuities are known *a priori* it is easy to incorporate this information into the regularization scheme for computing flow. Conversely, if the optic flow is known it is straightforward to identify the points of discontinuity. Since neither is known in advance, it becomes necessary to do *simultaneous* estimation of optic flow and segmentation [39]. The assumption is now that the flow is smooth except for certain, specific locations.

Blake & Zisserman [9] suggest a regularization method including *line-processes* to achieve simultaneous segmentation and flow estimation. Line-processes are a form of penalty system that allows the algorithm to insert discontinuities into the flow field, but at a cost. This cost may be less than trying to enforce the smoothness constraint at a place where a flow discontinuity *should* exist. The concept of a line-process can be generalized under the framework of robust statistics [7, 6]. In general, the cost function in a regularization problem may be highly nonlinear, and as such may have multiple minima. Many methods for optimization of nonlinear functions cannot guarantee convergence to a global minimum. Blake & Zisserman [9] suggest an algorithm which they call GNC (graduated non-convexity) that uses successive approximations to the objective function in order to speed convergence to a minimum. While this minimum is not guaranteed to be global, a better result is typically obtained.

Previously, the aperture problem was mentioned, which limits local measurement

to placing a constraint on the image velocity. If one assumes smoothness of flow then it is possible to integrate constraints over some small region in order to estimate flow in this region. The question is, "How large can we make this region?" Increasing the size of the region constrains the solution better by including more constraints, and it also provides improved noise immunity. However, the larger the region is made the more likely it is that the model used to estimate $\vec{u}(\vec{x})$ will be unable to account for spatial variations in the image velocity, or that the smoothness constraint will be violated. Therefore, there is a trade-off in choosing the size of the region. The problem of choosing the size on this region has been called the *generalized aperture problem* [6].

Robust Statistics and Optic Flow

The application of robust statistics to the problem of optic flow estimation [7, 6, 43] can be considered a generalization of Blake & Zisserman's work. Robust statistics [32] considers the effect of data that is not accurately described by its statistical model. Most data sets either do not conform exactly to a convenient statistical model or may have members that are generated by a process other than the one assumed by the model. In either case, attempting to estimate the parameters of a statistical model may be adversely affected by a small number of data points which are termed *outliers*. *Influence functions* [32, 7] may be used to evaluate the effect of a single data point on parameter estimates. An influence function measures the effect of a single observation on the value produced by an estimator. As an example, it can be shown that a least-mean-squares (LMS) estimator is not robust in the presence of outliers since the effect of a data point is proportional to its distance from the mean. As a result, one outlier far from the estimated mean will have an inordinately large effect on the estimate of the mean. Using influence functions one can devise a number of measures that allow the robustness of estimators to be compared [32]. Three important measures are:

Gross-error sensitivity. This quantity is a measure of the "worst-case" influence that a data observation will have on the estimator. It can be used as an upper bound on the asymptotic bias of the estimator.

Local-shift sensitivity. This quantity is a measure of the effect of minute perturbations in an observation on the estimator. This is important since there are errors introduced whenever a value is observed, either through round-off or measurement error or noise processes. This characteristic can be related to the derivative of the influence function, and can be infinite for influence functions with discontinuities.

Rejection point. This quantity measures the point at which an observation no longer has any effect on the estimator. Beyond this point, the influence function is identically zero. This is a desirable feature in order to reject gross outliers completely.

Hampel [32] suggests that it is possible to devise estimators that possess a low gross error sensitivity, a low local-shift sensitivity, and a finite rejection point. Estimators that limit the influence of outlier data, such as the Lorentzian estimator, have been proposed for regularization problems involving optic flow [7, 6]. The Lorentzian estimator is part of a family of robust estimators called *redescending estimators* that have desirable influence functions. They possess low gross-error sensitivity and have low local-shift sensitivity. While the rejection points of this estimator are at $\pm\infty$, its influence function "redescends" to nearly zero outside a central region. Sawhney *et. al.* [76] note that an infinite rejection point can be desirable in cases where initial estimates for parameters are not well known, but then estimators with redescending influence functions are necessary to still provide some protection from outliers. A plot of the influence function of a Lorentzian estimator can be seen in Figure 2.5.

It should be noted that many methods referred to as "robust" in the vision literature, such as median and trimmed-mean filters, and random sampling techniques for combining constraints, have infinite local-shift sensitivity, since a minute perturbation in a data point might change the value of the median. The term "robust" is used in computer vision literature often without clear reference to which measures of robustness are being applied. The application of robust statistical methods is intended to deal with violations of the brightness constancy constraint as well as the smoothness

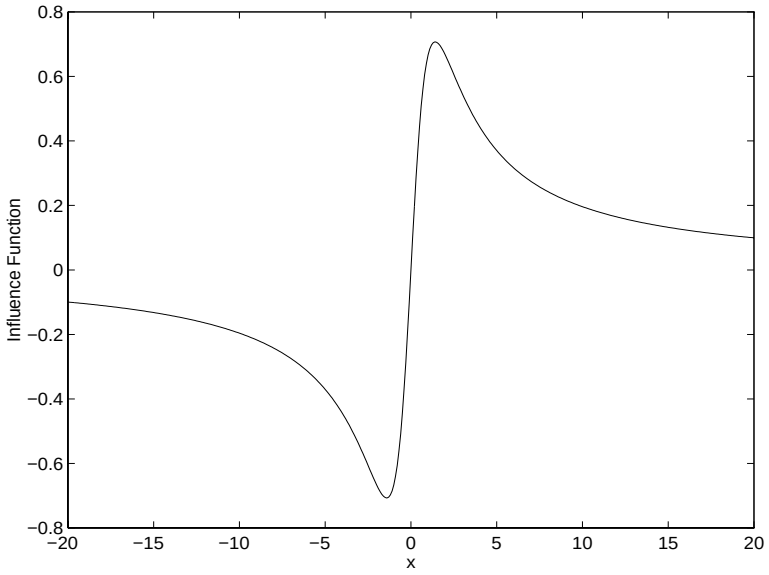


Figure 2.5: The influence function of a Lorentzian estimator is $\psi_\sigma(x) = \frac{2x}{2\sigma^2+x^2}$. The x-axis represents distance of an observation from the current estimate. As data deviates from the estimate its influence increases initially, but as the deviation increases the influence begins to decrease towards zero. This is characteristic of *redescending estimators*. The point at which the estimator begins to redescend is controlled by the scale parameter, σ. In this plot $\sigma = 1$.

constraint.

Mixture models [64] are statistical structures that attempt to model data which may arise from more than a single process. In the case of optic flow, flow in different regions (as defined by the segmentation) could be thought of as coming from different processes. The EM-algorithm [64, 15] allows for estimation of the parameters of these underlying models even though it may not be known in advance which data belong to which process. Component velocities can be integrated using mixture models and the EM-algorithm [40, 43]. Jepson & Black [43] use a mixture-model approach to cluster component velocities, and hence achieve improved optic flow estimates. Their method allows shared ownership of constraints amongst regions. Component velocities are generated using a *data conservation constraint*[10]—each constraint defines a line in $\vec{u}$ space upon which the correct flow will lie. Plotting these constraint lines should reveal multiple intersections, each corresponding to an existing flow within the data. It is

[10]This is a generic constraint which can be implemented by tracking contours of constant phase, intensity, *etc.*

premised that a region may contain multiple motions and that some constraints will be outliers due to violations of the data conservation constraint. The flow is modelled as having multiple layers, with each layer having a model, *e.g.* constant flow within a layer. Constraints are then clustered to layers, and the model parameters for each layer simultaneously estimated. One layer is dedicated to modelling constraints which are outliers. The EM-algorithm is used to achieve a *maximum-likelihood* estimation of parameters and ownership of constraints by layers. Two points are of note here. First, ownership of each constraint may be shared amongst several layers. This is reasonable since it is possible for a constraint line to pass through more than one intersection point. Second, this method allows constraints from non-contiguous image regions to be clustered together. This is important if transparent motion is present, since a region of the image may have more than one velocity. This approach to constraint clustering allows for an improved integration of constraints over methods which rely on line processes to perform the 2-D segmentation. Jepson & Black [43] present results using the constant-flow model, but indicate that the work is easily extended to using affine or rational functions to model flow in a layer. In related work, Jenkin & Jepson [40] use affine and rational functions of image location to estimate the flow. An outlier process was included in these models to account for constraints that could not be fit by any of the underlying processes. Mixture models can be thought of as a form of robust statistics.

Wang & Adelson [87] segment image regions into patches in which optic flow at any point could be modelled as an affine transformation of the image coordinates of that point. The image is arbitrarily broken into small patches, and affine flow parameters are estimated to model the flow in that patch. Patches for which parameter estimation results in a large residual error are rejected as likely containing object boundaries.[11] The remaining patches are grouped on the basis of similar parameters using a K-means approach. The final representation retains information about occlusion and depth ordering.

[11] No details are given as to how the decision of what constitutes a "large" residual error is made.

Ayer *et. al.* [2] propose a method that combines robust estimation of 2-D motion parameters (constant or affine models) within statically segmented sub-regions of an image. The static segmentation is based on pixel intensity of a single frame and used as a starting point for integrating constraints and estimating flow. Parameters are calculated not only by considering regions but also by using information over the entire sequence (regions are tracked by warping them according to the motion parameters). Parameters are estimated using *least median of squares* and *least trimmed squares*. These methods are robust up to a limited percentage of outliers. After motion parameters are estimated, the regions are compared for goodness-of-fit to the model, and regions with good fit are accepted. Those which are not accepted are used as the basis for a new round of motion parameter estimation. This concept of using data which do not fit the current parameters as the basis for motion segmentation is also used in this thesis.

In 2-D motion estimation no distinction is made between flow discontinuities caused by depth discontinuities and by IMOs—this leads to an over-segmentation of the image with respect to 3-D motion. This is characteristic of the 2-D segmentation approaches. Attempts at simultaneous segmentation and estimation of 3-D motion will be presented in Chapter 3. Other approaches to the optic flow problem include application of Kalman filters to carry forward information from one set of flow estimates to the next [78]. The Kalman filter generates covariance estimates for the measurements, and uses these estimates to fuse new measurements with previous ones. The covariance estimates also give a sense of the certainty of each measurement, a concept which is considered important when passing estimates to further processing stages [79]. The concept of improving estimates over time (incremental estimation) has been suggested by other researchers [6, 7]. Incremental estimation is an attempt to exploit temporal coherence in optic flow, as well as distributing the cost of computing flow over a number of frames from the sequence. Instead of trying to exhaustively refine flow estimates for each pair of frames, one attempts to improve the estimates over time, allowing only limited computation for each frame. Since the flow field is not expected to vary drastically between adjacent frames, the results of the previous

frame provide a good starting point for the estimation of the next frame's flow. It should be noted that this attempt to exploit temporal coherence in the flow field does not overcome the obstacles to flow estimation outlined previously.

Flow Segmentation and Independently Moving Objects

Why perform segmentation based on IMOs when the optic flow constrains the necessary segmentation information? The answer is that the segmentation of flow will over-segment the image with respect to IMOs. The flow segmentation will place a discontinuity at every location where the depth to scene points changes abruptly. While this does include IMOs, it also includes a lot more. The question then becomes, "Which parts of the flow segmentation are relevant to segmentation of IMOs?"

Effect of Poor Integration on Quality of Flow

If flow segmentation is poorly done, then an inappropriate combination of the information in the measured image velocity constraints will occur—this leads to noisy or erroneous flow estimates [39, 49, 7]. This, in turn, results in a poor estimate of the flow field, and will introduce errors into constraints on 3-D motion which will be generated later.

2.4 Difference Between 2-D & 3-D Motion

When one refers to 2-D motion, one is talking about the motion field or optic flow. This is a field of 2-D vectors, each at some particular spatial location in the image. Only the relative motion between the camera and a scene point, as well as the distance between them, affects the 2-D motion field. 3-D motion refers to the translation and rotation of a point in the world that induces the motion of its associated image point. It is in this domain that one can be concerned with identifying IMOs, as IMOs have different relative 3-D motion from other parts of the image. Previous work on the recovery of 2-D motion from images has been presented in this chapter. In the next chapter previous work on estimating 3-D motion from images will be presented.

Chapter 3

Previous Work on 3-D Motion Recovery

The recovery of egomotion parameters for an observer moving in a rigid environment is a much-studied problem. It is possible, under the right circumstances, to recover the observer's translational direction, rotation and also to recover information about the depth structure of the scene. The previous chapter described work on 2-D motion estimation, which involves identifying points in the image where depth discontinuities exist. This typically involves operating on optic flow or component velocities. Much work has also been done on the problem of 3-D motion segmentation. Most of this work has been an attempt to perform some form of clustering in a parameter space. In this chapter work done on egomotion and scene structure recovery, as well as work on 3-D motion segmentation, will be reviewed. These approaches are divided as follows:

1. Methods assuming orthographic projection,
2. Methods based on the *essential matrix*,
3. Methods requiring planar patches,
4. Direct methods, and
5. Subspace methods.

While some work does not fall neatly into one of these categories, these divisions highlight important features of previous attempts to solve the egomotion and 3-D motion segmentation problem. A further division regards whether the methods assume optic flow or discrete displacement of tracked features (this division does not apply to most of the direct methods).

The translational and rotational motion of a scene point with respect to the observer, as well as the distance from the observer to the point, all play a part in the motion of the image of the point. A number of methods exist for recovery of the motion components without *a priori* knowledge of the scene's depth structure. This process is often referred to as *egomotion recovery*, based on the premise that image motion is caused by a moving observer in a stationary environment. Other methods attempt simultaneous recovery of egomotion parameters and scene structure. Horn [39] suggests linear methods for recovering motion parameters when it can be assumed that the motion is purely translational or purely rotational. In the event of combined rotation and translation the equations become nonlinear, and their solution involves estimating relative depth.

3.1 Orthographic Methods

Tomasi & Kanade [82], using orthographic projection, demonstrate a method that factors point correspondences between images into their motion and shape components. This method assumes that a set of corresponding points can be identified between image pairs.[1] Assume that W is a $2F \times P$ matrix composed of P 2-D image points tracked over F frames, then it may be factored as

$$W = MS$$

where M is a $2F \times 3$ matrix of the rotational motion between frames, and S is a $3 \times P$ shape matrix containing the coordinates of the P points. In practice the

[1] The problem of determining the appropriate matching of points across different frames is referred to as the *correspondence problem* [85].

tracked image coordinates will be noisy, so a singular-value decomposition method is used to perform the factorization.

Costeira & Kanade [12] build on the work of Tomasi & Kanade [82] to produce a method which identifies IMOs. Starting from feature correspondences, a shape matrix is derived. From this is derived a "shape interaction matrix" which can be put in block-diagonal form, where each block represents a unique motion. No prior knowledge of motion or its segmentation is necessary. However, the method only works with orthographic projection.

Work by Koenderink & van Doorn [51] show that it is possible to recover affine structure from matched image features. Koenderink *et. al.* show that four points matched between two images are sufficient to define an affine basis from which the expected image motion of other image features can be calculated, assuming the images are formed using orthographic projection.

Lawn & Cipolla [54] make use of the affine-structure technique of Koenderink & van Doorn [51]. Their work employs the small-angle approximation, which allows perspective projection to be approximated by orthographic projection, in order to recover egomotion from image sequences. This method relies on looking for point correspondences which violate the rigidity assumption, and therefore indicate the presence of an IMO.

The use of orthographic projection is not always a realistic assumption for most imaging systems in use today. As has been mentioned previously, depth structure can only be recovered from orthographically projected image sequences in the event that a significant rotational component to the motion exists. This is directly opposite to perspective projection, where depth structure is only recoverable when significant translational motion is present. Perspective projection is a more common model for imaging systems.

Fortunately, the notion of *affine structure* as introduced by Koenderink & van Doorn [51] can be extended to the case of perspective projection. Faugeras [22] assumes perspective projection and shows how to construct a projective basis with the use of eight points. The additional points are necessary to compute the translational

direction (once translational direction is recovered, only 4 points are needed to define an affine basis, and 5 for a projective basis). The reconstruction of image "structure" is done without knowledge of the camera's *intrinsic parameters*: *focal length*, *optic centre*[2], and *pixel scaling* in terms of distance-per-pixel in both the horizontal and vertical directions. *Extrinsic parameters* are those related to the change in world coordinate systems between the views, *i.e.* rotation and translation. Separation of the projection matrix into *intrinsic* and *extrinsic* parameters is demonstrated in Luong *et. al.* [59]. This important result allows one to study structure from motion without painstaking camera calibrations. Work presented in this thesis assumes *a priori* knowledge of the camera's intrinsic parameters.

3.2 Essential-matrix Methods

A method for recovering egomotion using an *essential-matrix* has been proposed [57, 90]. Using point correspondences it is possible to form the matrix equation

$$\vec{x}' E \vec{x} = 0 \ , \tag{3.1}$$

where E is the essential-matrix and $\vec{x}'$ and $\vec{x}$ are the corresponding image locations of the same point, written as 3-vectors. The 3-D world coordinates of these points are $\vec{X}'$ and $\vec{X}$ respectively, and are related by $\vec{X}' = R\vec{X} + \vec{T}$ where $\vec{T}$ is a displacement vector and R is a rotation matrix. The matrix E is a function of $\vec{T}$ and R: $E = [T]_\times R$ where

$$[T]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$$

and $\vec{T} = [t_1 \ t_2 \ t_3]^T$. The matrix $[T]_\times$ is the equivalent of taking a cross product with $\vec{T}$, that is $[T]_\times \vec{x} = \vec{T} \times \vec{x}$. Since Equation 3.1 is homogeneous, it is only possible to

[2] In practical terms this is the location in the image of a point which lies on the camera's optical axis. While this is usually assumed to be in the centre of the image, it seldom is in practice.

recover E (and subsequently $\vec{T}$) up to a scale factor. Given a minimum of 8 point correspondences, a set of equations can be solved for the elements of E. The translational direction can be recovered by noting that $EE^T = [T]_\times [T]_\times^T$. The rotation parameters can then be recovered. A detailed study of the properties of essential matrices can be found in [62]. Weng *et. al.* [90] show that this method had significant error for translational directions parallel to the image plane and also that error increased as the angular field-of-view in the image decreased. This is a feature common to other methods, as shall be seen later. The essential-matrix methods are more suited to cases in which the displacement of image points is known. One can think of optic flow as providing information on the instantaneous motion of an image point, and feature tracking as providing discrete information on the movement of image points. While displacement over a time interval can be used as a first-order approximation to velocity, the displacements recovered by estimating optic flow are too small to effectively employ the essential matrix.

Torr [83, 84] suggests a stochastic approach to motion segmentation that makes use of essential-matrices. Starting with feature-correspondences, the pairs of image points are clustered, with each belonging to one (and only one) cluster. Each cluster represents some number of feature-pairs related by a single essential matrix, E. Clusters are created by randomly selecting 7 pairs and calculating E. More pairs are added with an eye to consistency: each new pair is tested to see if it meets a criterion of being at least 95% likely to belong to the cluster, as determined by a *t-test*. Small clusters may be pruned depending on the likelihood that they were randomly generated, and like clusters are merged. A special cluster, which makes use of a uniform probability density function, is used to capture data points which are not well-modelled by the other clusters. Finally, using an *integer programming* technique, the clusters are partitioned to form a segmentation.

This work deserves further comment in that it is the closest in both spirit and form to the methods described in this thesis. An attempt is made, using the methods of robust statistics, to cluster constraints on 3-D motion thereby performing a segmentation based on motion. The number of motion processes is determined by

examining the data for support of new processes. The major differences are that this work deals with optic flow as input data, the segmentation is performed based on constraints that are related solely to translational direction, and mixture models are used to model multiple motion processes.

3.3 Methods Requiring Planar Patches

Adiv [1] identifies regions in the image in which estimated optic flow is consistent with the movement of a planar surface, and groups these regions according to their mutual consistency for various 3-D motions. Each grouping therefore represents an independent motion. Sinclair [77] segments images by recovering the 3-D angular velocity field for the image, and uses a simple clustering algorithm for identifying planes in angular velocity space. This method also requires identifying planar surfaces in the image. Both Sinclair and Adiv's methods require the existence (and identification) of planar surfaces in the image. They both are examples of clustering methods. Work by Darell & Pentland and by Hanna cited in the next section also involves the assumption of planar surfaces in the image sequence.

In many man-made environments planar surfaces occur in abundance, so these methods could be expected to be useful. However, in considering natural environments, planar surfaces may be limited to the ground plane (*e.g.* a smooth field), or may not exist at all! Methods which could exploit the existence of planar surfaces, but do not require them, are potentially more useful.

3.4 Direct Methods

The approaches described in this section are referred to as *direct methods* because they either bypass the stage of estimating optic flow or because they treat the imaging process in a non-standard way. They may generate constraints on 3-D motion as part of a process to recover egomotion or identify IMOs, but they need not necessarily do so.

Darell & Pentland [14] have suggested segmentation and estimation based on a competitive scheme that attempts to assign constraints to different "layers" in a multi-layer model. Their work is formulated for the case of pure translation and assumes that each layer models the motion field of a planar surface, the image sequence has constant brightness and is formed using perspective projection. The constraints are formed from the spatial and temporal derivatives of the intensity image, and obey the relation

$$T_x\,dx + T_y\,dy + T_z(x\,dx + y\,dy) + dt = 0$$

where $\vec{T} = [T_x, T_y, T_z]^T$ is the translational motion of the support region. Each constraint is assigned a level of support for each layer. The method starts with more layers than are expected, and an initial guess for the translational motion of each. Based on the initial estimates, a support map is calculated for each layer, and then the translation estimates are updated using maximum likelihood estimation. This process continues iteratively until parameters converge. The number of layers is determined by excluding layers with too little support—in the optimal situation one layer will remain for each distinct region of motion. The advantage of using support maps over line processes is that layers do not require their supporting regions in the image to be contiguous. Additional morphological constraints are used to discourage support regions consisting of small numbers of contiguous points. This method is quite similar to methods of Jepson & Black [43], and Wang & Adelson [87] (described in the previous chapter), using a planar patch model in place of affine or rational models. The method is similar in flavour to the EM-algorithm, which will be described in a later chapter.

Hanna [33] uses a multi-resolution iterative approach for recovering egomotion and structure from component velocity, assuming local planar surface models in a manner similar to Darell & Pentland. This approach does not limit itself to purely translational motion. The method is also iterative, using first the model of local surface orientation to improve egomotion estimates, then using egomotion estimates to improve local surface models. Unlike that of Darell & Pentland, the method makes

no attempt at robust estimation of parameters. Also, local surface models are fit to small, fixed and overlapping image regions, so the notion of ownership of constraints by a layer does not exist here. This method is similar to those of Durić and Nelson [18, 67] in that it bypasses optic flow and attempts to relate component velocity directly to egomotion.

Nelson [67] describes a method for performing 3-D motion segmentation which could properly be thought of as a 3-D method since it only relies on 3-D motion parameters and measurements which can be made directly from image pairs in the sequence. Given knowledge of the observer motion, that is, assuming that $\vec{T}$ and $\vec{\Omega}$ are known, he compares expected properties of the motion field against measured component velocities, and where significant deviation is found assumes independent object motion. This method is direct because it avoids the need to even compute optic flow. Knowledge of $\vec{\Omega}$ uniquely defines the expected rotational component of the motion field. Knowledge of $\vec{T}$ constrains the translational component of the motion field to lie along lines of longitude on the unit sphere, all of which pass through the direction of translation. This method has the drawback of requiring *a priori* knowledge of the observer motion, and as such can not be considered to be a motion-estimation method. Also, no attempt is made to distinguish between different IMOs. Given the difficulties associated with the estimation of optic flow, bypassing this step is of considerable value. Heeger & Hager [36] proposed a method which, while not a direct method[3], is worth mentioning here because of its relationship to Nelson's method. They consider the problem of integrating information from several sensors. Specifically, in addition to considering information from an image sequence, they also have measurements from a set of positional and inertial sensors regarding the motion of the observer. This information may not be completely reliable and accurate, but it may provide a good initial estimate for the egomotion. As they point out, a good initial estimate for egomotion may greatly simplify identification of IMOs. Assuming that $\vec{T}$ and $\vec{\Omega}$ are known, then optic flow must lie along a line segment where different points on the line correspond to different values of $X_3(\vec{x})$, *i.e.*

[3]Their method uses pre-computed optic flow, and therefore is not direct.

different depths of the scene points in the real world. Measured flow that deviates from this line may indicate the presence of an IMO. Their model attempts to combine information from the multiple sources through the use of a threshold based on the *Mahalanobis distance*, and generate improved estimates for image velocity while at the same time detecting IMOs. This method can miss an IMO if it has a flow pattern that does not deviate from that caused by the egomotion (*cf.* Section 4.5) but, given accurate optical flow, it will not generate false positives.

Azarbayejani *et. al.* [3] use an extended Kalman filter to recursively estimate structure and egomotion over an image sequence. The input data for the filter are tracked image points, making this a feature-based method. The state vector encodes the motion parameters ($\vec{T}$ and $\vec{\Omega}$) as well as one structure (depth) parameter for each tracked point, and the measurement model describes the nonlinear nature of the perspective projection. This methodology also incorporates a measure of the certainty of each estimated state parameter through the updates of the covariance matrix. This method does not bypass the need for displacement estimates, and makes no attempt to deal with outliers in the measurement data. It is of interest, though, because it encodes information about the imaging process in the measurement equation, and information about temporal coherence in the state transition matrix.

When either rotation or translation dominate egomotion, it is possible to simplify the recovery of the 3-D motion parameters. Durić *et. al.* [18] use the *Frenet-Serret* motion model to derive a measure regarding the dominance of either type of motion. Intersecting constraints based on recovered component velocity are used to estimate the focus-of-expansion for translation or the axis of rotation.

3.5 Subspace Methods

While a complete description of subspace methods will be given in the next chapter, it is worth mentioning other methods that can be thought of as being within the domain of subspace methods. A method by da Vitoria Lobo [13] is based on the observation that it is possible to cancel the effect of rotation using three collinear

image points. He proposes an operator which, at a given point in the image, performs a summing operation along lines in multiple orientations passing through the point. The sum takes advantage of the cancelation effect and therefore only depends on the translational component of the image motion. It is shown that such a sum performed along a line which contains the FOE evaluates to zero. Therefore, if the operator is scanned over the image, it should have a minimum response at the location of the FOE. This also points to a method for detecting IMOs. Once the FOE is located, collinear triples of image points (each of which is chosen so that its line passes through the FOE) are tested by checking their sums and comparing to zero: triples with sums above a preset threshold are probably due to IMOs. In a noise-free image the threshold is unnecessary as the sums are expected to be exactly zero. It should be noted that image motion due to IMOs is a form of "noise" for this operator. It is reported to be robust to small patches of IMO image motion, but breaks down after a point. Jepson & Heeger [46] point out that the method of da Vitoria Lobo can be thought of as a special case of subspace methods. Sampling geometries that are collinear and lie on the same line as the FOE only require three flow samples to generate a constraint on the translational motion (*cf.* Section 4.2.2).

3.6 Limitations

The preceding sections have discussed: methods which involve orthographic projection; methods which require that images contain depth structure which is, at least locally, planar; methods based on the essential-matrix; and some methods which attempt to bypass the generation of 2-D motion estimates. These methods are used to estimate 3-D motion parameters or identify IMOs or both.

The approaches taken previously to 3-D motion segmentation are often considered robust, yet many would fail the tests set out by Hampel [32] with regards to *local-shift sensitivity*. Some methods require that planar surfaces exist in the image, or that individual features be tracked. Methods requiring orthographic projection are unrealistic in real situations since most cameras employ lenses that have too large

a field of view to be approximated as orthographic. This thesis pursues methods based on improved optic flow estimates [43, 40] and on segmentation of subspace constraints on 3-D motion [44, 45, 46] using the EM-algorithm as a basis for robust parameter estimation and clustering. The EM-algorithm, when used with a mixture of smooth distributions, has low local-shift sensitivity and has the advantage that it is guaranteed to improve the likelihood function while simultaneously performing clustering and parameter estimation.

Chapter 4

Generation of 3-D Motion Constraints Using Subspace Methods

In the preceding chapters current work on image segmentation based on 2-D and 3-D motion were discussed. This chapter looks at constraints on 3-D translation and rotation which allow rotation and translation to be separated from scene structure (and each other) in order to simplify the parameter estimation problem.

4.1 Motion Constraints

Given a measurement of optic flow for an image, it is possible to generate constraints on the 3-D relative motion underlying the motion field. The relative motion of the observer with respect to the world can be described by a translation, $\vec{T}$, and a rotation, $\vec{\Omega}$. This rotation is about an axis which passes through the nodal point of the imaging system. In our defined coordinate system the nodal point is at the origin. Consider the relative motion of a point in 3-D space, $\vec{X} = (X_1, X_2, X_3)^T$, where X_3 lies along the optical axis (Z-axis). The image of this point under perspective projection is $\vec{x} = (x_1, x_2, f) = \frac{f}{X_3}\vec{X}$ where f is the focal length of the imaging system. The motion field under perspective projection at this image point is given by Eqn. 2.2,

remembering that $X_3(\vec{x}) = X_3$ is the projection of $\vec{X}$ onto the optical axis. The flow field can be thought of as having two components—a translational component and a rotational component. As discussed in Chapter 2, $\vec{u}(\vec{x}) = \vec{u}_T(\vec{x}) + \vec{u}_\Omega(\vec{x})$ where $\vec{u}_T$ depends only on the translational motion and $\vec{u}_\Omega$ depends only on the rotational motion. Note that only the translational component is affected by the distance to points in the image. Therefore, any discontinuities in the motion field must be due to variations in depth, a fact exploited by Rieger & Lawton [74] in their method for recovering translational motion. Consider two points $\vec{X}_1$ and $\vec{X}_2$ such that

$$\frac{1}{X_3(\vec{x}_1)}\vec{X}_1 = \frac{1}{X_3(\vec{x}_2)}\vec{X}_2 \ .$$

Both points will have the same image location. Assuming that both of these points have the same motion $(\vec{T}, \vec{\Omega})$ relative to the observer, then the only difference in their flow components will be due to non-zero translation. Specifically,

$$\vec{u}(\vec{x}_1) - \vec{u}(\vec{x}_2) = \begin{bmatrix} 1 & 0 & -x_1/f \\ 0 & 1 & -x_2/f \end{bmatrix} \left(\frac{1}{X_3(\vec{x}_1)} - \frac{1}{X_3(\vec{x}_2)} \right) f\vec{T} \ .$$

This difference vector will pass through the FOE assuming that the FOE lies in the image plane. One such difference vector is not sufficient to recover the FOE, but two non-collinear ones are. In practice, the least squares minimum is used to determine the intersection of many difference vectors. Rieger & Lawton [74] approximated this vector by differencing the flow of nearby points and recovering the direction of translation. The dependence of the motion field on depth is also exploited by the subspace methods. These variations in depth can be classified into two types, depending on whether the depth variation is because of a boundary formed by an IMO or an object that is stationary in the environment.

4.2 Subspace Methods

It is possible to derive constraints on 3-D motion from the motion field (and hence from optic flow). Here a technique developed by Jepson and Heeger [45] called "subspace methods" is described. This technique allows the derivation of bilinear and linear constraints on 3-D motion, and takes its name from the use of a vector subspace to derive the linear constraints. Through appropriate integration of these derived constraints it is possible to recover the relative 3-D motion that gave rise to an observed optic flow field.

4.2.1 Bilinear Constraints

A simple algebraic manipulation of Eqn. 2.2 [44] allows us to derive the following *bilinear* constraint on $\vec{T}$ and $\vec{\Omega}$.

$$\vec{T}^T(\vec{x} \times \vec{u}(\vec{x})) + (\vec{T} \times \vec{x})^T(\vec{x} \times \vec{\Omega}) = 0 \ . \tag{4.1}$$

This is an exact constraint on the motion field, although it is nonlinear in its motion parameters. While it is nonlinear, it has a special form: If $\vec{T}$ is held constant then the equation is linear in $\vec{\Omega}$ and *vice versa*. It should be noted that only a single flow vector (and its image location) are required to define each constraint. The constraint is also independent of the depth of the point imaged at $\vec{x}$. Eqn. 4.1 can also be written as

$$\vec{T}^T(\vec{a} + B\vec{\Omega}) = 0$$

where $\vec{a}$ is a 3×1 vector-valued function of $\vec{x}$ and $\vec{u}(\vec{x})$, and B is a 3×3 matrix-valued function of $\vec{x}$. It is possible to derive a linear constraint on $\vec{T}$ from a set of bilinear constraints.

4.2.2 Linear Constraints

A linear constraint on $\vec{T}$ can be derived from 7 or more bilinear constraints of the form of Eqn. 4.1 [45]. Given optic flow sampled at K discrete points in the image,

$\{\vec{x}_k\}_{k=1}^K$, construct a constraint vector

$$w_i\vec{\tau}_i = \sum_{k=1}^{K} c_{ik}\left[\vec{u}(\vec{x}_k) \times \vec{x}_k\right] \tag{4.2}$$

where $\vec{\tau}_i$ is a unit vector, and w_i is the norm of the right-hand side of the expression. Through suitable choice of the $\vec{c}_i$ one can guarantee that the constraints $\{\vec{\tau}_i\}_{i=1}^N$ will be orthogonal to $\vec{T}$, *i.e.* $\vec{T}^T\vec{\tau}_i = 0$, $i = 1 \ldots N$. From Eqn. 4.1 it is seen that a sufficient condition on the $\vec{c}_i$ is that they are orthogonal to all quadratic forms involving $[\vec{x}_k]_1$ and $[\vec{x}_k]_2$ over the sample points. (Here $\vec{x}_k = [[\vec{x}_k]_1, [\vec{x}_k]_2, [\vec{x}_k]_3]^T$.) Specifically,

$$\begin{bmatrix} 1 & & 1 \\ [\vec{x}_1]_1 & & [\vec{x}_K]_1 \\ [\vec{x}_1]_2 & \cdots & [\vec{x}_K]_2 \\ [\vec{x}_1]_1^2 & & [\vec{x}_K]_1^2 \\ [\vec{x}_1]_1[\vec{x}_1]_2 & & [\vec{x}_K]_1[\vec{x}_K]_2 \\ [\vec{x}_1]_2^2 & & [\vec{x}_K]_2^2 \end{bmatrix} \vec{c}_i = \vec{0}\ .$$

The coefficients of $\vec{c}_i$ effectively annihilate the contribution due to $\vec{\Omega}$, and they only depend on the sampling geometry, not the flow measurements. It is possible to generate $K - 6$ linearly-independent constraint vectors for each set of K sampled points. These will form a basis for a subspace of $\mathcal{R}^K$.

Since the $\vec{c}_i$ are orthogonal to all quadratics in image location over the sampling geometry (as shown above), the technique requires a variation in depth that is not planar over the image region from which the optic flow is sampled in order to create a non-zero constraint. The practical importance of this is that no constraint can be generated if all the flow samples come from a single planar surface.

The subspace methods allow for generation of bilinear constraints on $\vec{T}$ and $\vec{\Omega}$, and linear constraints on $\vec{T}$. These constraints require measurement of optic flow over some sampling geometry, as well as a set of coefficients which are specific to the sampling geometry. These constraints allow for estimates of the motion parameters $\vec{T}$ and $\vec{\Omega}$ to be made.

4.2.3 Geometric Interpretation and Solution of Linear Constraints

The linear constraints on translation can be used to recover the direction of translation, but not its magnitude. This is a feature inherent in the problem itself [39], not the subspace methods. This can be seen by considering what happens when both $X_3(\vec{x})$ and $\vec{T}$ in Eqn. 2.2 are multiplied by the same scale factor—the resulting value of $\vec{u}(\vec{x})$ does not change. Given a set of weighted linear constraints $\{w_i\vec{\tau}_i\}_{i=1}^N$ the solution for $\vec{T}$ can be found by determining the eigenvector corresponding to the smallest eigenvalue of

$$D = \sum_{i=1}^{N} w_i^2 \vec{\tau}_i \vec{\tau}_i^T \ . \tag{4.3}$$

This is the equivalent of minimizing the expression

$$E(\vec{T}) = \vec{T}^T D \vec{T} = \sum_{i=1}^{N} w_i^2 \left(\vec{T}^T \vec{\tau}_i\right)^2 \ .$$

The computation is a simple least-squares problem and is easily performed with standard algorithms. It is worthwhile to consider a geometric interpretation of the constraints. In the event that one eigenvalue of D is significantly smaller than the other two, then the constraint vectors lie close to a great circle on the unit sphere, and the correct translational direction is the vector normal to the plane defined by this great circle.[1]

It is generally the case that the constraint vectors will not be uniformly distributed around this great circle. For example, if the sampling geometry is small, then the $\vec{\tau}_i$'s are very nearly orthogonal to the mean sampling direction

$$\vec{x}^i = \frac{1}{K} \sum_{k=1}^{K} c_{ik}^2 \vec{x}_k \ .$$

Since the angular extent of the image is limited, then so will be the extent of the constraints along the great circle. As the angular extent of the image is decreased

[1] In the absence of noise one would expect the constraint vectors to lie on this great circle, not just near it.

and noise is added, the constraints will cluster into a small region which will no longer clearly define a great circle [46]. In this case, there will be two eigenvalues of D with roughly the same magnitude and a third, larger eigenvalue. It will no longer be possible to assign a unique translational direction. Constraints of this nature could be thought of as providing a single linear constraint on the translational motion.

This latter case is important in the event of independent object motion. Since many objects will have a small angular extent with respect to the image, the constraints generated by these objects will not necessarily define a unique translation for the object. For objects that are small, it may also be the case that the constraints will mostly be generated from optic flow samples that lie across an IMO boundary, and, as will be explained below, this violates an underlying assumption of the subspace methods. First, consider the effects of noise on the linear constraints.

4.2.4 Effects of Noise on Linear Constraint Vectors

In the absence of noise, the constraints generated by the subspace methods are exact [45]. Since the measurement of optic flow is far from noiseless it becomes necessary to consider the effects of noise on the method. Jepson & Heeger [46] report a bias in the estimates of $\vec{T}$ when using optic flow with isotropic noise to generate linear constraints. This bias exists because isotropic noise in the flow leads to non-isotropic noise in the $\vec{\tau}_i$.

To see how the bias develops, consider noisy constraint vectors $\tilde{\tau} = \vec{\tau} + \vec{n}$ where $\mathrm{E}\{\vec{n}\} = \vec{0}$, and

$$\mathrm{E}\{\vec{n}\vec{n}^T\} = \sigma^2 \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ -x & -y & x^2+y^2 \end{bmatrix}.$$

This covariance matrix for the noise vectors may be derived assuming isotropic, zero-mean noise in the optic flow. When the D matrix is constructed, it is seen that

$$\tilde{D} = \sum_{i=1}^{N} w_i^2 \tilde{\tau}_i \tilde{\tau}_i^T$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} w_i^2 \left(\vec{\tau}_i \vec{\tau}_i^T + \vec{\tau}_i \vec{n}_i^T + \vec{n}_i \vec{\tau}_i^T + \vec{n}_i \vec{n}_i^T \right) \\
\mathrm{E}\{\tilde{D}\} &= D + \mathrm{E}\left\{ \sum_{i=1}^{N} w_i^2 \vec{n}_i \vec{n}_i^T \right\} \\
&= D + \sigma^2 M \\
M &= \sum_{i=1}^{N} w_i^2 \begin{bmatrix} 1 & 0 & -x_i \\ 0 & 1 & -y_i \\ -x_i & -y_i & x_i^2 + y_i^2 \end{bmatrix} . \qquad (4.4)
\end{aligned}
$$

The noise adds a term to the expected value of our $\tilde{D}$ matrix which will, in general, affect the eigenvectors of $\tilde{D}$ and therefore our estimate of $\vec{T}$.

Jepson & Heeger [46] suggest a dithering method to make the noise in the $\vec{\tau}_i$ isotropic, thereby removing the bias. In this thesis, a different approach is taken. Since the form of the noise covariance for the $\vec{\tau}_i$ can be derived, it is possible to perform a re-scaling of the $\vec{\tau}_i$ into a space where the noise is isotropic, make the required estimation of $\vec{T}$, and convert the results back to the original space. To see how this works, first note that adding a scaled version of the identity matrix to D does not alter the eigenvectors, *i.e.* if $\tilde{D} = D + \sigma^2 I_3$, then $D\vec{x} = \lambda\vec{x} \Rightarrow \tilde{D}\vec{x} = (\lambda + \sigma^2)\vec{x}$. Note that the eigenvectors of the two matrices are identical, and the ordering of the eigenvalues is preserved. One can achieve re-scaling by pre- and post-multiplying $\tilde{D}$ by the inverse square-root of the covariance matrix M. This gives us $M^{-1/2} D M^{-1/2} + \sigma^2 I_3$, which will have the same eigenvectors as $\hat{D} = M^{-1/2} D M^{-1/2}$. This operation is also referred to as *pre-whitening* [34]. Choose the eigenvector $\vec{x}$ that corresponds to the minimum eigenvalue of $M^{-1/2} D M^{-1/2}$, namely $M^{-1/2} D M^{-1/2} \vec{x} = \lambda\vec{x}$. The new estimate for the translational direction is $\vec{T} = M^{-1/2}\vec{x}$.

Observe that $M^{-1} D (M^{-1/2}\vec{x}) = \lambda M^{-1/2}\vec{x}$. The estimate for $\vec{T}$ is an eigenvector of $M^{-1}D$, not D. However, since D represents the noise-free constraints, its minimum eigenvalue is 0. This guarantees that pre-multiplying D by M^{-1} will not change the corresponding eigenvector. Therefore the estimate $\vec{T}$ corresponds to the noise-free estimate, and the bias has been removed. It can be shown that this is also the maximum-likelihood (ML) estimate. In practice it will not be possible to completely

remove the bias, as the estimate for the form of M will not be perfect. In Chapter 8 the method of rescaling is applied to the case of multiple object motion using ownership probabilities calculated with the EM-algorithm.

4.3 Relation of Subspace Methods to Rieger & Lawton's Method

It is worthwhile here to point out the relation between the subspace methods and work done by Rieger and Lawton [74] with respect to determining translational direction, $\vec{T}$. Two points $\vec{x}_1$ and $\vec{x}_2$ that image to the same point $\vec{x}$ but differ in their depths will have different motion field vectors. Rieger and Lawton noted that the difference in their motion field vectors will describe a line in the image plane that passes through the FOE. Their method involves taking neighbouring[2] image points and measuring the differences in the flow vectors, and fitting these differences to a line in the image plane. By finding the intersection of these lines it is possible to estimate the FOE.

Subspace methods work in a similar fashion. The relation

$$\vec{\tau} = \sum_{k=1}^{K} c_{ik} \left[\frac{f}{X_3(\vec{x}_k)} \vec{T} \times \vec{x}_k \right] = \vec{T} \times \left(f \sum_{k=1}^{K} \frac{c_{ik}}{X_3(\vec{x}_k)} \vec{x}_k \right)$$

shows that $\vec{\tau}$ is a vector orthogonal to the translational direction, $\vec{T}$. The constraint $\vec{\tau}$ defines a plane, containing $\vec{T}$, which intersects the image plane in a line. This line will therefore intersect the FOE. The linear constraint $\vec{\tau}$ can be thought of as the error in interpolating the function $\vec{u}(\vec{x}_k) \times \vec{x}_k$ by a quadratic polynomial in image location. To illustrate this point, consider using the function

$$\hat{f}(\vec{x}; \alpha, \beta, \gamma) = \begin{bmatrix} \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1^2 + \alpha_4 x_1 x_2 + \alpha_5 x_2^2 \\ \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_1 x_2 + \beta_5 x_2^2 \\ \gamma_0 + \gamma_1 x_1 + \gamma_2 x_2 + \gamma_3 x_1^2 + \gamma_4 x_1 x_2 + \gamma_5 x_2^2 \end{bmatrix}$$

[2] Most optic flow algorithms return only a single flow vector for a given image location. By choosing neighbouring flow vectors it is possible to approximate the flow-difference vectors.

to interpolate the function $\vec{f}(\vec{x}) = \vec{u}(\vec{x}_k) \times \vec{x}_k$. Let $G = [\vec{f}(\vec{x}_1), \ldots, \vec{f}(\vec{x}_K)]$ be a measurement of $\vec{f}$ at K points. A least-squares solution for the parameters $[\vec{\alpha}\ \vec{\beta}\ \vec{\gamma}]$ is given by

$$[\vec{\alpha}\ \vec{\beta}\ \vec{\gamma}] = \left(FF^T\right)^{-1} FG^T$$

where

$$F = \begin{bmatrix} 1 & & 1 \\ [\vec{x}_1]_1 & & [\vec{x}_K]_1 \\ [\vec{x}_1]_2 & \cdots & [\vec{x}_K]_2 \\ [\vec{x}_1]_1^2 & & [\vec{x}_K]_1^2 \\ [\vec{x}_1]_1[\vec{x}_1]_2 & & [\vec{x}_K]_1[\vec{x}_K]_2 \\ [\vec{x}_1]_2^2 & & [\vec{x}_K]_2^2 \end{bmatrix}.$$

Assume that F is of full rank and that $K > 6$. This gives $F^T[\vec{\alpha}\ \vec{\beta}\ \vec{\gamma}] = \hat{G}^T$. If one defines $e = G^T - \hat{G}^T$ and choose $\vec{c} \in \text{null}(F)$, then it is simple to show that

$$e^T\vec{c} = G\vec{c}\,, \tag{4.5}$$

where it is recognized that $G\vec{c} = \vec{\tau}$. Therefore, the linear constraint $\vec{\tau}$ is a function of the interpolation error. Note that the interpolation parameters $\vec{\alpha}$, $\vec{\beta}$ and $\vec{\gamma}$ do not appear on the right-hand side of Eqn. 4.5. This implies that the only part of the interpolation error that is used to construct $\vec{\tau}$ is that part which is orthogonal to the space spanned by the columns of F. This means that it is never necessary to explicitly solve for the interpolation parameters. A quadratic function will obviously be capable of interpolating the rotational component perfectly, but given sufficient depth variation will be unable to do the same for the translational component. Therefore, the interpolation error will be entirely due to the translational component.

The subspace methods are similar in flavour to the method of Rieger & Lawton, but have the advantage of being exact in the absence of noise.

4.4 Relation of Subspace Methods to the 'E' Matrix

'E'-matrix methods have been presented in detail in Section 3.2. Subspace methods are similar to 'E'-matrix methods in that both use algebraic manipulation to factor out the effect of rotation from the observed visual motion. 'E'-matrix methods are different in that they are formulated for the case of discrete motion, as opposed to the instantaneous case for which subspace methods have been derived.[3] Discrete motion can be described by

$$\vec{X}' = R\vec{X} + \vec{T}$$

where $\vec{X}$ and $\vec{X}'$ are the locations of a point before and after the motion, R is a rotation matrix, and $\vec{T}$ is a displacement vector. ($\vec{T}$ should not be confused with instantaneous translational velocity used elsewhere in this thesis.) Both methods allow for straightforward recovery of the translational direction. Jepson & Heeger [45] report improved performance of the subspace methods over 'E'-matrix methods in the presence of noise. Faugeras [21] also notes the sensitivity of the 'E'-matrix to noise. Weng *et. al.* [90] enumerate a number of cases that are problematic for the 'E'-matrix method. First, if the locations of the matched feature pairs used to recover E lie on a quadratic containing the centre of projection of the camera at the time the two images were captured, then the solution for E becomes degenerate. This is not a large problem, as generally more than 8 points are used to recover E and the increase in the number of points decreases the likelihood that the degenerate case will be encountered. Second, in the event that $||\vec{T}||$ is close to zero, the recovery of the translational direction is not reliable. This case is of interest when small displacements are considered, such as optic flow. While the subspace methods will not work for $\vec{T} = 0$ (see below), the use of small displacements is acceptable. It should be noted that the accurate estimation of optic flow is a more difficult task than tracking features. Third, the error in estimating both translational direction

[3] Attempts have been made to formulate subspace methods for the discrete case [88].

and rotation increases as the translational direction deviates from the optical axis. In other words, the method is more accurate for translation perpendicular to the image plane. This manifests itself in the form of a bias towards the optical axis in the estimates for translational direction, and is a problem shared with the subspace methods. As was shown in Section 4.2.4, this bias can be removed when using linear constraints.

4.5 Situations Where the Subspace Methods Fail

In this section situations where the subspace constraints will be insufficient to determine if an object is moving independently will be discussed. From the form of the linear constraints (Eqn. 4.2) and the bilinear constraints (Eqn. 4.1) it is seen that when $\vec{T} = 0$ there will be no constraint. In practice, noise will cause constraints recovered in this case to be non-zero, but this can be handled by using a threshold based on the SNR. Therefore, in the event that no translational motion is present, it will not be possible to use subspace constraints to detect IMOs. The problem of segmenting purely rotational motion fields in the presence of IMOs is considerably less difficult, and is not considered in this thesis.

A more serious problem arises when the spatial extent of the IMO is small and its motion generates a motion field that is locally consistent with that of the background. This can be stated more formally by requiring that the plane defined by the translational velocities (observer relative to background and observer relative to object) contains the image location vector of the object. In the event that the two translational velocities are parallel then a family of planes is defined, and it is obvious that one of these planes will contain the image location vector of the object. An example of this is the case when an observer on a moving train watches a car moving along a road parallel to the train tracks. In this situation the car presents a motion field that is parallel to the motion field caused by the train's translation. The car may be moving at a different speed and possibly even in the opposite direction. The flow from the car is consistent with the flow generated by the background. This can be seen by

considering a $\vec{\tau}$ constraint constructed from flow samples from the car. If the sign of $\vec{u}(\vec{x}_k)$ changes for every flow sample used to construct a linear constraint $\vec{\tau}$, then the constraint itself only changes sign, as can be seen from Eqn. 4.2. This means that the linear constraint recovered from the car will be consistent with that recovered from the background, and the linear and bilinear constraints will be insufficient to identify the IMO. The recovery of relative depth information (scene structure) may be used to help resolve this ambiguity. Depth structure recovered using estimates of egomotion can be expected to be inconsistent for IMOs. For example, if the car has the same translation as the train and no rotation, then the observer perceives no image motion related to the car. Since egomotion is non-zero, the only possible interpretation for this region of zero flow, assuming that it arises from the static environment, is that it represents depth structure at infinite distance. The observer would interpret this as a moving "hole" in the road, and reject the interpretation as unrealistic.[4] The case of identical translation is a special one. In general, the recovered depth values for IMOs incorrectly assumed to be part of the static environment will evolve (in time) in a manner which is not consistent with the egomotion estimates. This allows for an alternate method of identifying IMOs when subspace methods fail. This is discussed in detail in Chapter 11.

In the event that the flow samples used to generate a linear constraint came partly from the car and partly from the background, the information conveyed by the constraint may change. However, in this case an underlying assumption of the subspace methods is violated, namely that the flow samples all come from a rigid environment. In image sequences containing IMOs one should expect this assumption to be violated with regularity. In the next section this case is considered.

[4]Note that this requires contextual knowledge on the part of the observer.

4.5.1 Effects of Independently Moving Object Boundaries

The discussion of subspace methods thus far has ignored a basic assumption made by the subspace methods. This assumption is that the motion parameters $\vec{T}$ and $\vec{\Omega}$ define the motion of the observer relative to a static environment (*rigidity assumption*). This assumption is violated when considering real image sequences which contain IMOs. It is therefore necessary to consider the consequences of violating the rigidity assumption.

The subspace methods assume that the bilinear constraints which are combined to generate a linear constraint represent the images of points which are all moving with the same relative 3-D velocity with respect to the observer. In a static environment, where everything is stationary except for the observer, this requirement is satisfied. However, to analyze image sequences in which there is independent object motion, it is necessary to understand what happens when bilinear constraints from both background and IMOs are combined. It can be shown [45] that, for a rigid environment, the following relation holds:

$$\vec{\tau} = \sum_{k=1}^{K} c_k \left[\frac{f}{X_3(\vec{x}_k)} \left(\vec{T} \times \vec{x}_k \right) \right] . \tag{4.6}$$

This will facilitate an understanding of what happens when linear constraints are generated across a boundary of an IMO. The discussion which follows assumes that the IMO has no rotational motion with respect to the background. If it did, then the c_k would fail to cancel the rotational terms. Although it is not generally the case that the IMO will have no rotation with respect to the background, the following analysis will still provide insight into the formation of linear constraints across IMO boundaries. Consider the case where the sampling points represent flow from the background and an IMO. The translational velocity of the background with respect to the observer is $\vec{T}_{back}$ and the translational velocity of the object with respect to

the observer is $\vec{T}_{obj}$. Define the characteristic function

$$\chi(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \text{ represents a point on the background} \\ 0, & \text{if } \vec{x} \text{ represents a point on the object} \end{cases}$$

and also $\vec{T}_r = \vec{T}_{obj} - \vec{T}_{back}$. Eqn. 4.6 can be rewritten as

$$\vec{\tau} = \sum_{k=1}^{K} c_k \left[\frac{f}{X_3(\vec{x}_k)} \left(\left(\chi(\vec{x}_k)\vec{T}_{back} + (1 - \chi(\vec{x}_k))\,\vec{T}_{obj} \right) \times \vec{x}_k \right) \right] . \tag{4.7}$$

Rearranging terms it is possible to write

$$\begin{aligned} \chi\vec{T}_{back} + (1-\chi)\vec{T}_{obj} &= \frac{1}{2}\vec{T}_{back} + \chi\vec{T}_{back} - \frac{1}{2}\vec{T}_{back} + \frac{1}{2}\vec{T}_{obj} + \frac{1}{2}\vec{T}_{obj} - \chi\vec{T}_{obj} \\ &= \frac{1}{2}\left[\vec{T}_{back} + \vec{T}_{obj}\right] + \left(\frac{1}{2} - \chi\right)\vec{T}_{obj} - \left(\frac{1}{2} - \chi\right)\vec{T}_{back} \\ &= \frac{1}{2}\left[\vec{T}_{back} + \vec{T}_{obj}\right] + \left(\frac{1}{2} - \chi\right)\vec{T}_r \end{aligned} \tag{4.8}$$

and by inserting Eqn. 4.8 into Eqn. 4.7 the following relation is obtained,

$$\vec{\tau} = \frac{1}{2}\left[\vec{\tau}_{back} + \vec{\tau}_{obj}\right] + \sum_{k=1}^{K} c_k \left[\frac{f}{X_3(\vec{x}_k)} \left(\frac{1}{2} - \chi(\vec{x}_k) \right) \vec{T}_r \times \vec{x}_k \right] \tag{4.9}$$

where $\vec{\tau}_{back}$ and $\vec{\tau}_{obj}$ are the constraints that would have been recovered if only the background or object motions (respectively) existed. The relative translation of the object with respect to the background is given by $\vec{T}_r = \vec{T}_{back} - \vec{T}_{obj}$. At the two extremes, it is seen that the generated constraints are those that would be expected for the individual motions alone. In general, the resulting constraint is the average of the two, plus an additional term which depends on the relative velocity of the object to the background. In practice it is observed that many of these constraints lie near the average constraint $\frac{1}{2}\left[\vec{\tau}_{back} + \vec{\tau}_{obj}\right]$, suggesting that the second term in Eqn. 4.9 is often small in comparison to the first term.

If *a priori* segmentation information is available, the constraints can be generated using custom sampling geometries that never cross boundaries of IMOs. Generation of the required $\vec{c}$ coefficients is straightforward once the sampling geometry is known.

4.6 Summary

In this chapter the use of subspace methods to generate linear and bilinear constraints on 3-D image motion has been described. The underlying assumptions of subspace methods have been considered, as well as the effect of noise on the derived constraints. The presence of IMOs violates the rigidity assumption, and any linear constraints generated across an IMO boundary must be considered erroneous. A quantitative description of the effect of IMO boundaries on constraint generation is given, affording insight into the expected nature of these constraints. A comparison of subspace methods with the methods of Rieger & Lawton [74] and also the 'E'-matrix methods shows that the methods have much in common, although they are by no means identical. Cases in which subspace methods are insufficient to detect IMOs are detailed.

It is proposed that linear and bilinear subspace constraints are suitable for clustering to perform motion segmentation. This allows motion segmentation based on 3-D motion, rather than on the discontinuities found in 2-D image motion. This is an important ability in systems which need to navigate in unknown environments. It now remains to describe a method for performing segmentation of constraints. In the next chapter, the concept of mixture models is presented as a suitable vehicle for performing this segmentation. The EM-algorithm is proposed for the calculation of mixture proportions and parameters of the individual process distributions.

Chapter 5

Mixture Models & The EM-Algorithm

In Chapter 4 the use of subspace methods for generating constraints on the 3-D motion parameters (translation and rotation) underlying an optic flow field measurement was discussed. One underlying assumption of the methods was that the flow field was due to an observer moving in a static (rigid) environment. When one considers image sequences containing IMOs, this assumption is no longer valid. If the segmentation of the image were known *a priori*, according to which parts of the image belong to the static environment and which parts belong to IMOs, it would be possible to generate constraints accordingly and recover the motion parameters for these different regions. Conversely, if the number of IMOs and their velocities relative to the observer were known, determining the segmentation of the image would be considerably simpler. Unfortunately, neither information is available for use. In this thesis I make use of a statistical concept known as mixture models to attempt a simultaneous solution for segmentation and motion parameters.

5.1 Mixture Models

Mixture models are a class of statistical models used when a set of observations may have more than one underlying process, *i.e.* any given data point in the set will

have been generated by one of several processes [64]. In general, it is not known which observations have been generated by which process, nor are the parameters of the individual processes known. In this case, a *mixture of distributions* is used to model the data, with each process[1] having its own distribution and parameters. It is assumed that the form of the underlying distributions is known, as well as the number of distributions. The general formulation for a mixture of distributions with observations x and parameters ϕ is

$$p(x|\phi) = \sum_{j=1}^{m} \pi_j p_j(x|\phi_j)$$

$$\sum_{j=1}^{m} \pi_j = 1$$

$$0 \leq \pi_i \leq 1$$

where m is the number of distributions in the mixture, $\{\pi_j\}_{j=1}^{m}$ are the mixture proportions of the distributions, and p_j is the jth probability density with parameters ϕ_j. Given a set of observations $\{x_i\}_{i=1}^{n}$ one would like to i) estimate the parameters for each underlying distribution, and ii) determine the probability that a given data point is the result of a given process. The second objective is commonly referred to as *clustering*.

It was noted above that the number of distributions in the mixture was assumed to be known in advance. This is not always the case. Testing for the number of processes in a mixture is a difficult and, in general, unsolved problem [64]. In a specific problem, however, it may be possible to use domain-specific information to estimate the number of processes.

The specific application of mixture models to the problem of 3-D motion segmentation will be discussed next.

[1] The processes are also referred to as *populations* or *modes*.

5.2 Application of Mixture Models to Motion Segmentation

In using mixture models to assist with motion segmentation, constraints on 3-D motion (both bilinear and linear) will be used as the observed data, and the parameters of the mixture distributions will include the motion parameters. If linear constraints are used then $\vec{T}$ is included as a parameter in the underlying distribution. In the case where the data are composed of bilinear constraints both $\vec{T}$ and $\vec{\Omega}$ are used. In each case a uniform distribution will be used to model outliers, and when fitting the mixture model to the linear constraints will attempt to estimate the number of processes in the mixture by examining the structure of constraints assigned to the outlier distribution.

5.2.1 Segmentation of Linear Constraints to Recover $\vec{T}$

In this section the form of the distributions to be used when attempting to fit a mixture model to the linear constraints will be proposed. The use of a mixture model requires that the form of the distribution be specified in advance. If one considers a mixture involving M underlying translations $\left\{\vec{T}_j\right\}_{j=1}^{M}$, then the probability density function (PDF) for constraint $\vec{\tau}$ is taken to be

$$\begin{aligned} p(\vec{\tau}|\vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M) &= \pi_0 p_0 + \sum_{j=1}^{M} \pi_j p(\vec{\tau}|\vec{T}_j, \sigma_j) \\ \sum_{j=0}^{M} \pi_j &= 1 \\ 0 \leq &\pi_j \leq 1 \end{aligned} \tag{5.1}$$

where the variances $\left\{\sigma_j^2\right\}_{j=1}^{M}$ will depend on the noise in the optic flow used to generate the linear constraints. The parameters π_j and $(\vec{T}_j, \sigma_j)$ are unknowns and can be estimated.

Now consider the form of $p(\vec{\tau}|\vec{T}_j, \sigma_j)$. The $\vec{\tau}$ constraints are derived to be orthog-

onal to the translational motion, $\vec{T}$. The following PDF is proposed, defined over the surface of the unit sphere (*i.e.* $\vec{\tau}$ and $\vec{T}$ are unit vectors):

$$p(\vec{\tau}|\vec{T}_j, \sigma_j) = \frac{1}{\gamma} \exp\left\{ -\frac{(\vec{\tau}^T \vec{T}_j)^2}{\sigma_j^2} \right\}$$
$$\gamma = \frac{4}{3}\pi \left(2.0 + \exp\{-1/\sigma_j^2\}\right) \ . \tag{5.2}$$

The form of the PDF requires some explanation. The term $\vec{\tau}^T \vec{T}_j$ represents the deviation from the desired orthogonal relationship between $\vec{T}_j$ and the constraint $\vec{\tau}$. A Gaussian model is used for this deviation, or error. The choice for γ is determined subject to the condition that the function must integrate to unity over the surface of the unit sphere. Figure 5.1 shows a graphical representation of this function on the unit sphere. Note that the constraint $\vec{T}^T \vec{\tau} = 0$ also admits the possibility of $-\vec{T}$.

Based on the noise analysis of Section 4.2.4 it would be appropriate to replace σ_j^2 in Eqn. 5.2 with $\vec{T}_j^T C \vec{T}_j$ to represent the anisotropic noise distribution in the $\vec{\tau}$. Here, C is a covariance matrix of form similar to the matrix M in Eqn. 4.4. However, in an effort to keep the model simple (and linear) the PDF described in Eqn. 5.2 is adopted. For the purposes of clustering $\vec{\tau}$ constraints this appears sufficient, and final estimates for $\vec{T}_j$ can be corrected after the clustering is complete.

Eqn. 5.1 contains the term $\pi_0 p_0$. This term is meant to model outliers in the data, as in Jepson & Black [43]. The outlier distribution is modelled by a uniform distribution, $p_0 = 1/4\pi$.

Given a mixture of distributions for our motion processes and a constraint, it is possible to calculate the probability that a constraint belongs to a given process. This is useful because it is necessary to be able to identify which constraints belong to which motion processes in order to perform segmentation and also estimate motion parameters for a process. The need for ownership probabilities will become clear when the 'M'-step of the EM-algorithm is discussed in Section 5.3. Denote the probability that $\vec{\tau}_i$ belongs to process j by s_{ij}. It may be calculated as

$$s_{ij} = \pi_j p(\vec{\tau}_i|\vec{T}_j, \sigma_j) / p(\vec{\tau}_i|\vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M) \ . \tag{5.3}$$

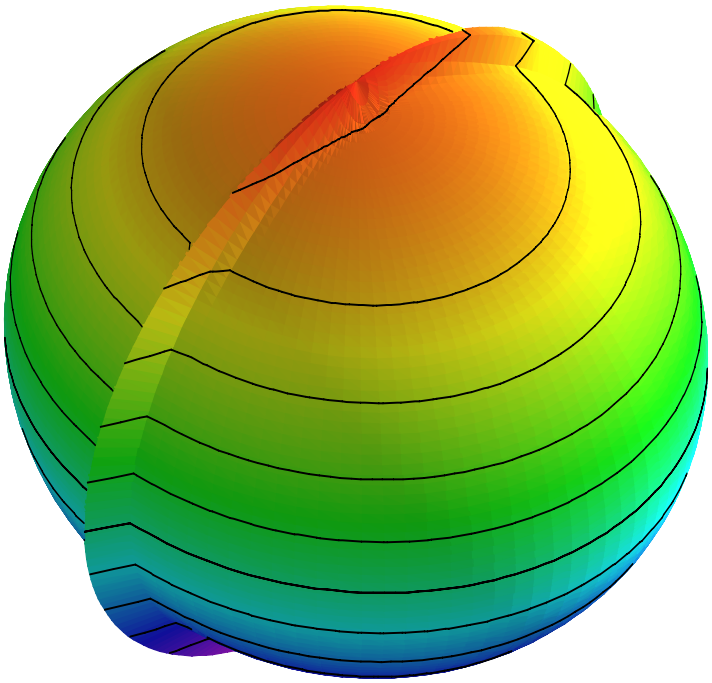


Figure 5.1: The displacement from the sphere demonstrates the function $p(\vec{\tau}|\vec{T}) = \frac{1}{\gamma}\exp\left\{\vec{\tau}^T\vec{T}\vec{T}^T\vec{\tau}/\sigma^2\right\}, \gamma = \frac{4}{3}\pi(2 + e^{-1/\sigma^2})$. This function is used for determining the probability that a linear constraint belongs to a particular translational direction.

5.2.2 Segmentation of Bilinear Constraints to Recover $\vec{T}$ and $\vec{\Omega}$

Determining the probability of a given bilinear constraint follows a similar approach to that in the previous section. In referring to a bilinear constraint the flow estimates and their image locations, $\{\vec{u}_k, \vec{x}_k\}_{k=1}^{K}$, are specified. The probability of a constraint with respect to a mixture of motion processes, $\left\{\vec{T}_j, \vec{\Omega}_j\right\}_{j=1}^{M}$, is

$$\begin{aligned} p(\vec{u}_k|\vec{x}_k, \vec{T}_1, \vec{\Omega}_1, \sigma_1 \ldots \vec{T}_M, \vec{\Omega}_M, \sigma_M) &= \pi_0 p_0 + \sum_{j=1}^{M} \pi_j p(\vec{u}_k|\vec{x}_k, \vec{T}_j, \vec{\Omega}_j, \sigma_j) \\ \sum_{j=0}^{M} \pi_j = 1, \qquad & 0 \leq \pi_j \leq 1 \ . \end{aligned} \tag{5.4}$$

This is the bilinear equivalent of Eqn. 5.1. The variances and mixture proportions in this equation are different than those in the linear case, but we will use the same terminology to avoid introducing new symbols.

The PDF relating a bilinear constraint to a particular motion process is:

$$p(\vec{u}_k|\vec{x}_k, \vec{T}_j, \vec{\Omega}_j, \sigma_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left\{ -\frac{(\vec{T}_j^T(\vec{a}_k(\vec{u}_k) + B_k\vec{\Omega}_j))^2}{2\sigma_j^2\left([\vec{T}_j \times \vec{x}_k]_1^2 + [\vec{T}_j \times \vec{x}_k]_2^2\right)} \right\} \tag{5.5}$$

where $[\vec{T} \times \vec{x}]_p$ is the pth component of $\vec{T} \times \vec{x}$. The PDF is a Gaussian whose argument is the deviation of the constraint, evaluated at $\vec{T}_j$ and $\vec{\Omega}_j$, from zero. Ownership probabilities are calculated as in Eqn. 5.3, substituting the bilinear distributions for the linear ones.

The form of Eqn. 5.5 requires some explanation. The bilinear constraint can be written as $(\vec{T} \times \vec{x})^T\vec{u} + (\vec{T} \times \vec{x})^T(\vec{x} \times \vec{\Omega}) = 0$. This is the equation of a line L in the space defined by (u_1, u_2), the non-zero components of $\vec{u}$. The line is defined by the values of $\vec{T}$ and $\vec{\Omega}$. For a given constraint line one is interested in the minimum distance from the measured flow vector $\vec{u}'$ to the constraint line (see Figure 5.2). This

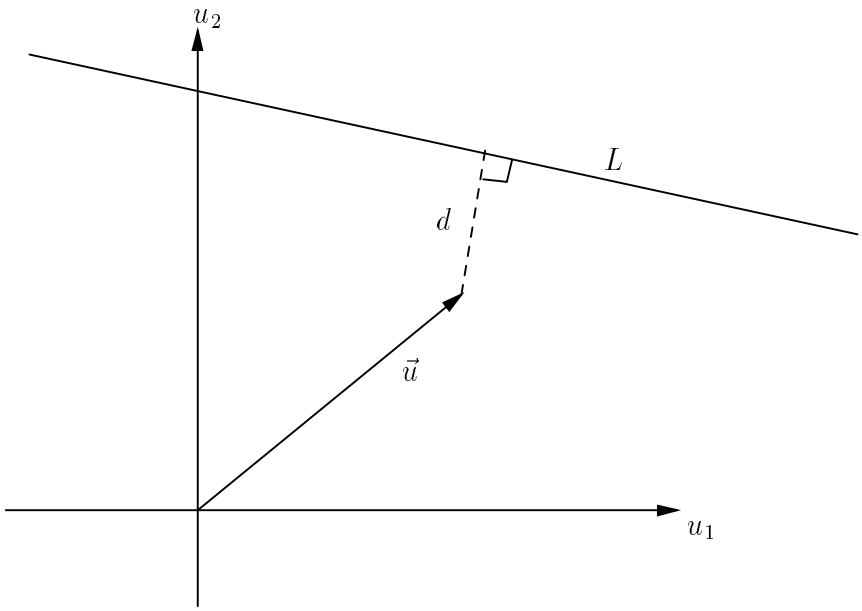


Figure 5.2: The error measure used in determining ownership for bilinear constraints is the minimum distance from the measured flow vector $\vec{u}'$ to the constraint line L defined by the values of $\vec{T}$ and $\vec{\Omega}$.

distance is given by

$$d = \frac{[\vec{T} \times \vec{x}]_1 u_1' + [\vec{T} \times \vec{x}]_2 u_2' + (\vec{T} \times \vec{x})^T (\vec{x} \times \vec{\Omega})}{\sqrt{[\vec{T} \times \vec{x}]_1^2 + [\vec{T} \times \vec{x}]_2^2}} \; .$$

It is seen that Eqn. 5.5 is just a Gaussian PDF in the variable d. The term

$$\sigma^2 \left([\vec{T} \times \vec{x}]_1^2 + [\vec{T} \times \vec{x}]_2^2 \right)$$

turns out to be equivalent to the quantity $\vec{T}^T C \vec{T}$ for the covariance matrix defined for a single constraint, and assuming zero-mean additive noise in the flow.

5.3 The EM-Algorithm as a Solution

The concept of a mixture model provides a powerful tool for modelling complex data. In order to obtain estimates for the process parameters and mixture proportions it is necessary to proceed carefully. If the ownership probabilities for each constraint

were known, it would be a straightforward matter to estimate the optimal parameters for the corresponding motion processes. Conversely, if the motion process parameters were known it would be straightforward to estimate the ownership probabilities. However, attempting to simultaneously estimate ownership and process parameters is not as simple. Furthermore, it is necessary to contend with outliers in the data, such as those that occur when a linear constraint is formed from flow samples that cross independent moving object boundaries.

Researchers solving problems of this nature have turned to a technique known as the *EM-algorithm* [15, 64]. The EM-algorithm is an iterative, 2-step method where "EM" stands for *expectation* & *maximization*, the two basic steps involved.

The algorithm starts with an initial guess for the motion parameters. The expectation step assigns an ownership probability for each constraint to each motion process on the assumption that the current motion parameters are the correct ones. The mixture proportions are also estimated at this time. The maximization step solves for the motion parameters on the assumption that the assigned ownership values are correct. Exact details of how this is done for the motion models is given in the next chapter. Each expectation-maximization pair constitutes one iteration of the algorithm. Dempster *et. al.* [15] have shown that each iteration of the EM-algorithm is guaranteed to improve (or, at worst leave unchanged) the likelihood of the model. The likelihood, in the linear-constraint case for example, is given by

$$L(\vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M) = \prod_{i=1}^{N} p(\vec{\tau}_i | \vec{T}_1, \sigma_1 \ldots \vec{T}_M, \sigma_M) \; . \tag{5.6}$$

The case in which the likelihood is left unchanged corresponds to having found a local maximum in the likelihood function.

No results are given for the rate of convergence to a maximum likelihood point, although methods for decreasing the cost of each iteration have been proposed [65]. Also, there is no guarantee that the EM-algorithm will converge to a global maximum.[2]

[2] In fact, given that a typical likelihood function will be quite nonlinear in terms of the estimated parameters, it is most likely that a randomly chosen initial guess would lead to a local, and not global, maximum being obtained.

5.4 Summary

In this chapter the concept of a "mixture of distributions" was presented as a tool for modelling image motion containing egomotion and IMO's. The EM-algorithm provides an iterative method for performing simultaneous segmentation and parameter estimation. As the reader will see, mixture models and the EM algorithm provide an elegant framework in which to perform 3-D motion segmentation. In the next chapter the details of implementing the EM-algorithm for subspace constraints are considered. Some of the problems inherent in the EM-algorithm, specifically the matter of determining initial guesses and the number of distributions, will be presented and discussed in the context of the problem at hand.

Chapter 6

3-D Motion Segmentation

In Chapter 5 mixture models and the EM-algorithm, and their application to the problem of motion segmentation, were discussed. The probability of a given constraint (linear or bilinear) given specific motion parameters was used to develop a sense of "ownership" of a constraint by a motion process. In this chapter the details pertaining to the "M"-step of the EM-algorithm will be refined. The EM-algorithm, while extremely useful in determining process parameters and their mixture proportions, has inherent difficulties. The likelihood function for a given mixture of distributions will almost certainly be nonlinear. Since methods for optimizing nonlinear equations seldom guarantee that a global maximum or minimum will be achieved, the initial guess for the motion processes and mixture parameters is critical. Also, the problem of deciding on the number of distributions in a mixture is a difficult and, in general, unsolved problem [64]. Both of these issues are discussed in this chapter.

6.1 Clustering Linear Constraints

First consider the clustering of the linear constraints. These constraints allow solution for translational direction independent of rotation. The linear constraints will also be used to determine the number of motion processes. Once estimates for the number of translational motions and their directions have been made, it is possible to generate initial estimates for the rotations of each process. Estimates for both $\vec{T}$ and $\vec{\Omega}$ may

be further refined by clustering the bilinear constraints.

6.1.1 Estimating Motion Parameters

The "M"-step of each iteration of the EM-algorithm attempts to estimate the translational direction of each process. The data are the weighted linear constraints, $\{w_i\vec{\tau}_i\}_{i=1}^N$. Recall that the "E"-step generates estimates of the probability that the ith constraint belongs to the jth process, namely s_{ij}. With the inclusion of ownership probabilities, Eqn. 4.3 is modified to become

$$D_j = \frac{\sum_{i=1}^N s_{ij} w_i^2 \vec{\tau}_i \vec{\tau}_i^T}{\sum_{i=1}^N s_{ij}} \; . \tag{6.1}$$

The translational direction $\vec{T}_j$ belonging to the jth process is then estimated by the eigenvector corresponding to the minimum eigenvalue of D_j.

6.1.2 Estimating Variances

In the ownership function of Eqn. 5.2 a variance σ_j^2 is used to help determine the ownership of constraint i by process j. This variance is also determined during the "M"-step using the relation

$$\sigma_j^2 = \frac{\sum_{i=1}^N s_{ij} (\vec{\tau}_i^T \vec{T}_j)^2}{\sum_{i=1}^N s_{ij}} \; . \tag{6.2}$$

Note that this expression, like Eqn. 5.2, is independent of the weights w_i. This is due to the fact that it is only the angle between $\vec{\tau}_i$ and $\vec{T}_j$ that affects the ownership probability, and not the magnitude of $w_i\vec{\tau}_i$.

6.2 Generating Initial Guesses

It has been shown how the parameters of a motion process are updated during each iteration of the algorithm, but has yet to be discussed how to generate the initial guesses. A poor choice of initial guess for motion parameters may lead to slow con-

vergence, or convergence to a local, not global, maximum of the likelihood function. It is therefore important to generate a good initial guess. One can do this by solving as if only one process existed. Initial guesses for translational direction can be made by assuming the linear constraints represent a single translational motion and solving for $\vec{T}$ as described in Section 4.2.3. Specifically, solve for the eigenvector corresponding to the smallest eigenvalue of D as in Eqn. 4.3. This results in $\vec{T}_1$. It is possible to estimate σ_1^2 using Eqn. 6.2 with $s_{i1} = 1$ for $i = 1 \ldots N$. Since the entire set of $\vec{\tau}_i$ may contain other processes and outliers this estimate for σ_1^2 is likely too large—therefore for an initial guess it is reduced by half. By using a mixture model with two processes, one for outliers and one for the dominant translation process, it is possible to refine this estimate and determine outliers. Assuming that constraints generated by any other motion processes will be assigned ownership to the outlier process, the outlier population is then probed for evidence that other translational motions exist. This procedure continues until no new processes can be found. Estimates for process variances can be made in the same way, but since the outlier population may contain multiple processes as well as *bona fide* outliers, one expects these estimates to be too large.

Since we now possess an initial guess for $\vec{T}_1$ and σ_1^2, the EM-algorithm is used with two processes to classify the $\vec{\tau}_i$ as belonging to either $\vec{T}_1$ or the outlier population. As will be shown in Section 6.3, it is possible to examine the structure of the outlier population to determine if other processes exist.

6.3 Splitting Processes

It is necessary to determine the number of processes in the 3-D motion estimation problem. It was mentioned earlier that this is, in general, a difficult problem. In the context of 3-D motion segmentation, however, it may be possible to use domain-specific information to determine the number of processes. As outlined in the previous section, it is assumed that there is at least 1 process (this will usually be egomotion). Any data which do not support this process are deemed to be "outliers". Next,

probe the structure of the outlier process to see if there is evidence that a second process exists. Whenever all the underlying motion processes are not represented, it is assumed that some or all of the constraints belonging to unrepresented processes will be assigned ownership to the outlier population. If there is evidence for a new process, the hypothesis is revised to include this new process and re-cluster the data. After re-clustering, again examine the new outlier population to see if there is evidence for yet another process. This continues until there is no longer any evidence for new processes, or until new processes stop being unique. In short, the approach is to examine the structure of the outlier population after each clustering in order to decide if there is evidence for another process, and if so to generate an initial guess for that process. The problems of generating initial guesses and determining the number of underlying processes are seen to be related.

In order to examine the structure of the outlier population calculate D_0 according to Eqn. 6.1 and examine its eigenvalues, $\lambda_1 \geq \lambda_2 \geq \lambda_3$. It is expected to find one of three cases.

1. The smallest eigenvalue is significantly smaller than the other two ($\lambda_1 \geq \lambda_2 \gg \lambda_3$). This indicates the possibility of one new translational direction, *i.e.* the outlier population constraints form a great circle[1] on the unit sphere. This initial estimate for this single direction is given by the eigenvector corresponding to λ_3.

2. The two smallest eigenvalues are of similar size as compared to λ_1, ($\lambda_1 \gg \lambda_2 \approx \lambda_3$). This indicates that the constraints are clustered in a small region on the unit circle in a roughly circular pattern. In this case there is an entire plane of possible directions for $\vec{T}$. This plane will be defined by the eigendirections corresponding to λ_2 and λ_3.

3. All eigenvalues are of roughly equal magnitude ($\lambda_1 \approx \lambda_2 \approx \lambda_3$). This suggests that the constraints in the outlier population are distributed approximately

[1]Considering noise, a more likely approximation is that of a flattened (pancake-like) ellipsoid.

uniformly over the surface of the unit sphere. There may or may not be unique underlying translations, but there is no indication of a preferred direction for $\vec{T}$.

In order to distinguish between the first two possibilities, compare λ_2 to the geometric mean of the largest and smallest eigenvalues, $\sqrt{\lambda_1 \lambda_3}$. In the first case one new translation processs added, defined by the eigenvector corresponding to λ_3. In the second case two new translational directions defined by the eigenvectors corresponding to λ_3 and λ_2 are added. The addition of a new translational direction to the model requires that a new process be included in the EM iterations. A variance for each new process is estimated based on the constraint ownerships for the outlier population and Eqn. 6.2. The EM-algorithm is performed on the model that includes the new process(es), with the initial values for the old processes being the same as their final values before splitting took place. This "splitting" of the outlier population continues until either the mixture proportion of the outlier population π_0 becomes too small, indicating it has ownership of few constraints, or until the new translational processes cease to be unique as compared to the processes already existing. This comparison takes place at the end of clustering (*i.e.* at the conclusion of the EM-algorithm), and is accomplished by comparing $p(\vec{T}_j|D_j)$ to $p(\vec{T}_l|D_j)$, where l is the index of the newly added process. The comparison involves D_j, the D matrix constructed from the constraints owned by process j. The newly added process l is compared to all the previous processes. The comparison function is defined as

$$\begin{aligned} p(\vec{T}|D) &= \frac{1}{\gamma} \exp\left\{-\vec{T}^T D \vec{T}\right\} \\ \gamma &= \frac{4}{3}\pi(e^{-\lambda_1} + e^{-\lambda_2} + e^{-\lambda_3}) \end{aligned} \tag{6.3}$$

where the λ_i are the eigenvalues of D. One recognizes $\vec{T}^T D \vec{T}$ as the residual error of the estimate $\vec{T}$ with respect to the constraints which compose D. Ideally this residual is zero, but in the presence of noise it will be greater than zero. The use of the negative of this residual as the argument to an exponential will give a Gaussian-shaped distribution on the surface of the unit sphere. When the rank of D is 1 the

distribution is a Gaussian ridge along a great circle along the unit sphere. When the rank of D is 2, it is two 2-D Gaussian surfaces located at opposing points on the sphere. The factor γ is chosen so that the integral of $p(\vec{T}|D)$ over the unit sphere is 1. Each new process is therefore compared to each of the previously-existing processes. When a newly-spawned process is deemed too similar to an already existing one, they may be merged. A new process that "owns" too few constraints at the end of clustering may be deleted.

A summary of the splitting algorithm is as follows:

1. Start by assuming a single translational process. An estimate of its translational direction is calculated by using Eqn 4.3 and finding the eigenvector corresponding to the minimum eigenvalue of D. Note that using Eqn 4.3 is tantamount to assuming that all constraints belong to this process with probability 1. An estimate of variance is computed, and since it is likely too large it is reduced.

2. Use the EM-algorithm to cluster constraints between the single translation process and an outlier population. The outlier population is modelled as a uniform distribution over the unit sphere.

3. Examine the outlier population for evidence of other translational directions. This is done by forming D_0 according to Eqn. 6.1. Note that the quantity s_{i0} is the probability that the ith constraint belongs to the outlier population. The eigenvalues of D_0 are computed.

4. Check to see if $\lambda_1 \approx \lambda_2 \approx \lambda_3$. If so, no new translational directions are added and the splitting process is done.

5. If $\lambda_2 > \sqrt{\lambda_1 \lambda_3}$ then add one new translational direction. The initial estimate is the eigenvector corresponding to λ_3. Estimate a variance using Eqn. 6.2 where $\vec{T}_j$ is the new translational direction and s_{ij} is replaced by s_{i0}.

6. If $\lambda_2 < \sqrt{\lambda_1 \lambda_3}$ then add two new translational directions. The initial estimates are given by the eigenvectors corresponding to λ_3 and λ_2. Calculate estimates of variances for these processes as described in the previous step.

7. Repeat clustering using the EM-algorithm and the new processes.

8. Check the new processes. If they are too similar to existing processes then merge them with the appropriate processes.[2] If a new process has too little support, *i.e.* its mixing proportion falls below some preset threshold, then discard it. In either of these cases the splitting process is complete.

9. If none of the new processes were merged or discarded, then go to Step 3 and continue.

The splitting process is repetitive and terminates when new processes cease to be unique or garner too little support. Once the process terminates, it is hoped that the number of IMOs in the scene has been identified. Once the clustering is complete, it is possible to correct for the anisotropic nature of the noise associated with the linear constraints.

6.4 Clustering Bilinear Constraints

Once the linear constraints have been clustered, an estimate for the number of motion processes as well as their translational directions exists. It is now possible to use the bilinear constraints to refine estimates of the $\vec{T}_j$ as well as to estimate the associated $\vec{\Omega}_j$. It is hoped that improved estimates may be obtained by clustering the bilinear constraints: the linear constraints may be contaminated by being generated across boundaries of IMOs, whereas the bilinear constraints will not have this problem.[3]

6.4.1 Generating Initial Guesses for Rotation

The final estimates for translational directions from the linear constraint clustering can be used to generate initial estimates for the rotation parameters. The number of

[2] Merging two processes is accomplished by combining their ownership weights, and calculating a new D matrix for the process. From this a new $\vec{T}$ and variance can be estimated.

[3] This assumes, of course, that the integration of 2-D motion constraints to recover optic flow was done properly.

processes is now fixed. For each $\vec{T}_j$ we calculate a least-squares estimate for $\vec{\Omega}_j$:

$$\vec{\Omega}_j = \left(\sum_{k=1}^{K} B_k^T \vec{T}_j \vec{T}_j^T B_k \right)^{-1} \sum_{k=1}^{K} B_k^T \vec{T}_j \vec{T}_j^T \vec{a}_k \ . \tag{6.4}$$

Unfortunately, the ownership values for the linear constraints cannot easily be transferred to the bilinear constraints. This is due to the fact that, in general, more than one bilinear constraint is used to generate a linear constraint, and that a particular bilinear constraint may be used in the estimation of more than one linear constraint. Therefore, the initial estimates for rotation are done using all constraints. The results obtained by segmenting with these initial estimates seem good.

6.4.2 Estimating Motion Parameters

In the "E"-step ownership probabilities for constraint k to process j are calculated and denoted by s_{kj}. In order to update the estimates for $\vec{T}$ and $\vec{\Omega}$ based on these ownership probabilities it is necessary to minimize the function

$$\begin{aligned} f(\vec{T}_j, \vec{\Omega}_j) &= \sum_{k=1}^{K} s_{kj} \left[\vec{T}_j^T \left(\vec{a}_k + B_k \vec{\Omega}_j \right) \right]^2 \\ \|\vec{T}_j\| &= 1 \end{aligned} \tag{6.5}$$

subject to the constraint on $\vec{T}_j$ and holding the s_{kj}'s fixed. This is a problem in non-linear optimization, and can be solved using a variety of algorithms [16]. One method is to use a Newton-Raphson algorithm to perform the minimization, incorporating the method of Lagrange multipliers [28] to to enforce the constraint $||\vec{T}_j|| = 1$.

6.4.3 Estimating Variances

The variances σ_j^2 used in the ownership functions of Eqn. 5.4 for the bilinear clustering are estimated once the motion parameters have been updated. The variances are

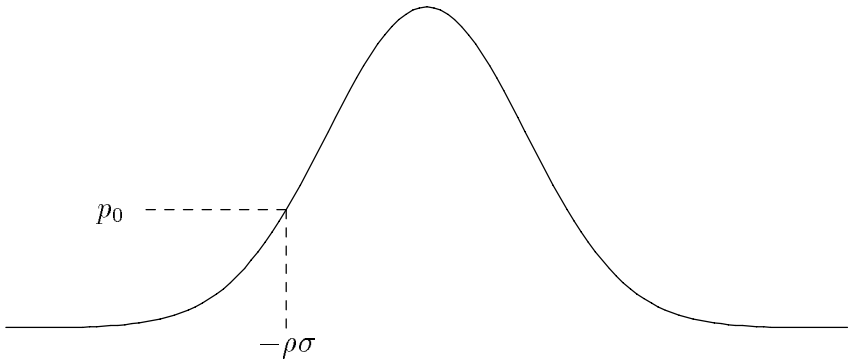


Figure 6.1: The value of p_0 is chosen to give an ownership probability of 0.5 to a point lying a distance of $\rho\sigma$ from the centre of the distribution. This is done by assigning p_0 the value of the distribution at that point. In the case of multiple distributions, the largest value of σ_j is used.

estimated according to the relation

$$\sigma_j^2 = \frac{\sum_{k=1}^{K} s_{kj} \left[\vec{T}_j^T \left(\vec{a}_k + B_k \vec{\Omega}_j \right) \right]^2}{\sum_{k=1}^{K} s_{kj}} \; . \tag{6.6}$$

The initial guess for σ_j^2 can also be generated via this equation by setting $s_{kj} = 1$ for $k = 1 \ldots K$. As with the initial estimate for the variances used in the linear clustering, it is likely that this variance estimate will be too conservative. Again, decrease the initial estimate by a factor of 2.

6.4.4 Choice of p_0

The choice of the constant p_0 in Eqn. 5.4 deserves mention. It is chosen to give an ownership probability of 0.5 to a point which lies ρ standard deviations away from the centre of the main distribution. Therefore, the value of p_0 will change as the σ_j's change. Figure 6.1 gives a graphical interpretation of this. This technique is similar to one used by Jepson & Black [43]. In the presence of multiple non-outlier processes, choose the largest process variance to perform the calculation.

6.5 Recovering Depth Estimates

As indicated by the discussion of recovery of structure from motion in Chapter 4, it is possible to estimate the scene structure given an estimate for $\vec{T}$ and $\vec{\Omega}$. This recovery of structure can assist in dealing with ambiguous motions, as will be seen in Chapter 11.

As can be seen in Eqn. 2.2, the inverse-depth term $1/X_3(\vec{x})$ is multiplied by the translation $\vec{T}$. This means that $\vec{T}$ or $1/X_3(\vec{x})$ can only be recovered up to a multiplicative constant. As mentioned earlier, it is only possible to find the translational direction, $\vec{T}$, such that $\|\vec{T}\| = 1$. When recovering depth information one must be content to recover *relative depth*, not absolute depth.

From Eqn. 2.2 observe that the relation for relative depth may be written as

$$\frac{f}{X_3} = \frac{(P(\vec{x})\vec{T})^T \left[\vec{v}(\vec{x}) - \vec{\Omega} \times \vec{x}\right]}{\|P(\vec{x})\vec{T}\|^2} \tag{6.7}$$

where $\vec{v}(\vec{x})$ is the depth-scaled projected velocity introduced in Section 2.3.1. The quantity $\vec{v}$ is related to $\vec{u}$ by $\vec{v}(\vec{x}) = P(\vec{x})\vec{u}(\vec{x})$ where the operator $P(\vec{x}) = I - \vec{x}\vec{x}^T/\|\vec{x}\|^2$ performs a projection onto a sub-space orthogonal to $\vec{x}$. Here a single multiplicative factor is assumed to be applied to all $\vec{X}$ to account for having redefined $\|\vec{T}\| = 1$.

Chapter 7

Interpretation of Motion Constraint Clusters

One may not always be so lucky as to have linear constraints which define great circles on the unit sphere. A more likely scenario is that the constraints will form *sub-clusters* on a given great-circle. In this event, clustering with an algorithm that only looks for great-circles may not give the best results. Figure 7.1 shows a specific case in which two clusters representing distinct translational directions might be confused by segmenting only with respect to great circles. It is possible to further segment constraints that are consistent with a great circle, once the great circle has been identified. To do this it is not only required that the constraints are orthogonal to the desired translational direction, $\vec{T}$, but that they lie close together on the surface of the unit sphere. This can be accomplished by requiring $\vec{\tau}^T \vec{L} \approx 1$, where $\vec{L}$ is a "location vector" describing the location of the sub-cluster on the unit sphere.

It is convenient to define a new vector, $\vec{L}_\perp = \vec{L} \times \vec{T}$, since one can apply the more

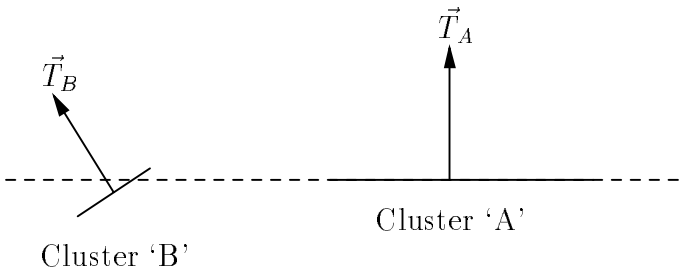


Figure 7.1: This figure shows two clusters of constraints labelled A and B. Cluster A defines a translational direction $\vec{T}_A$ which is in a different direction than $\vec{T}_B$ defined by cluster B. However, if cluster B has a small spatial extent and happens to lie across the great circle defined by cluster A, it may be assigned ownership as if it belonged to cluster A. In this case, the use of small, localized cluster ownership functions is necessary to differentiate between these unique translational directions.

familiar constraint $\vec{\tau}^T \vec{L}_\perp = 0$. The ownership function used is

$$\begin{aligned} p(\vec{\tau}|\vec{T}, \vec{L}_\perp) &= \frac{1}{\gamma} \exp\left\{\vec{\tau}^T D \vec{\tau}\right\} \\ D &= \frac{\vec{T}\vec{T}^T}{\sigma_T^2} + \frac{\vec{L}_\perp \vec{L}_\perp^T}{\sigma_\perp^2} \\ \gamma &= \frac{4}{3}\pi(1 + \exp\{-1/\sigma_T^2\} + \exp\{-1/\sigma_\perp^2\}) \; . \end{aligned} \tag{7.1}$$

An example of this function for $\sigma_T = 0.03$ and $\sigma_\perp = 0.2$ is shown in Figure 7.2.

7.1 Finding Clusters Using the EM Algorithm

In Chapter 6 the problem of segmenting linear constraints using the EM algorithm was considered. The PDF used to calculate ownership values for constraints was based on the model of great-circles of constraints which represented a single translational direction. Now consider the problem of finding clusters of constraints as defined in the previous section. The PDF defined in Eqn. 7.1 is employed to define the ownership of a constraint to a cluster process. Each translational process gives rise to some number of cluster processes[1]. The exact number of cluster processes is not important, but it should be greater than the number of clusters expected. Each cluster process takes

[1] In this work, 3 cluster processes are generated for each translational process.

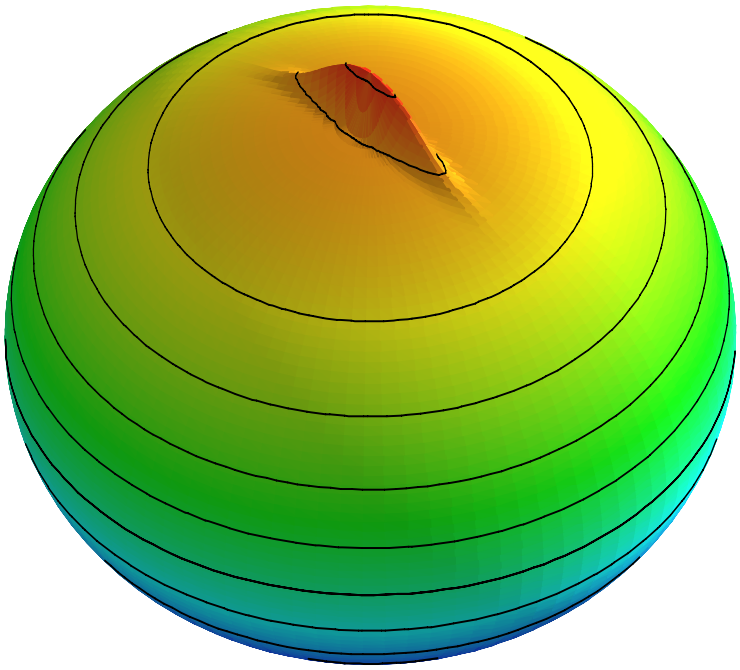


Figure 7.2: **Spherical Gaussian (Rank = 2)** This figure demonstrates the function used for determining the probability that a linear constraint belongs to a particular translational direction and a particular location on the unit sphere.

the translational direction of its spawning great circle as an initial estimate for $\vec{T}_j$. The values of $\vec{L}_j$ are uniformly distributed around the great circle. Estimates for σ_T can also be taken from the spawning translational process, and values for $\sigma_\perp$ are based on the spacing of the $\vec{L}_j$ around the great circle. The values for both σ_T and $\sigma_\perp$ are updated during each M-step based on the population of constraints assigned to the process and their deviation from the values estimated for $\vec{T}_j$ and $\vec{L}_k$. It is now possible to apply the EM algorithm. Note that the number of cluster processes remains fixed until the EM algorithm completes. At that point in time, clusters can be merged based on both their translational directions and location vectors. It is important to note that two clusters with similar location vectors but different translational directions represent two separate clusters: this is what might be expected at the intersection of two great circles which represent distinct translational motions.

Each cluster does not necessarily represent a distinct translational motion. There may be multiple clusters aligned along a great-circle representing a single motion, just as there may be multiple clusters at a single location representing multiple motions. The interpretation of clusters so as to hypothesize about the motions they represent requires further consideration.

7.2 Individual Clusters Give Only Weak Support

Assume a cluster which is characterized by $\vec{T}$, σ_T, $\vec{L}_\perp$ and $\sigma_\perp$. By the definition of $\vec{T}$, it will always be true that $\sigma_T < \sigma_\perp$. It is possible think of a cluster as an ellipse on the surface of the unit sphere: its centre is defined by $\vec{L}$, it's orientation is given by $\vec{T}$, and its eccentricity is defined by σ_T and $\sigma_\perp$ (see Figure 7.3). This cluster may or may not provide strong support for the translational direction $\vec{T}$. For example, consider the case where $\sigma_T \approx \sigma_\perp$. The direction $\vec{L}_\perp$ is as likely a translational direction as is $\vec{T}$, and this cluster only constrains the translation to lie near the plane containing $\vec{T}$ and $\vec{L}_\perp$. On the other hand, if $\sigma_T \ll \sigma_\perp$ then $\vec{T}$ becomes a much more likely translational direction than $\vec{L}_\perp$. Therefore, a cluster of linear constraints becomes more powerful as the ratio $\sigma_\perp/\sigma_T$ increases. It should be noted that the ownership function for clusters

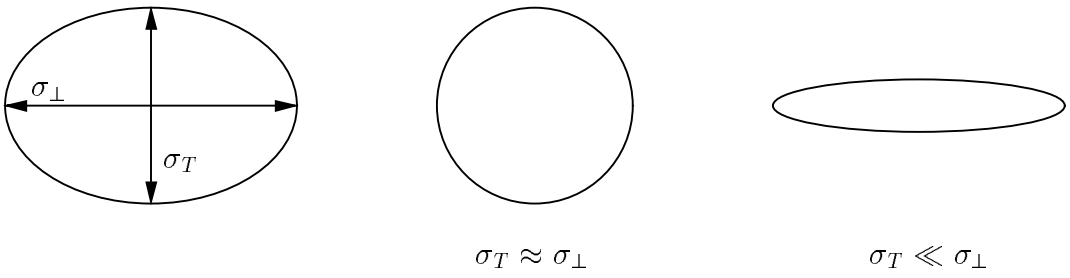


Figure 7.3: This figure shows the ellipses of uncertainty associated with a cluster. This ellipse defines the region associated with one standard-deviation of uncertainty: most constraints belonging to the cluster will fall in this region. a) The variance σ_T defines the width of the minor axis, and that $\sigma_\perp$ defines the width of the major axis. b) When $\sigma_T \approx \sigma_\perp$ there is a roughly circular region of uncertainty, and the cluster does not strongly define a translational direction. c) If $\sigma_T \ll \sigma_\perp$ then the cluster becomes elongated and defines a translational direction much more clearly. As $\sigma_\perp \to \infty$ the cluster becomes a great circle.

given in Eqn 7.1 approaches that for great circles given in Eqn 5.2 as $\sigma_\perp$ increases.

Just because a cluster may not uniquely define a translational direction does not render its information useless. It is possible to combine information from different clusters in order to hypothesize about the underlying translational motions.

7.3 BruteSac

In general one expects to recover a number of motion constraint clusters from an image sequence. As was discussed in the previous section, each individual cluster may not provide strong support for particular translational direction. It therefore becomes necessary to combine the information provided by individual clusters in order to hypothesize about underlying translational motions.

Assume that three clusters, which are all ambiguous about the exact direction of $\vec{T}$, exist. If the locations of the clusters do not all lie on a single great circle it may be necessary to hypothesize 2 or more possible directions of translations. Each pair of clusters gives rise to a different interpretation for translational direction. For example, the first two clusters support the translational direction $\vec{T}_{12} = \vec{L}_1 \times \vec{L}_2$. The ith and jth clusters support the translational direction $\vec{T}_{ij} = \vec{L}_i \times \vec{L}_j$. In the case of three

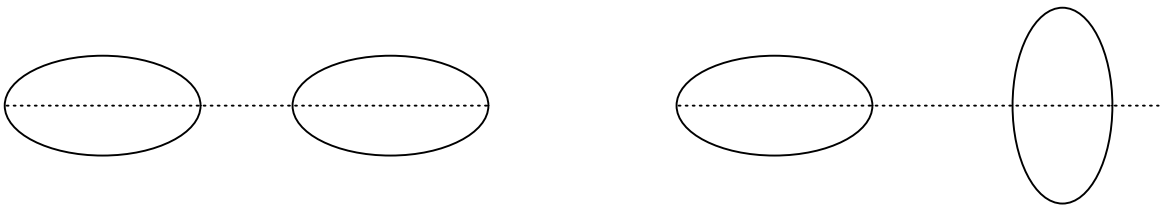
clusters there are $\begin{pmatrix} 3 \\ 2 \end{pmatrix} = 3$ possible interpretations. In general, for n clusters there will be $\begin{pmatrix} n \\ 2 \end{pmatrix} = (n^2 - n)/2$. It is expected to find only a small number of clusters for any image sequence, so evaluating all possible pairs is not too onerous. I term this method of generating all possible motion hypotheses "BruteSac" after the "RanSac" (random sample consensus) algorithm [17] which allows for random selection of data points for clustering.

Each cluster can be thought of as a "super-constraint". Imagine that there are 5,000 constraints clustered around the direction $\vec{L}_1 = [0\ 0\ 1]^T$ and 50 constraints clustered around $\vec{L}_2 = [0\ 1\ 0]^T$. The translational direction can be estimated as $\vec{T}_{12} = \vec{L}_1 \times \vec{L}_2 = [1\ 0\ 0]^T$. In this approach each cluster is given equal weight. Had a proportionally larger weight been given to the cluster with 5,000 constraints, the $\vec{T}$ direction associated with that cluster alone may have been choosen, which may not be correct if the angular spread of constraints around $\vec{L}_1$ is small and $\sigma_{T_1} \approx \sigma_{\perp_1}$.

By combining information from separate motion clusters it should be possible to generate hypotheses about underlying translational motions. If the number of clusters is reasonably small then all possible motions can be estimated. Next, it becomes important to attempt an ordering of the hypothesized motions based on likelihood.

7.3.1 Methods of Rank-Ordering Hypotheses

The hypotheses generated by considering constraint clusters in pairs may be ordered according to their likelihood. Specifically, for each cluster-pair it is possible to compare how well each cluster matches the translational direction suggested by the other. This is illustrated in Figure 7.4. The pair of clusters in 7.4a) are more compatible than those in 7.4b). As a result the pairing in 7.4a) would lead to a preferred interpretation over that from 7.4b). In the event that $\sigma_T \approx \sigma_\perp$ for both clusters in the pair then it is not possible to make such a comparison, since $\vec{T}$ and $\vec{L}_\perp$ become interchangeable: in this case one would likely rank this configuration after pairs similar to



a) b)

Figure 7.4: This figure shows the comparison of pairs of clusters. Each cluster pair can be examined for compatibility: the clusters in a) are very compatible since their major axes are collinear indicating that they have the same preferred translational direction, whereas the clusters in b) are less compatible since their major axes are not collinear.

the case in 7.4a) but perhaps ahead of those in 7.4b).

In order to perform the ranking, note that each of the n clusters will have an associated $\vec{T}_j$ and D_j.[2] One can compute the probability of the ith cluster's translational direction with respect to the jth cluster by computing $p(\vec{T}_i|D_j)$ as in Eqn. 6.3. If one generates the $n \times n$ matrix whose ijth entry is $p(\vec{T}_i|D_j)$, then each column corresponds to one cluster and how the translational directions of other clusters compare to it. A "score" for each cluster can be generated by summing or multiplying the elements of its column in the matrix. It is also possible to generate a log-likelihood value according to

$$l_j = \sum_{i=1}^{n} \log p(\vec{T}_i|D_j)$$

where l_j represents the likelihood of the jth process. This may give a low scoring when multiple motions exist: it may be more productive to consider only the two or three largest entries in a column. Once a score is computed, the hypothesis can be ranked according to score. The method outlined so far considers only hypotheses which coincide with the translational directions computed for at least one cluster. Additional hypotheses can be generated according to $\vec{T} = \vec{L}_i \times \vec{L}_j$ and scored in a similar fashion.

Observe that clusters of linear constraints may be considered pair-wise in order to

[2] Note that the matrix D_j when used for comparing clusters is based solely on the linear constraints and not on the location vectors.

generate hypotheses regarding the translational motions in the scene. These hypotheses can be ranked according to their likelihood for the purposes of further processing. Such processing may be performed by systems that incorporate other information about the world (perhaps from context, perhaps from other sensor modalities).

7.4 Summary

In this chapter clusters of linear constraints have been considered. A *cluster* is defined as a group of constraints in close proximity on the surface of the unit sphere. It was shown that the use of clusters can resolve situations which would fail to be detected by requiring constraints to lie on a great circle alone. Individual constraints may have only a weak preference for a particular translational direction, and therefore it may be necessary to consider the information from a number of constraint clusters in order to hypothesize about the underlying translational motions. Practical examples of constraint clusters are shown in Chapter 8.

Chapter 8

Results from Synthetic Sequence

In this chapter synthetic optic flow containing an IMO is analyzed. This allows testing of the methods proposed for motion segmentation. I discuss how the synthetic optic flow is created, and the results of clustering on the linear constraints. Results from the anisotropic noise correction discussed in Section 4.2.4 are presented and discussed, and finally a discussion of the effects of fixation on the linear constraints is presented. Figure 8.1 shows two depth maps used to calculate synthetic flow according to Eqn. 2.2.

8.1 Methods

8.1.1 Generation of Synthetic Flow

Synthetic optic flow can be generated through the use of a depth-map and Eqn. 2.2. Each point in the depth map, $\vec{x}$, has an associated depth, $X_3(\vec{x})$. One can therefore generate $\vec{u}(\vec{x})$ for each point given a choice for $\vec{T}$ and $\vec{\Omega}$. In order to generate synthetic flow containing an IMO, two depth-maps are used, each with different translational and rotational velocities. Figure 8.2 shows synthetic flow generated with the depth-maps for the office and a cube. The translation of the office relative to the observer is $\vec{T} = [\ 1 \quad 0 \quad 1\ ]^T$. The translation of the cube relative to the observer is $\vec{T} = [0\ 1\ 0]^T$. A rotation has been added to simulate the observer "fixating" a point near the centre

Figure 8.1: On top is a depth-map (Z-buffer) from a computer generated image of an office. Below is a depth-map for a cube. These two depth-maps can be used to generate a synthetic flow field containing an IMO.

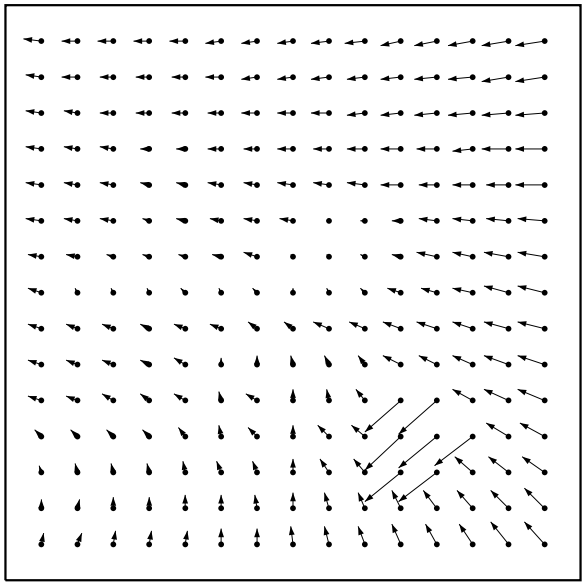


Figure 8.2: This figure shows synthetic optic flow generated from the depth maps shown in Figure 8.1. The observer is moving with a translational velocity of $\vec{T} = [1\ 0\ 1]^T$ with respect to the background. The cube is falling, and has a translational velocity of $\vec{T} = [0\ 1\ 0]^T$. A rotation has been added to simulate the observer fixating a point near the centre of the image.

of the image.[1] The angular extent of the image is taken to be 45°.

In order to make the flow field more realistic, 10% random noise has been added as $\hat{u}(\vec{x}) = \vec{u}(\vec{x}) + \vec{n}$. The noise component $\vec{n}$ was chosen from a 2-D isotropic normal distribution having a standard deviation equal to 10% of $||\vec{u}(\vec{x})||$. The noisy flow field is shown in Figure 8.3.

8.1.2 Generation of Linear Constraint Vectors

The $\vec{\tau}$ constraints are generated according to the convolution method of Jepson & Heeger [46]. A 15×15 convolution mask is created by modifying a *difference-of-Gaussians* (DOG) function so as to satisfy

$$F\vec{c} = 0$$

[1]The point chosen has image coordinates of (69,49) where (0,0) is the upper-left corner. This corresponds to the top-left corner of the back of the chair.

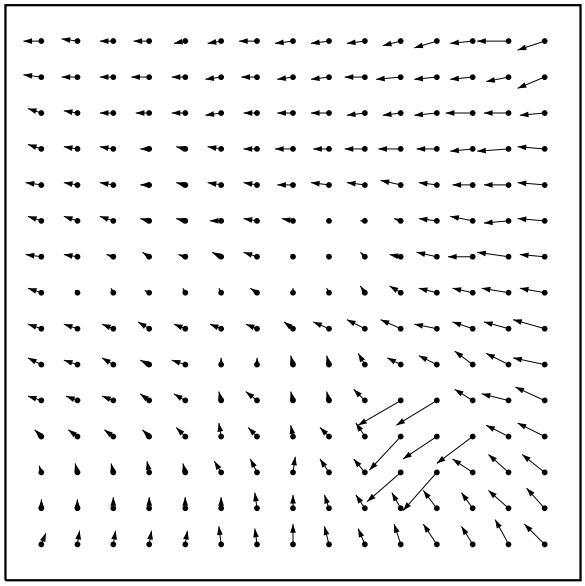


Figure 8.3: The same optic flow field as in Figure 8.2, but with 10% random noise added.

where F is a matrix constructed from the image sampling locations of the convolution mask. The null space of F is invariant under affine transformations of these coordinates, so it is only necessary to generate the coefficients once, and not each time the convolution mask is moved. The DOG is chosen to have a *centre* standard deviation of 1.5 pixels, and a *surround* standard deviation of 3 pixels. The c_k are normalized so that $\sum_{k=1}^{K} c_k^2 = 1$. As each $\vec{\tau}$ is generated, it is thresholded according to its magnitude—those constraints with a SNR less than 5 are discarded. Since the flow is simulated, one can use the following definition of SNR [46],

$$\text{SNR} = \frac{||\vec{\tau}||}{\rho \sigma_u}$$

where

$$\sigma_u^2 = \sum_{k=1}^{K} c_k^2 ||\vec{u}(\vec{x}_k)||^2$$

is a weighted average of the magnitudes of the flow vectors used to construct $\vec{\tau}$. The SNR estimate is based on the assumption of 10% relative noise in the optic flow, *i.e.* $\rho = 0.1$. Figure 8.4 shows the relative magnitude and image location of the recovered constraints. Notice that regions containing depth discontinuities give

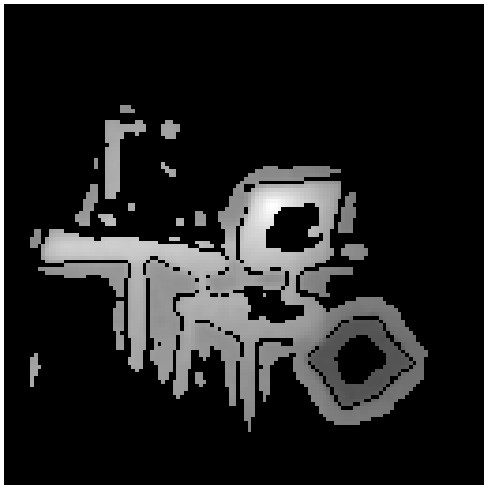


Figure 8.4: This is a plot of the magnitudes of the $\vec{\tau}$'s recovered from Figure 8.2. Regions containing depth-discontinuities give rise to the largest constraints. Constraints having an SNR of less than 5 were removed.

rise to constraints with larger magnitudes. Approximately $3,000$ constraints were recovered for this image.

The constraints, $\{\vec{\tau}_i\}_{i=1}^{N}$, each have an associated image location, $\{\vec{x}^i\}_{i=1}^{N}$. These image locations are derived by a weighted sum of the sampling coordinates for the convolution:

$$\vec{x}^i = \sum_{k=1}^{K} c_k^2 \vec{x}_{ik}$$

where $\{\vec{x}_{ik}\}_{k=1}^{K}$ are the sampling locations of the ith patch. The constraints and their image locations form the input data to the clustering stage.

8.2 Results

8.2.1 Recovery of Processes

In Figure 8.5 the result of segmenting the constraints according to underlying translational direction (great circles on the unit sphere) is shown. The method used to split processes and fit process parameters to the "great-circle" model is as described in Chapter 6. There are 4 processes recovered, but the 1st and 3rd processes are found to be similar, and are merged according to the criteria specified by Eqn. 6.3. The

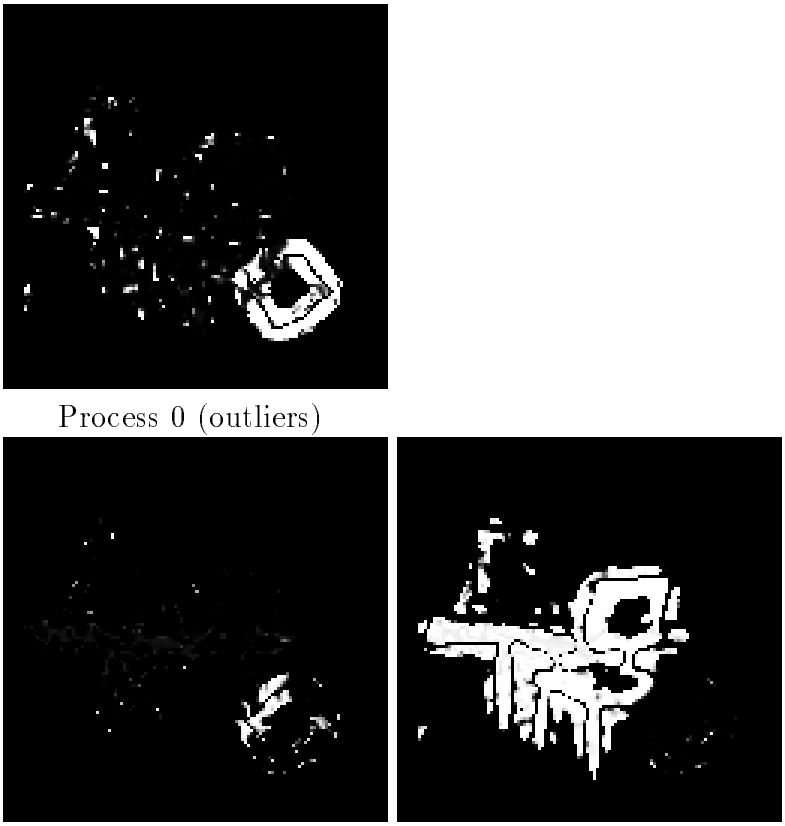


Figure 8.5: This figure shows the segmentation of the $\vec{\tau}$'s based on translational direction. Process 2 clearly belongs to the background motion, whereas Process 1 has garnered support from some of the constraints generated at the cube-background boundary. The outlier population also has considerable support from these constraints, as well as some from the background.

support maps (ownership probabilities) shown in Figure 8.5 show the results after the merging has been performed.

Table 8.1 shows the numerical values associated with the data in Figure 8.5. Including outliers there are 4 processes recovered. Approximately 22.5% of the constraints were marked as outliers, while the remainder were split among 3 processes. The parameters recovered for the first and third processes are shown here prior to being merged. Process 2 represents the background. It owns about 71.5% of the constraints, and has a translational direction that is roughly in the $[\ 1 \quad 0 \quad 1\ ]^T$ direction. The deviation from the expected value is due to a bias, to be discussed later in this chapter.

	Process 0 (outliers)	Process 1	Process 2	Process 3
Mixtures	0.225157	0.009908	0.715241	0.049695
$\vec{T}$		$\begin{bmatrix} -0.3510 \\ -0.0033 \\ -0.9364 \end{bmatrix}$	$\begin{bmatrix} 0.5654 \\ 0.0013 \\ 0.8248 \end{bmatrix}$	$\begin{bmatrix} -0.2146 \\ -0.0222 \\ -0.9764 \end{bmatrix}$
σ		0.0041	0.0632	0.0084

Table 8.1: This table shows the estimated parameters recovered by clustering the linear constraints. Process 2 represents the background constraints. Process 1 and 3 were merged, and represent the cube- background outliers.

In Chapter 7 further breaking of translational processes (great circles) into clusters is considered. For each of the translational processes recovered, initialize 3 clusters evenly spaced around the corresponding great-circle. Therefore, this new segmentation starts with 6 processes (not including the outlier process). Segmentation into clusters, along with merging of like clusters, gives the segmentation shown in Figure 8.6. In addition to outliers, 4 clusters remain after merging is complete.

Table 8.2 shows the results of this further segmentation. Processes 1, 3 and 4 appear to own most of the background constraints, with Process 3 having ownership of 65.8% of the total constraints. It should be noted that the processes discussed here do not have a one-to-one correspondence with those of Figure 8.5 and Table 8.1. As one would expect with this type of cluster, each process has one large singular value (eigenvalue) and two small ones. This indicates that the constraints in each process occupy a small segment of a great circle. Processes 1, 3 and 4 have similar location vectors $\vec{L}$ and translational directions. Process 2 owns constraints belonging to the cube-background boundary. It has a location vector significantly different to that from the background constraints, suggesting that these constraints occupy a different location on the unit sphere. While the recovered translational direction is not that of the cube,

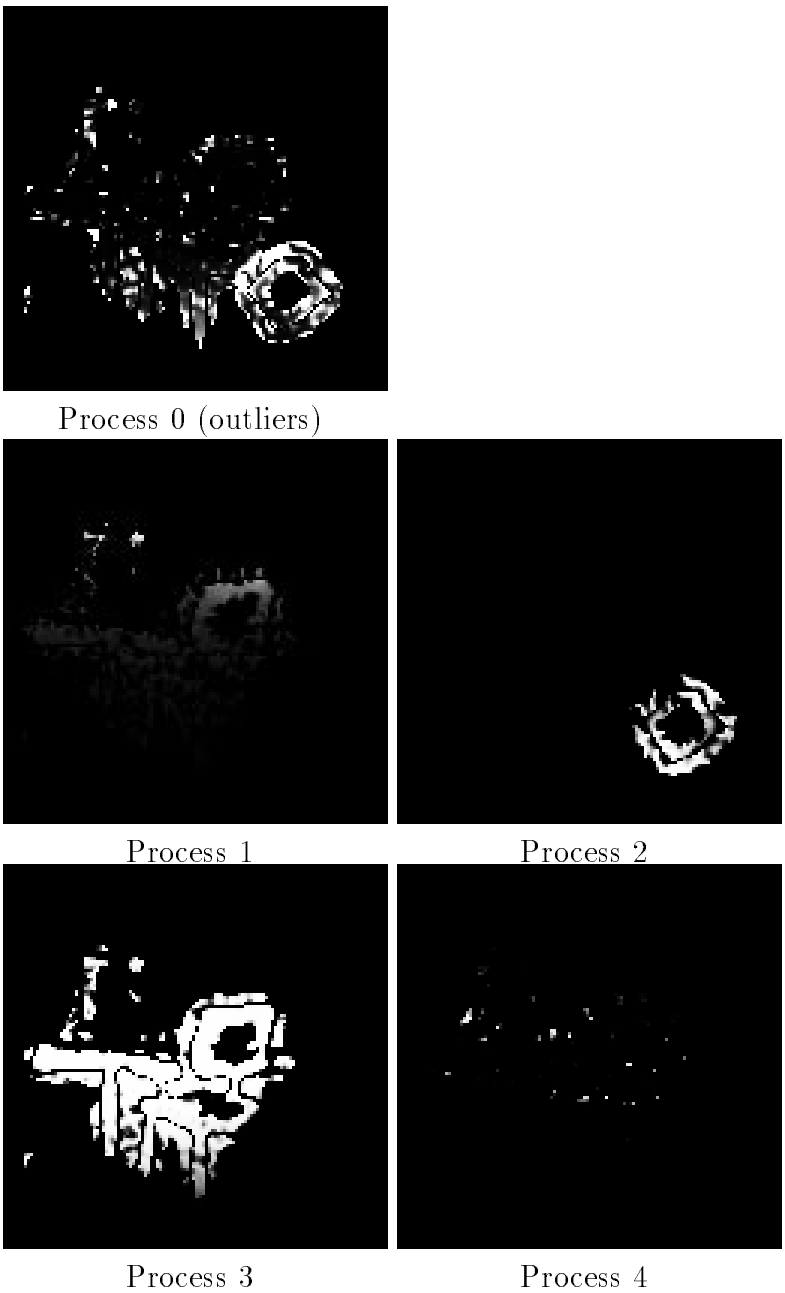


Figure 8.6: This figure shows the segmentation of the $\vec{\tau}$'s based on subdividing the great-circles from Figure 8.5. Process 2 is a stronger segmentation of the cube. Process 3 represents the background, with some support from the background constraints also going to Processes 1 and 4.

	Process 0 (outliers)	**Process 1**	**Process 2**	**Process 3**	**Process 4**
Mixtures	0.232701	0.000086	0.098971	0.658103	0.010140
$\vec{T}$		$\begin{bmatrix} 0.7468 \\ 0.0062 \\ 0.6650 \end{bmatrix}$	$\begin{bmatrix} 0.3213 \\ -0.0659 \\ 0.9447 \end{bmatrix}$	$\begin{bmatrix} 0.6008 \\ 0.0027 \\ 0.7994 \end{bmatrix}$	$\begin{bmatrix} 0.8964 \\ 0.2380 \\ 0.3738 \end{bmatrix}$
$\vec{L}$		$\begin{bmatrix} -0.0729 \\ 0.9947 \\ 0.0727 \end{bmatrix}$	$\begin{bmatrix} -0.8522 \\ -0.4552 \\ 0.2581 \end{bmatrix}$	$\begin{bmatrix} -0.0214 \\ 0.9997 \\ 0.0127 \end{bmatrix}$	$\begin{bmatrix} -0.2590 \\ 0.9659 \\ 0.0061 \end{bmatrix}$
Singular Values		0.9924316 0.0071166 0.0004508	0.9957408 0.0031297 0.0011288	0.9825196 0.0153044 0.0021774	0.9958853 0.0023045 0.0018101
σ		0.1512	0.0477	0.1598	0.0527

Table 8.2: This table shows the results from segmenting the translation processes reported in Table 8.1. Each cluster has one large singular value and two small ones—this is characteristic of this type of cluster. Process 1 and 3 represent the background, and they have similar location vectors, $\vec{L}$. Process 2 represents the cube, and has a very different location vector. The recovered translational direction for Processes 1 and 3 is close to what is expected.

namely $\vec{T} = [\ 0 \quad 1 \quad 0\ ]^T$, note that after correction for the anisotropic noise (next section) the recovered translational direction, $\vec{T} = [\ -0.5221 \quad 0.7080 \quad -0.4755\ ]^T$, has a strong vertical component. This corrected translation appears to be *very* close to $[\ 0 \quad 1 \quad 0\ ]^T - [\ 1 \quad 0 \quad 1\ ] \rightarrow [\ -0.5 \quad 0.7071 \quad -0.5\ ]$, as predicted in Chapter 4.

8.2.2 Anisotropic Noise Fix

It has been observed that linear methods for recovering translational direction suffer from a significant bias [46, 47, 48, 19]. The bias is caused by the anisotropic nature of the noise in the linear constraint vectors, as discussed in Section 4.2.4. Recovered translational directions are biased in the direction of the optical axis ($\hat{z}$). Methods for

the removal of this bias have been proposed [48, 19]. These involve making an estimate of the covariance matrix and subtracting it prior to recovering the translational direction. This requires knowledge of the scaling of the covariance matrix as well as its structure. As discussed in Chapter 4 the approach used involves estimating the covariance matrix (up to a scale factor) and using it to rescale the constraints. The rescaling can be performed directly on the D matrix, to give

$$\hat{D} = C^{-1/2} D C^{-1/2} \ .$$

Now estimate the minimum eigenvalue and use the associated eigendirection as our estimate, $\hat{T}$. Next, reverse the scaling to give

$$\vec{T} = C^{-1/2} \hat{T} \ .$$

It now remains to decide how to estimate the covariance matrix. Each constraint has a covariance matrix defined by

$$C_i = \sigma^2 \sum_{k=1}^{K} c_k^2 \begin{bmatrix} 1 & 0 & -x_k \\ 0 & 1 & -y_k \\ -x_k & -y_k & x_k^2 + y_k^2 \end{bmatrix}$$

where $\{\vec{x}_k\}_{k=1}^{K}$ are the flow-sampling points for generating the constraint, and $\vec{x}_k = [\ x_k \quad y_k \quad 1\ ]^T$. Assume that σ^2 is the same for all the constraints, and ignore it in subsequent estimates of covariance matrices.[2]

Two methods are used for estimating the covariance matrix. The first method generates a covariance based on the average location of the constraints for a given process. As mentioned previously, the "average" location of a $\vec{\tau}$ constraint is given by

$$\vec{x}^i \ = \sum_{k=1}^{K} c_k^2 \vec{x}_k \ .$$

[2]This is, strictly speaking, not true. However, since we are most interested in the form of the covariance and not the scaling, it is a reasonable approach.

	True	Uncorrected	Method 1	Method 2
$\vec{T}$	$\begin{bmatrix} 0.7071 \\ 0.0000 \\ 0.7071 \end{bmatrix}$	$\begin{bmatrix} 0.6316 \\ 0.0005 \\ 0.7737 \end{bmatrix}$	$\begin{bmatrix} 0.7236 \\ 0.0037 \\ 0.6890 \end{bmatrix}$	$\begin{bmatrix} 0.7196 \\ 0.0035 \\ 0.6932 \end{bmatrix}$
error	0.0°	5.7523°	1.4172°	1.0907°

Table 8.3: This table shows the results of correcting for the anisotropic nature of the noise on the estimated translational direction. The results are tabulated over 5 trials, each of which uses a different seed to the random number generator to add noise to the optic flow. Both Method 1 and Method 2 provide considerable improvement over the uncorrected case.

It is then possible to generate an estimate for the form of the covariance matrix as

$$C^j = \sum_{i=1}^{N} s_{ij} \begin{bmatrix} 1 & 0 & -x^i \\ 0 & 1 & -y^i \\ -x^i & -y^i & (x^i)^2 + (y^i)^2 \end{bmatrix} .$$

The second method generates a covariance matrix based on a weighted average of the covariance matrices for individual constraints within the process. The covariance matrix for the jth process is estimated as

$$C^j = \frac{\sum_{i=1}^{N} s_{ij} C_i}{\sum_{i=1}^{N} s_{ij}} .$$

Table 8.3 shows the results from applying a rescaling for both methods of generating the covariance matrix. Both methods for estimating the covariance matrix give considerable correction to the data. The second method appears superior, since it is possible to come up with an estimate for the covariance matrix for each $\vec{\tau}$ constraint.

8.3 Effect of Fixating the Background

Fixation of the background has an important impact on the problem of motion segmentation. Figures 8.4 and 8.7 show the magnitudes and image locations of the

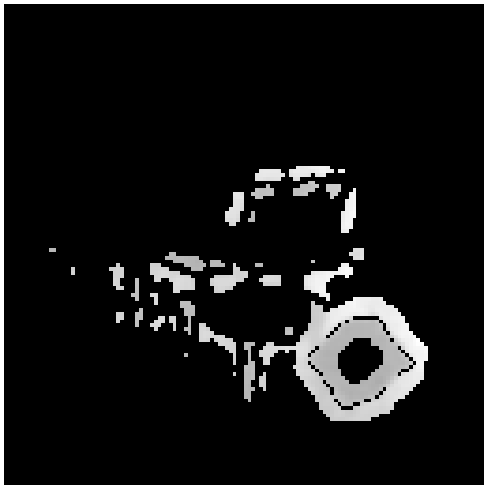


Figure 8.7: This is a plot of the magnitudes of the $\vec{\tau}$'s recovered when no background point was fixated. When compared with Figure 8.4 it is observed that fewer constraints made the SNR cut-off level. The addition of a rotation to fixate the background has improved the SNR of the recovered constraints.

constraint vectors for the cases where the background is and is not (respectively) fixated. When the background is fixated, more constraints are present. This can be interpreted as more constraints having a large enough SNR to meet the thresholding condition. The effect of noise on the background constraints is therefore diminished by fixating a point in the background. The top-left corner of the back of the chair is the fixation point, and its constraints also have the highest SNRs. Fixating the background makes the detection of independent object motion easier by making the background constraints cluster more tightly, thus making the separation of object-outlier constraints easier. Trials in which different background points were fixated all gave reliable segmentation and egomotion estimation. Figure 8.8 shows the (noisy) optic flow associated with the case of no fixation.

The improvement in SNR with fixation can be understood by observing that the rotation by which fixation is achieved is designed to reduce the optic flow to zero at the point of fixation. In the neighbourhood of this point the magnitude of the flow may still be non-zero, but it will be reduced. The process of fixation can therefore be thought of as reducing the dynamic range of flow magnitudes. If the associated noise process is proportional to flow magnitude, then the noise will also be reduced. But

the rotation does not affect the information required to recover the linear constraints, namely the flow variation due to the translational component of motion and depth variation in the scene. As a result a higher SNR is obtained. It should be noted that the choice of which scene point to fixate is of importance. For example, if a point $\vec{P}$ at infinite distance is fixated, the required rotation is

$$\vec{\Omega}_f = -\frac{\vec{T} \times \vec{P}}{||\vec{P}||^2} = 0 \ .$$

It therefore is necessary to choose a fixation point in the foreground of the scene.

The effect of improved SNR through fixation is *completely* dependent on assuming a relative noise model for the optic flow. If the noise model is independent of flow magnitudes, no improvement in SNR of linear constraints can be expected through fixation. In this chapter a relative noise model has been used in which the average value of the noise added is a function of the magnitude of the flow vector to which the noise is added. Fleet [24] suggests that this relative noise model is good for a range of optic flow magnitudes. This can be seen by considering the fact that estimating optic flow can be reformulated as the problem of estimating the tilt of a plane in frequency space [24, 35]. If the plane's tilt is estimated with constant angular error, then the actual tilt (slope) error is relative to the magnitude of the tilt. To see how this affects the SNR of the linear constraints, note that fixation reduces the magnitude of the flow vector, and hence the associated noise. On the other hand, the "signal" does not decrease since it is entirely conveyed in the translational part of the image motion, not in the rotational part. However, Fleet's analysis is tied to a particular method for computing flow. Constant noise models are more appropriate for recent optic flow algorithms which use shifting to reduce residual flow and hence improve flow estimates. Shifting assumes the initial estimate for flow is reasonably accurate. The problem then becomes that of estimating a residual flow which has magnitude near zero. The error can be (iteratively) reduced to the point where the residual flow has constant error, *i.e.* the dependence on the flow magnitude is eliminated.

Any system that performs fixation during egomotion and assumes a relative noise

model has the potential to improve the recovery of constraints which can be used in turn to recover egomotion and perform IMO detection. There is evidence to suggest that the relative noise model may be applicable to human vision [63]. Many creatures can achieve fixation through eye movements (eye movements can be thought of as rotations around the origin of an observer-centred coordinate system). The opto-kinetic and vestibulo-ocular reflexes (OKR and VOR respectively) are eye-movements that serve to keep the image of the world stabilized while a human observer moves through it [11]. The OKR stabilizes images by tracking points of interest, and when necessary acquires a new tracking target through a quick movement called a *saccade*. The opto-kinetic reflex may in fact have a purpose slightly different from that which is commonly believed. While its ability to stabilize images on the retina is obviously beneficial to vision as a whole, it may be that it has a greater importance in terms of its effect on enhancing detection of IMOs. (In evolutionary terms, the IMO is the entity most likely to want to eat the observer, or be eaten by the observer.) Nelson [67] suggests that

> "A reasonable heuristic for interaction with the real world is if it is moving, you should probably pay attention."

It has been proposed that saccades are *time-optimal* [60, 20, 55, 56], which would serve to minimize disruption of the fixation phase of OKR. This would be an effective strategy from the evolutionary point of view.

The VOR does not directly achieve fixation, as it serves mainly to cancel the effects of head rotation by counter-rotating the eyes. However, there exists a counterpart called the *linear vestibulo-ocular reflex* (LVOR) which serves to fixate objects during linear accelerations of the head [69, 70, 71]. These eye-movement reflexes may serve to reduce the dynamic range of image motion on the retina, thereby making it more sensitive to subtle variations which code the information required by the motion interpretation centres in the brain.

Ballard [4] has previously suggested that fixation is an important component in an active vision paradigm which seeks to reduce computational complexity of certain visual tasks, such as recovering depth structure through motion parallax. His analysis

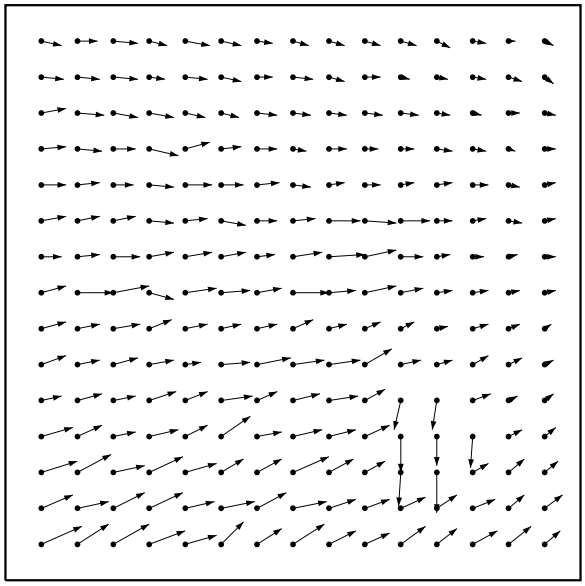


Figure 8.8: This plot shows the optic flow field for the case of Figure 8.3, but without fixation (*i.e.* pure translation).

is quite different in that it is not based on an analysis of the effects of noise, but rather whether an exocentric reference frame can be used to reduce the computational load.

8.4 Summary

This chapter has presented results from a synthetic flow sequence containing an IMO. The advantage of such a sequence is that it is possible to know the exact relative motions between the observer, the background and the IMO. The agreement between these values and those recovered by the segmentation process based on linear constraints is quite good. This sequence has allowed validation of the derivation of what happens when linear constraints are generated across IMO boundaries. Constraints segmented using great circles were further segmented into clusters, thereby improving the segmentation. This sequence also allows validation of the method for removing the bias inherent in the linear method for recovering translational direction. Finally, a discussion of the effect of fixating background points on segmentation is presented. It is suggested that this fixation improves our ability to perform segmentation of IMOs, and may provide a rationale for some eye-movement reflexes observed in biological

vision systems.

The next chapter presents a real image sequence which is segmented using linear and bilinear constraints.

Chapter 9

Results from Forklift Sequence

This chapter shows the results from applying motion segmentation to a real image sequence. The sequence analyzed consists of 10 frames, each 640×480 pixels in size. The sequence was captured from a video camera mounted on a robot translating roughly along the optical axis of the camera. The robot's speed was not measured, but was the equivalent of a fast walk. There is one IMO in the scene: a forklift moves from left-to-right across the robot's path, at speeds of up to 50 pixels/frame. The industrial environment is quite irregular but provides good texture for optic flow recovery, and good depth structure for the generation of $\vec{\tau}$ constraints. Figure 9.1 shows a frame from the sequence. Subsequent sections will outline the methods of analysis for this sequence and the subsequent results.

9.1 Methods

9.1.1 Recovery of Affine/Rational Flow Patches

Optic flow for this sequence was generated by clustering constraints that are consistent with either constant, affine or rational models of flow [40]. The constant model for flow is given by

$$\vec{u}(\vec{x}) = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} . \tag{9.1}$$

Figure 9.1: This is a frame from a sequence (of 10 frames) collected by a robot-observer translating roughly along the optical axis in an industrial environment. The forklift and its driver are translating to the right. The boxes indicate image regions for which affine or rational models for optic flow have been fitted. The focus-of-expansion (FOE) of the background motion for each frame in the sequence has been indicated by a '$\times$' (see Section 9.2)

The constant model for flow is useful for fitting small regions of flow in the image. The affine model for flow is

$$\vec{u}(\vec{x}) = \begin{bmatrix} \alpha_0 + \alpha_2 x_1 + \alpha_4 x_2 \\ \alpha_1 + \alpha_3 x_1 + \alpha_5 x_2 \end{bmatrix} \tag{9.2}$$

recalling that $\vec{x} = [\ x_1 \quad x_2 \quad 1\ ]^T$. This model allows the flow to vary in a linear fashion over the patch. The rational model provides the position $\vec{x}'$ of the displaced point as a function of the original position, $\vec{x}$.

$$\vec{x}'(\vec{x}) = \frac{1}{\alpha_6 x_1 + \alpha_7 x_2 + 1} \begin{bmatrix} \alpha_0 + \alpha_2 x_1 + \alpha_4 x_2 \\ \alpha_1 + \alpha_3 x_1 + \alpha_5 x_2 \end{bmatrix} . \tag{9.3}$$

The rational model adds two parameters to the affine model, and is an exact representation of the displacement of image points from a planar surface undergoing rigid motion [21]. Since the time period between frame samples is small in this sequence,[1] displacement can be used to provide a good model for estimating flow:

$$\vec{u}(\vec{x}) \approx \vec{x}'(\vec{x}) - \vec{x} . \tag{9.4}$$

The rational model is particularly useful for tracking surfaces that are known to be planar, such as the floor [40].

Constraints clustered were component velocities recovered by tracking of contours of constant phase [24, 25, 26]. The integration of these constraints was accomplished by the application of the EM-algorithm to solve for ownership and the parameters for each image region (patch) [40, 43].

The boxes in Figure 9.1 show the 6 patches of optic flow recovered for this image sequence. (The sixth box appears as a single line through the middle of the image—it encompasses the lower portion of the image and is the entire width of the image. The left, right and bottom boundaries are not apparent.) The boundary of each box is

[1] The frame rate was about 15 frames per second, which is small relative to the velocity of the observer.

a measure of the region over which constraints for a given affine or rational model were found. Notice that boxes may overlap, since there may be component velocity constraints belonging to more than one patch in a given image region. To recover a flow estimate for a particular patch, one need only apply the appropriate equation from Eqns. 9.1, 9.2 and 9.4 using the parameters for the specified patch and a value for $\vec{x}$ that lies within the bounding box for that patch. The patch for the floor is modelled with the rational model. The patches for the moving forklift, the stationary forklift, the pillar, the back wall, and the mock-up windows (top-left of image) are all modelled using the affine model. For each frame of the image sequence a set of affine or rational parameters are computed for each patch. As well, the boundaries of each patch (with the exception of the floor patch) are updated. Figure 9.2 shows recovered flow samples from the frame shown in Figure 9.1.

While it may seem that these patches have already segmented the image, it is over-segmented with respect to independent object motion.

9.1.2 Generation and Clustering of Constraints

Samples of optic flow from the patches modelled are then used to generate the linear constraints used for clustering. Six flow samples were generated for each patch by sampling the flow model at the four corners of each patch as well as two interior points. Flow samples from two patches are required to generate a linear constraint, due to the fact that flow from a single affine patch is insufficient to generate a linear constraint (see Section 4.2.2). Let $\{\vec{x}_k\}_{k=1}^{12}$, where $\vec{x}_k = [\ x_{k,1} \quad x_{k,2} \quad 1\ ]^T$, be the sampling locations over a pair of patches. Choose $\vec{c}$ such that $\vec{c} \in \text{null}(F)$ where

$$F = \begin{bmatrix} 1 & \dots & 1 \\ x_{1,1} & & x_{12,1} \\ x_{1,2} & & x_{12,2} \\ x_{1,1}^2 & & x_{12,1}^2 \\ x_{1,1}x_{1,2} & & x_{12,1}x_{12,2} \\ x_{1,2}^2 & \dots & x_{12,2}^2 \end{bmatrix}.$$

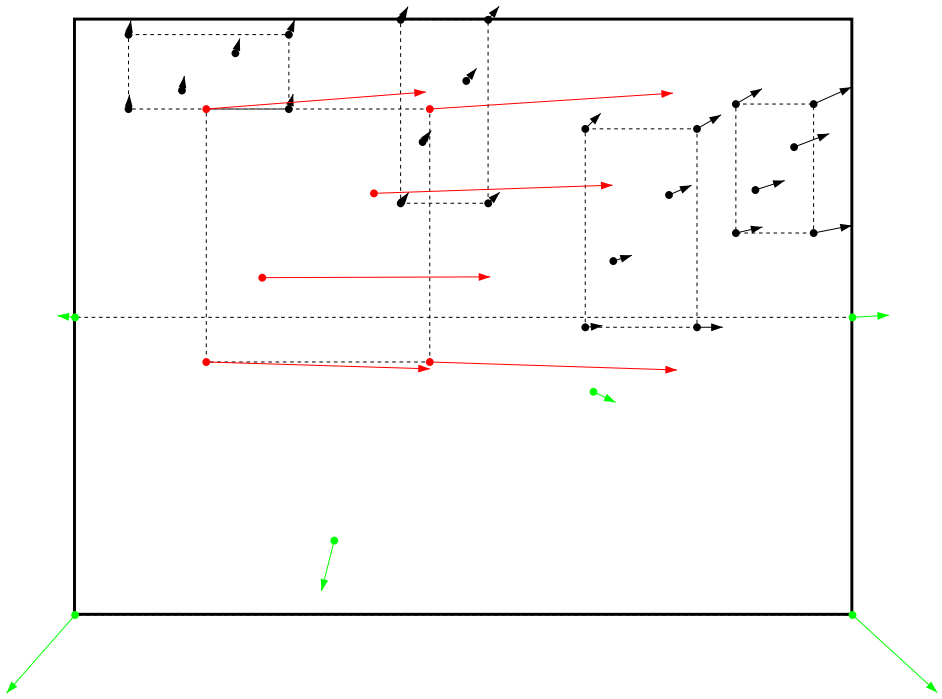


Figure 9.2: This figure shows the optic flow recovered from the frame shown in Figure 9.1. Each dashed box indicates a patch described by an affine or rational flow model. For each box six samples of the flow have been plotted, according to the model used to recover that patch. The green flow vectors are from the floor patch. The red flow vectors are from the moving forklift.

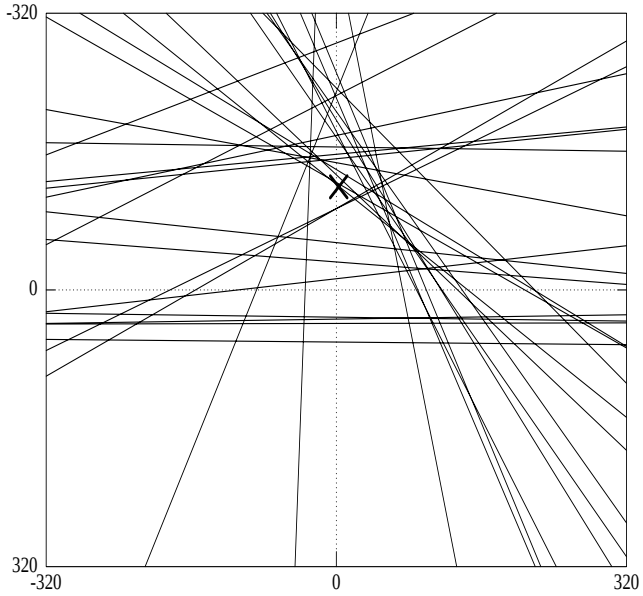


Figure 9.3: The linear translation constraints plotted in the image space for the frame shown in Figure 9.1. Each constraint defines a plane through the origin, which is shown intersecting the image plane $X_3 = f$. Units are shown in pixels. The '$\times$' shown is the FOE for the background motion after clustering the linear constraints. The FOE for the moving forklift is to the extreme left and does not appear in this figure.

Assuming that F is of full rank (one can choose the $\vec{x}_k$ to make this true), there will be 6 such $\vec{c}$ that satisfy this requirement. Finding the $\vec{c}$'s can be accomplished by using a singular-value decomposition to find null(F) [72]. The results are discussed below in Section 9.2.

Each patch is paired with the floor patch and a set of 6 linear constraints are computed for each pair. A plot showing the recovered linear constraints is shown in Figure 9.3. All constraints are then clustered according to the methods outlined in Chapter 6.

It may seem that the floor patch is getting special treatment here, but it is not unreasonable to consider that the floor (or ground) plane would be tracked separately by a robot observer, since the assumption of a planar surface to travel on, such as a road, sidewalk, or floor, can be accurately tracked through image analysis [40]. However, the segmentation method will work even if the patches are paired at random.

Clustering the linear constraints determines the number of motion processes and generates an estimate for the translational direction of each. The next step is to generate the bilinear constraints. Since each flow sample generates a single constraint, there will be 6 constraints for each patch. (This is in contrast to 6 linear constraints for each patch-pair.) The same flow sampling locations are used for the bilinear constraints as for the linear constraints. Clustering of the bilinear constraints proceeds as outlined in Chapter 6. Results from clustering the bilinear constraints are given in Section 9.2.

9.2 Results

Linear constraints were generated from the affine- and rational-flow parameters as described in the previous section. The recovered linear constraints are shown in Figure 9.3. The linear constraints were clustered using the great-circle model described in Chapter 6. The number of processes was estimated according to the splitting algorithm described in that chapter. The clustering of linear constraints resulted in two motions being identified. The first motion process was directed along the optical axis, and the second process was roughly along the horizontal axis. In Table 9.1 the section entitled "Initial Guesses" contains the estimates of translational direction as found by clustering the linear constraints. No correction for noise-anisotropy has been performed here.

The linear constraints generated by combining flow samples from the moving forklift and floor patches will be strongly biased by the motion of the forklift. The flow samples from the floor patch are relatively small in magnitude compared to those from the moving forklift patch. The discussion in Section 4.5.1 leads to an expectation that these linear constraints will strongly indicate a motion along the $\hat{x}$-axis, despite the fact that these constraints are generated across the boundary of an IMO. From Table 9.1 one sees that the initial guess for $\vec{T}_2$ is, in fact, almost entirely in the $\hat{x}$ direction.

Figure 9.4 shows the bilinear constraints for each of the two recovered motion

Initial guesses:
Process 1: $\vec{T}$ = [-0.0002 -0.0925 0.9957] $f\vec{\Omega}$ = [0.33 7.98 4.69] $\vec{\sigma}$ = 3.48084
Process 2: $\vec{T}$ = [-0.9948 0.0216 0.0996] $f\vec{\Omega}$ = [-2.94 -99.55 -6.55] $\vec{\sigma}$ = 0.64782
Final Results:
Mixtures: 0.1866 0.7075 0.1059
Process 1: $\vec{T}$ = [0.0102 -0.0925 0.9957] $f\vec{\Omega}$ = [2.09 2.27 -0.10] $\vec{\sigma}$ = 0.07033
Process 2: $\vec{T}$ = [-0.9948 0.0295 0.0972] $f\vec{\Omega}$ = [-4.13 -99.26 -5.18] $\vec{\sigma}$ = 0.05602
Process 1: FOE = (13.20, -119.24)
Process 2: FOE = (-13137.98, 390.18)

Table 9.1: Results from fitting the linear constraints (initial guesses) and bilinear constraints (final results) for the frame shown in Figure 9.1. The units for $f\vec{\Omega}$ are pixels/second.

processes. For motion process 1 the constraints intersect, giving an estimate of the FOE. One sees the improved sense of intersection of these constraints as compared to Figure 9.3. The refined estimates of $\vec{T}_1$ and $\vec{\Omega}_1$ have therefore improved the estimate for the FOE. Figure 9.1 shows the estimated FOEs for all 10 frames. Their location in the image suggests that the camera was aimed slightly to the left and below the direction of travel, *i.e.* the FOE lies above and to the right of the optical ($\hat{z}$) axis.

The constraints as seen by motion process 2 appear as horizontal lines when projected onto the $x_3 = f$ plane. If these were projected onto the plane x_1 = constant another intersection of constraints would be seen, analogous to an FOE. Again, the refinement of the estimates for $\vec{T}_2$ and $\vec{\Omega}_2$ has clarified the translational direction indicated by the corresponding constraints.

Figure 9.5 shows the ownership probabilities for each bilinear constraint with respect to each process. From this figure motion process 1 appears to "own" the constraints from the floor, stationary forklift, pillar, back wall and mockup window patches. Motion process 2 appears to "own" the constraints generated by the moving forklift patch. This makes it possible to conclude that the motion of the moving forklift relative to the observer is different from that of the rest of the environment. It is worth noting that bilinear constraint #1, from the floor patch, is incorrectly identified as belonging to the second motion process (moving forklift). If one refers to Figure 9.2, then this constraint represents the flow sample from the top-right corner of the floor patch. It is not difficult to see that the direction of this flow vector

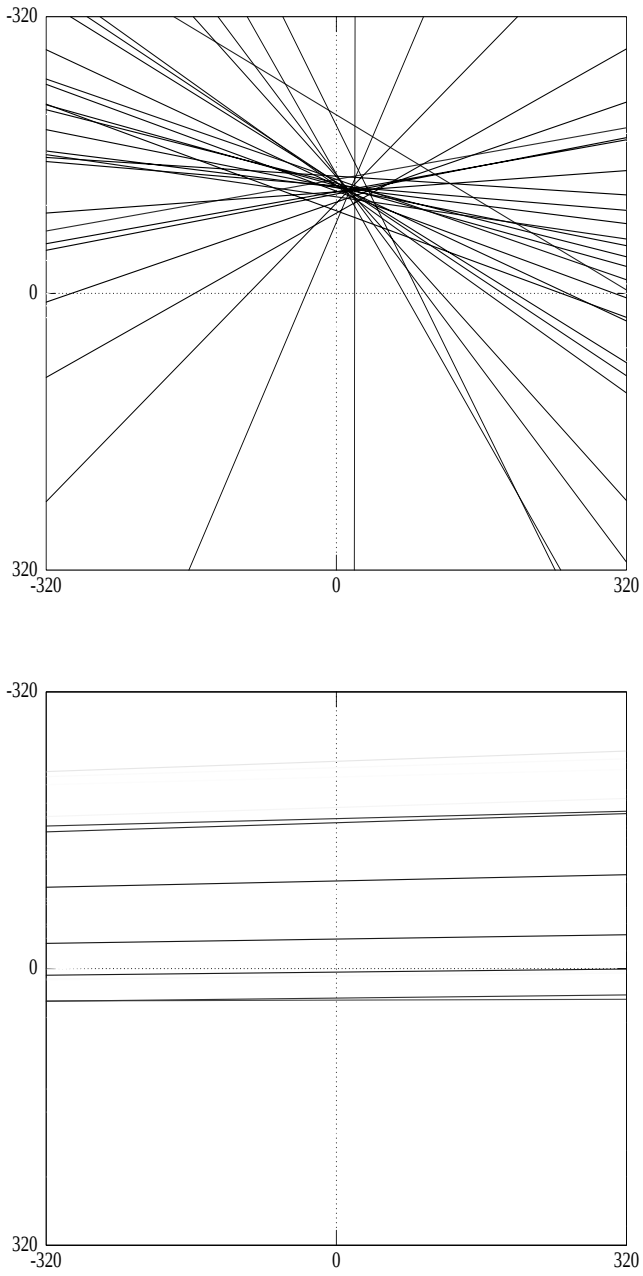


Figure 9.4: The top plot shows the translation constraints derived from the bilinear constraints as seen by the first motion. On the bottom is the plot of constraints as seen by the second motion. As seen in Figure 9.5, the second motion primarily owns constraints generated from the moving forklift. Grey-level indicates ownership probability for each motion.

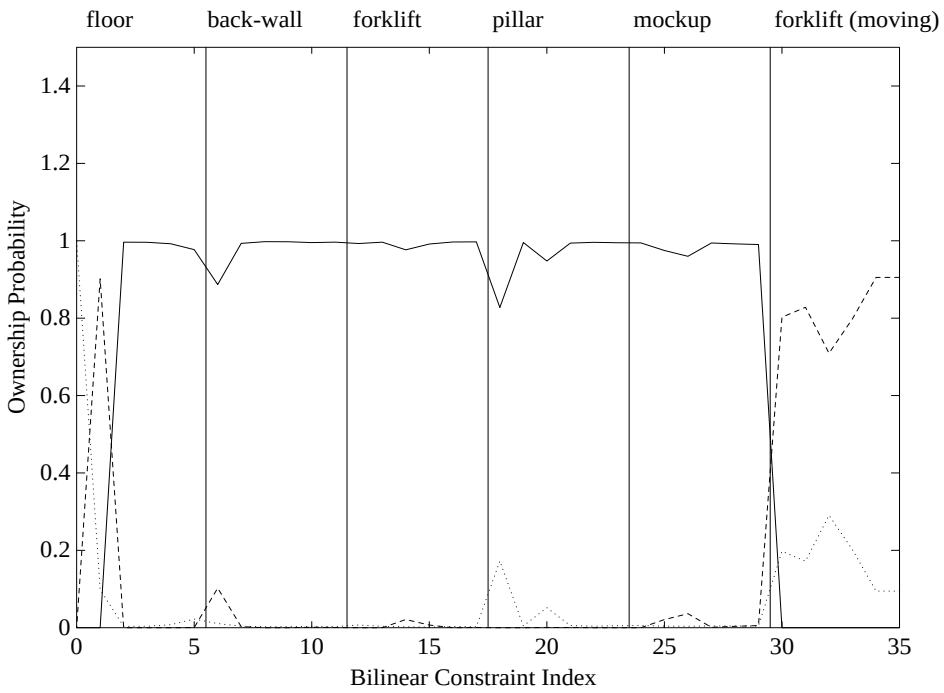


Figure 9.5: The ownership probabilities for the bilinear constraints as fitted by the EM algorithm. The solid line indicates ownership by the first motion, the dashed line the second motion, and the dotted line the outlier process. Bilinear constraint #1 (from the floor patch) probably shows a strong ownership by the second motion process because its horizontal motion is consistent with that of the moving forklift.

is consistent with those from the moving forklift, and could easily be identified as belonging to the second motion process. In this case, the grouping of the constraints into patches is known *a priori*, but remember that some constraints may be shared by more than one process, or even assigned to an incorrect process. Observe that the flow sample from the top-left corner was marked as an outlier, so there may be some error in the recovered flow at these extremes of the floor patch. In general the segmentation of constraints has done a good job of representing the separate motions present in the image sequence. Compelling information has been presented that identifies the forklift as an IMO.

9.2.1 Depth Estimation

For recovering depth estimates, the flow from each patch was fitted to a planar model, $f/X_3 = \hat{n}^T \vec{x}$. Remember that the third component of $\vec{x}$ is one. To achieve this fit it is necessary to solve 6 equations in 3 unknowns:

$$\begin{bmatrix} \vec{x}_1^T \\ \vdots \\ \vec{x}_6^T \end{bmatrix} \hat{n} = \begin{bmatrix} \frac{f}{X_3(\vec{x}_1)} \\ \vdots \\ \frac{f}{X_3(\vec{x}_6)} \end{bmatrix} .$$

This is easily solved using standard algorithms for singular-value decomposition and a back-substitution method that takes advantage of the knowledge of the singular values to find a "solution" to the system of equations [72]. Since there are more equations than unknowns, there may not be a solution, and if there is a solution it may not be unique. It is possible to pick a value for $\hat{n}$ which in the first case minimizes the residual error and in the second case returns a minimum norm solution, and the back-substitution algorithm used does exactly this. The fit is expected to be good for rational patches since they are already based on the rigid motion of planar surfaces, but may not be as good for affine patches. Figure 9.6 shows the relative inverse-depth values recovered for the floor, pillar, stationary forklift, back wall, and mockup windows. The recovered values for the moving forklift using the motion parameters for process 1 were consistently negative over the sequence, and are not shown. Since it is assumed that anything the camera can "see" has a positive depth, this gives us additional evidence for the conclusion that the moving forklift is moving independently.

9.3 Summary of Forklift Sequence Results

It has been shown that the IMO in this sequence was correctly identified by the motion segmentation algorithm. The estimates of the egomotion return reasonable results. We also get a good estimate of the direction of travel for the moving forklift. If the

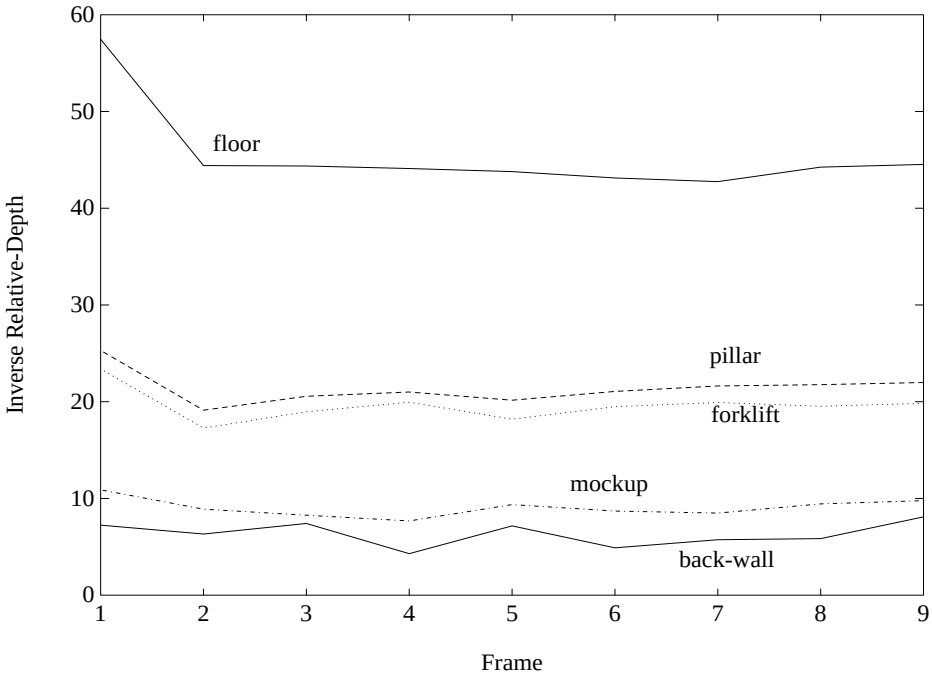


Figure 9.6: The recovered inverse-depth values for each patch over the sequence are plotted. The moving forklift is excluded, as depth values recovered for it are not valid. Note that the relative distances to various objects are correct.

forklift had been moving more slowly, we would not have done as well in recovering the direction of its motion. Recovered relative inverse-depth estimates help support the conclusion that the moving forklift is an IMO, as well as provide information about the structure in the scene.

Chapter 10

Results from JQ Sequence

This chapter presents the results from applying motion segmentation to another image sequence. This sequence consists of 2 frames, each 360×240 pixels in size. The sequence involves egomotion, although the "ground-truth" value of this motion is unknown. There is one IMO in the scene: a man moves from right to left. The optic flow from this sequence was generated as a dense flow field which was then subsampled in order to generate linear and bilinear constraints. One frame from the sequence is shown in Figure 10.1, and the associated dense optic flow field is shown in Figure 10.2.

10.1 Methods

The flow shown in Figure 10.2 was generated according to a method by Black & Jepson [8]. First a coarse estimate of the flow field is made. This estimate can be quite inaccurate and does a poor job of representing motion boundaries such as occluding surfaces at depth discontinuities. Image brightness is then used to identify image regions which may have similar motion—this motion is assumed to be consistent with moving planar surfaces. Flow in these regions is fit to a parametric model for rigid planar motion using a robust estimation scheme. Finally, deformations from the planar surface model are allowed to compensate for cases where the model doesn't fit the image data well enough. (If this error gets too large, the data point is rejected as

Figure 10.1: Shown is a frame from the JQ sequence. The man in the centre of the image is an IMO. While the rest of the environment is considered stationary, the camera is undergoing egomotion.

an outlier.) The result is a dense flow field with good overall accuracy.

In order to keep the number of bilinear constraints reasonable, the dense flow field was subsampled. A 10×10 grid of patches was superimposed over the image, and 6 points sampled from each patch using the same sampling geometry as for patches in Chapter 9. The subsampling used gave 600 bilinear constraints out of a total possible number of 86,400. The sampled flow can be seen in Figure 10.3. Linear constraints were generated for each patch by randomly choosing one other patch in the image and pairing the two as described in Chapter 9. Once the sample points were chosen and linear constraints generated, linear clustering took place as in the previous chapter. The field-of-view was assumed to be 45°. It is important to note that once the sampling points and their associated flow vectors are determined, the analysis of this sequence is identical to that of the previous chapter.

Once an estimate for the number of processes and their translational directions are found using the linear-constraint clustering, clustering of bilinear constraints is used to refine estimates for translational direction and rotation for each process. The algorithm is the same as in the previous chapter. Each sampling point provides a single bilinear constraint.

(a)

(b)

Figure 10.2: Horizontal and vertical components of the computed optic flow field are shown in (a) and (b), respectively. Lighter values denote larger components to the right and down. This dense flow field was generated using a parametric-plus-deformation method which uses a planar-surface model as a starting point for flow estimation [8].

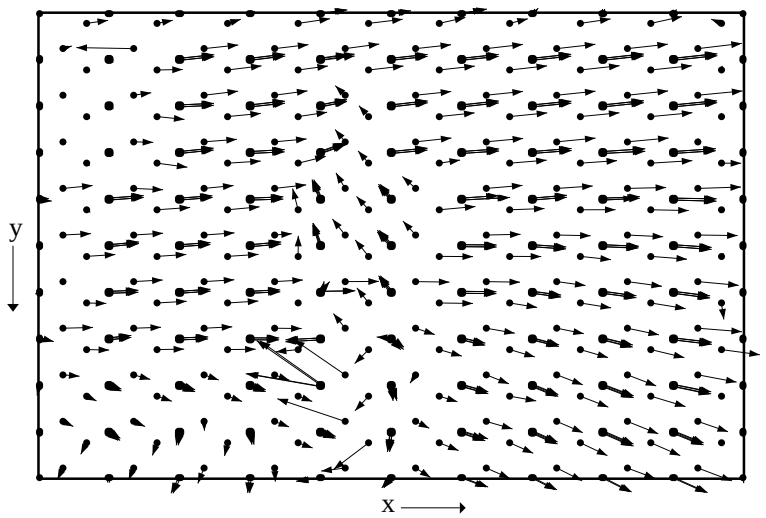


Figure 10.3: Flow samples derived from the dense flow fields depicted in Figures 10.2(a) and (b). The samples represent a 10×10 array of equal-sized patches, each of which contains 6 flow samples.

10.2 Results

Clustering the linear constraints results in two processes being identified. Both motion processes have strong horizontal components to their translational motions, and both are to the left. The estimated translational directions along with their associated FOE are reported in Table 10.1. Figure 10.4 shows the linear constraints prior to segmentation. The linear constraints are predominantly horizontal in a streak that covers the entire image region. Approximately 38% of the linear constraints are marked as outliers, with 56% being owned by the first motion process.

Clustering of the bilinear constraints results in the parameters found in Table 10.2. The two translational directions are both near the x-axis, and to the left of the image's centre. Approximately 3% of the bilinear constraints are marked as outliers, with the majority of the remainder going to the first process. The first process is seen to have a large value for σ. Unfortunately these parameters cannot be trusted. Examining relative inverse-depth values[1] shows that many are negative, and therefore physically

[1]The concept of *relative inverse-depth* will be discussed in detail in Chapter 11. Briefly, it is possible to recover information about depth structure in the image given an estimate for $\vec{T}$ and $\vec{\Omega}$. These estimates are only recovered up to a scale factor, and hence are relative. The method lends itself to recovering the inverse of the depth structure, which is sufficient for the purpose at hand.

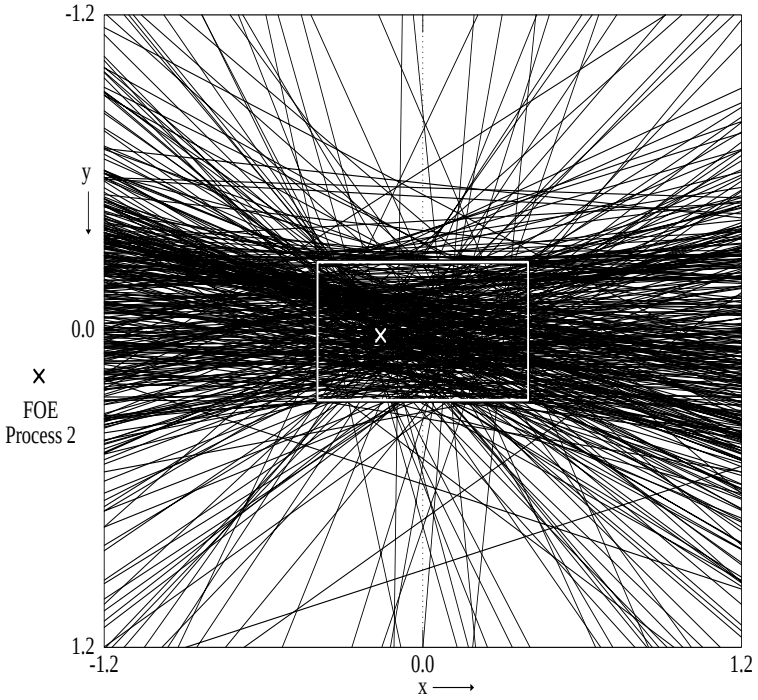


Figure 10.4: The linear constraints are shown. The white × marks the recovered FOE for Process 1 and the black × marks the FOE for Process 2. The inset box shows the extent of the image. The axes are labelled with normalized coordinates (pixels divided by focal length).

unrealistic. What went wrong?

10.3 Problems With the JQ Analysis

In the previous section the clustering of linear and bilinear constraints has led to a solution which yields unrealistic values for relative inverse-depth. There appear to be two reasons for this result. The first reason is that the results of clustering the linear constraints appear to be inaccurate. Since these results form the initial guess for clustering the bilinear constraints, one might expect an adverse effect on subsequent stages of processing. The second reason involves the effect of the value of σ used while clustering bilinear constraints. Both of these problems will now be discussed in further detail.

	Process 0 (outliers)	Process 1	Process 2
Mixtures	0.3817	0.5662	0.0521
$\vec{T}$		$\begin{bmatrix} -0.1571 \\ 0.0167 \\ 0.9874 \end{bmatrix}$	$\begin{bmatrix} 0.8157 \\ 0.1259 \\ -0.5647 \end{bmatrix}$
σ		0.0630	0.0161
FOE		(-69,7)	(-628,-97)

Table 10.1: This table shows the estimated parameters recovered by clustering the linear constraints. The value of σ reported is a measure of the variance of linear constraints from the great circle defined by $\vec{T}$ on the sphere (as defined in Eqn. 5.1). FOE values are rounded to the nearest pixel.

10.3.1 Poor Results From Linear Clustering

The linear constraints fail to provide us with a good estimate of the translational direction. This is likely due to the lack of variation in the flow from the background (see Figures 10.2, 10.3, and 10.8), possibly due to small variation in depth with respect to the camera's motion. Also, a number of flow vectors, primarily along the image boundaries, are obvious outliers and may have caused erroneous results. While the algorithm is designed to remove outliers, structure in the outlier population may be confused with motion processes and incorrectly segmented. Examination of ownership of constraints by the second process shows that they do not provide a clear segmentation of the IMO. It is difficult to know whether the linear constraint clustering picked the correct number of motion processes through chance or whether information from the IMO actually was found.

	Process 0 (outliers)	Process 1	Process 2
Mixtures	0.0315	0.6929	0.2756
$\vec{T}$		$\begin{bmatrix} -0.0600 \\ -0.0354 \\ 0.9976 \end{bmatrix}$	$\begin{bmatrix} -0.4766 \\ 0.0648 \\ 0.8767 \end{bmatrix}$
$\vec{\Omega}$		$\begin{bmatrix} -0.101 \\ 1.491 \\ 0.226 \end{bmatrix}$	$\begin{bmatrix} -0.714 \\ 2.040 \\ -0.591 \end{bmatrix}$
σ		1.1992	0.119613
FOE		(-26, -15)	(-236, 32)

Table 10.2: This table shows the estimated parameters recovered by clustering the bilinear constraints. The value of σ is a measure of the variance of the bilinear constraints with respect to the values of $\vec{T}$ and $\vec{\Omega}$ (as defined in Eqn. 5.4). FOE values are rounded to the nearest pixel.

10.3.2 Effect of σ on Bilinear Clustering

Further examination of the bilinear constraints shows that there are multiple solutions, and that the number and exact parameters for these solutions vary depending on the value of σ used to perform the non-linear optimization. Table 10.3 shows the FOE values recovered for different values of σ. In each case the value of σ was held fixed and the bilinear clustering performed using a variety of starting points (this amounts to a search of the parameter space). In each case a single motion process was assumed. It is apparent that multiple solutions exist for the bilinear constraint clustering. As the value of σ is decreased more local solutions can be expected to appear. This is in fact observed in this sequence. As the value is lowered to 0.5 four solutions become apparent. As the value is lowered to 0.2 six solutions are found. The change in the number of solutions is due to changes in the objective function being minimized. As σ is decreased more local minima appear. It should be noted that the exact locations of these solutions also change with the value of σ. The full complexity of this nonlinear estimation problem is evident.

The problem encountered here does not suggest that clustering bilinear constraints is the wrong thing to do, rather that bilinear constraint clustering may be more complicated in some cases than others. In particular, in the case where the initial guess provided by clustering linear constraints is poor, recovering the correct segmentation from the bilinear constraints may require a more-involved analysis.

10.4 Results From Annealing

This section reports the results obtained by performing the bilinear constraint clustering, but using a different method of controlling the value of σ. Results presented in Section 10.2 involved estimating the value of σ for each motion process at each iteration of the EM algorithm. As discussed in the previous section, clustering the bilinear constraints can result in multiple solutions, the number and exact parameters of which depend on the value of σ used. In this section a different method has been used for controlling σ: instead of estimating σ for each process on each iteration, σ is

σ	# of Solutions	Solutions for FOE (pixels)
1.5	5	(-33.96, -19.79) (-13.29, 49.76) (-49772.82, 335.88) (96.81, -7.06) (-803.08, -66.41)
1.0	2	(94.17, -14.72) (-36.70, -22.63)
0.5	4	(-34.81, -9.08) (-198781.56, 34854.55) (378.78, -105.02) (-408.98, 497.18)
0.2	6	(-132.09, 8.19) (-47.25, -30.38) (24951.19, -2549.07) (3310.07, 3731.67) (392.08, -127.33) (89.10, 112.33)

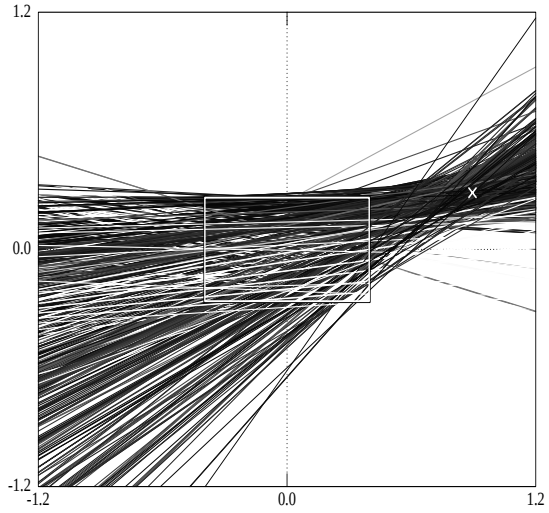
Table 10.3: Different values for σ lead to a different number of solutions for $\vec{T}$ and $\vec{\Omega}$. The FOE solutions for four different values of σ are shown. Each set of solutions was determined by assuming one motion process and using a fixed value for σ. Clustering was performed starting from a number of initial guesses for $\vec{T}$.

assigned an initial value which is then decreased on each iteration towards a pre-set minimum. This is similar to decreasing a temperature parameter in annealing methods. The effect is to allow each process to "see" a large number of constraints initially, and as the EM algorithm proceeds the "field-of-view" available to each process is narrowed. Even though the value of σ is changed using a pre-determined schedule, a new value of σ is calculated after each "M-step" for comparison to the value actually used.

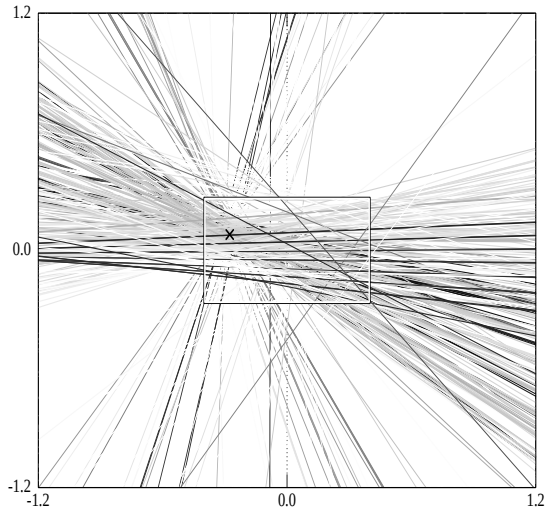
The bilinear constraints are again segmented using the initial guesses generated from the linear constraint clustering. The initial and final values for σ were 1.0 and 0.3 respectively, with sigma being decreased by a factor of 0.95 on each iteration. The final bilinear constraints are shown in Figure 10.5. The FOE for the first motion process is up and to the right, and it lies outside the physical boundaries of the image.

Process 1 has ownership of 73% of the constraints. The constraints associated with Process 1 show a strong clustering to the top-right of the graph, outside the extent of the image. On trials in which the value of σ used for clustering bilinear constraints is allowed to drop below 0.2 the intersection of the constraints is seen to become even more sharply defined. The constraints associated with Process 2 do not have as clear an intersection, nor are they as strongly owned. This is seen by observing that the grey-level in which each constraint is plotted is proportional to its ownership by the process in question.

As has already been discussed in this chapter, the behaviour of σ used in the bilinear clustering stage is of considerable interest for this data. Figure 10.6 shows the relationship between estimated values of σ (one for each motion process) and the controlled value of σ_{actual} used for determining ownership. The dotted line shows the σ estimates for Process 1, which will later be associated with the background constraints. As σ_{actual} is lowered, $\sigma_{estimated}$ experiences sudden drops and then regions of stable behaviour. The stable regions (plateaus) occur at around $\sigma_{estimated} = 0.5$ and 0.2. Although not shown in this figure, for values of $\sigma_{actual} > 1.3$ the value of $\sigma_{estimated}$ is stable around 1.2. This suggests that structure can be seen at several different levels in the flow data. Large σ might correspond to the interpretation that the flow estimates were pure noise. The values of σ below 0.3 (estimated noise



Process 1



Process 2

Figure 10.5: The bilinear constraints are shown after clustering. The top plot, labelled "Process 1" represents constraints owned by the first motion process, and has a FOE up and to the right (outside of the image). Process 2 has a FOE within the image, just left of the image centre. The inset box shows the extent of the image. The axes are labelled with normalized coordinates (pixels divided by focal length). The FOE for each process is marked with an $\times$, and the grey-level of each constraint is proportional to its ownership by the process.

	Process 0 (outliers)	Process 1	Process 2
Mixtures	0.1028	0.7288	0.1684
$\vec{T}$		$\begin{bmatrix} 0.6525 \\ -0.2089 \\ 0.7284 \end{bmatrix}$	$\begin{bmatrix} -0.2658 \\ -0.0769 \\ 0.9610 \end{bmatrix}$
$\vec{\Omega}$		$\begin{bmatrix} 0.550 \\ 5.391 \\ 0.686 \end{bmatrix}$	$\begin{bmatrix} 1.048 \\ 1.497 \\ 4.193 \end{bmatrix}$
FOE		(389.23, -124.64)	(-120.18, -34.76)

Table 10.4: This table shows the estimated parameters recovered by clustering the bilinear constraints. For both processes, the values shown are for $\sigma = 0.3$, the lower limit used in the annealing process.

for the flow in this sequence) likely correspond to the structure in the constraints from the underlying motion processes. The plateaus in Figure 10.6 for Process 1 suggest that different types of structure in the constraints can be seen, each with a different interpretation. The values of $\sigma_{estimated}$ associated with Process 2 do not show a similar plateau structure and are always larger than σ_{actual}. The fact that Process 2 does not have plateaus suggests that it may just pick up constraints that do not match Process 1 but match Process 2 well enough to avoid assignment to the outlier population. It is to be expected that as σ_{actual} is lowered, that $\sigma_{estimated}$ will follow.

The large starting value of $\sigma = 1.0$ used during bilinear clustering makes up somewhat for the poor estimate provided by the linear constraints. Attempts at allowing the algorithm to estimate its own σ resulted in a value that remained large (see Section 10.2). In order to get a reasonable result it was necessary to artificially depress this value. As can be seen from Figure 10.6, once it has been forced below

a value of about 0.7, the estimates for σ will drop to about 0.5 and remain there. It is therefore necessary to get the value of σ low in the first place. It should be noted that the value of $\sigma_{estimated}$ depends not only on σ_{actual} but also on the values of $\vec{T}$ and $\vec{\Omega}$ for each iteration. Experiments performed with large starting values for σ tended to be less successful. Using the initial estimated σ and then forcing it smaller was successful.

For $\sigma = 1.0$ there are two solutions for the FOE: they are (-36.70,-22.63) and (94.17, -14.72). The latter solution appears to grow into the solution of (389.23, -124.64), as reported in Table 10.4, as σ is decreased. This solution would appear to be the "correct" one in that it is the only one which results in the majority of the constraints corresponding to positive depth values (see Figure 10.8). This solution also provides a meaningful segmentation of the constraints into outliers, background motion and independent object motion (see Figure 10.7). Other solutions found had many negative depth values, and did not perform as well on the segmentation task. The first solution can be understood by making an observation about the bilinear constraints. Bilinear constraints all pass through the image location vector $\vec{x}$ to which they belong. This means that if σ is too large the minimization could possibly pick a point near the centre of the image, as it is "near" all the constraints. In the event that the true FOE lies outside the image, this may be an appealing alternative for the algorithm. In this instance the true FOE does indeed appear to lie well outside the image, as seen in Figure 10.5. The first solution bears some resemblance to the first solution found by clustering the linear constraints.

In Figure 10.7 one sees the ownership maps for the motion processes and outlier population. Process 1 has strong support from constraints derived from flow vectors which represent the background motion (egomotion). Ownership of constraints derived from the IMO are shared between the outlier process and Process 2. Flow vectors which could be considered outliers along the right image boundary and near the top-left part of the image are also assigned to these processes. Since there were no estimates of certainty associated with the flow vectors, it is not known if the outliers in the flow can be separated from flow estimates involving the IMO. The segmentation

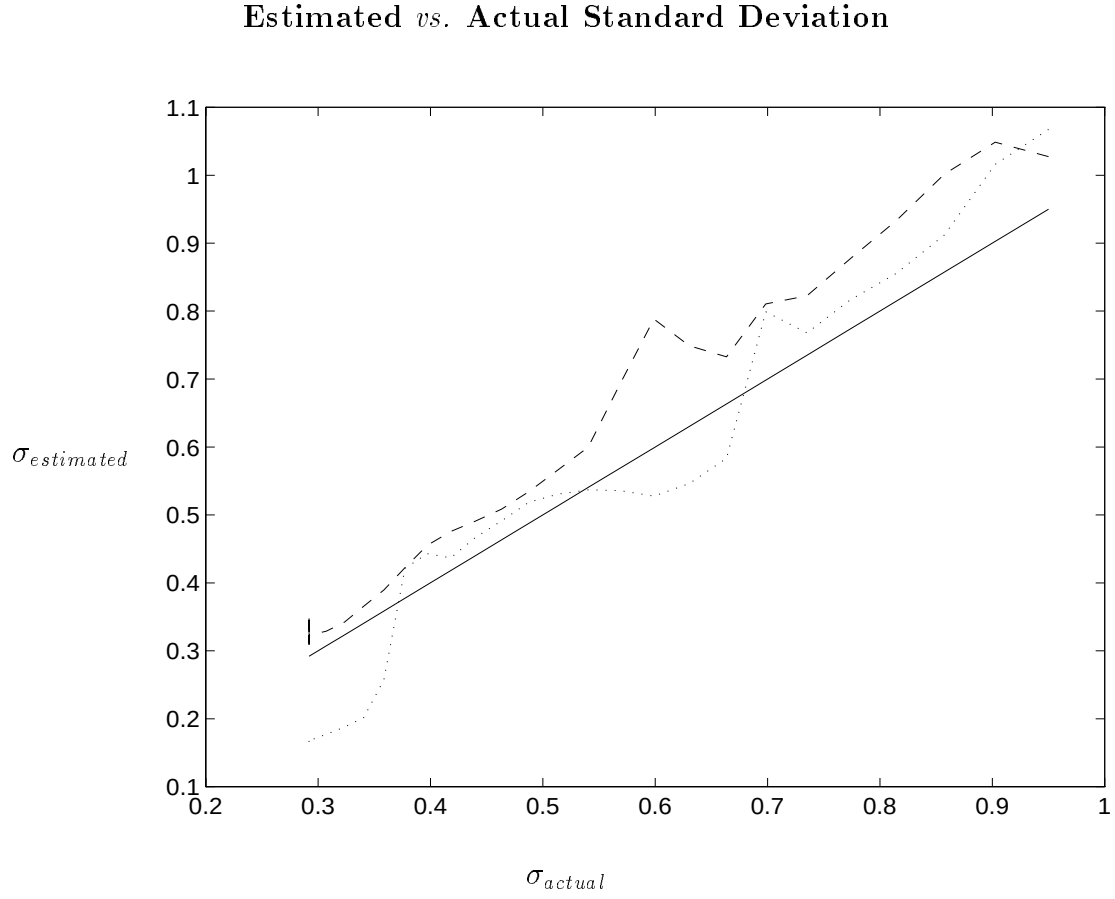


Figure 10.6: The relationship between the value of σ used for clustering and the estimated value for each process is shown. The dotted line is the estimated value of σ for the first motion process, and the dashed line is the estimate for the second motion process. The dotted line has plateaus at around 0.5 and 0.2. The solid line is provided as a reference for the actual σ.

between background and IMO appears to be quite good.

Figure 10.8 shows a histogram of depth values recovered using motion parameters from Process 1. All but a small number (2) of the flow samples give positive values. Since negative depth values indicate an object behind the camera they indicate outliers in the flow vectors or possibly an IMO for which the motion parameters are not valid. The x-axis represents *relative inverse-depth*, a concept which will be discussed in further detail in Chapter 11. The peak near 0 indicates that most of the depth structure in the scene is relatively distant, a fact that supports the observation that the flow field is quite uniform.

The fact that a good solution was obtained from the bilinear clustering using this "annealing" method and the linear solution as an initial estimate is almost certainly a matter of luck. On trials which used an initial value of $\sigma > 1.3$ the clustering converges to the wrong solution. However, solutions obtained by starting with the solutions for a fixed $\sigma = 1.5$, and then decreasing σ result in the two solutions discussed above. This again underscores the problem of a good initial guess for the bilinear clustering.

10.5 Summary of JQ Sequence Results

It was shown in the last chapter that linear and bilinear subspace constraints can be used to effectively segment IMOs in image sequences. In that image sequence there was rich depth structure and accurate flow was available. In this chapter it was shown that although the constraints are still useful in performing the desired segmentation, the method is by no means fool-proof. The lack of variation in the flow due to limited depth structure, together with less-accurate flow estimates containing outliers (both in terms of an IMO and bad flow estimates at the image boundaries) caused the linear clustering to be inconclusive with regards to an initial estimate for translational direction or the number of motion processes. The clustering of the bilinear constraints, while it yielded the results one would hope for, only did so after careful consideration surrounding the choice of σ, and careful discarding of "erroneous" solutions. Verifying the correct solution involves the search of a solution

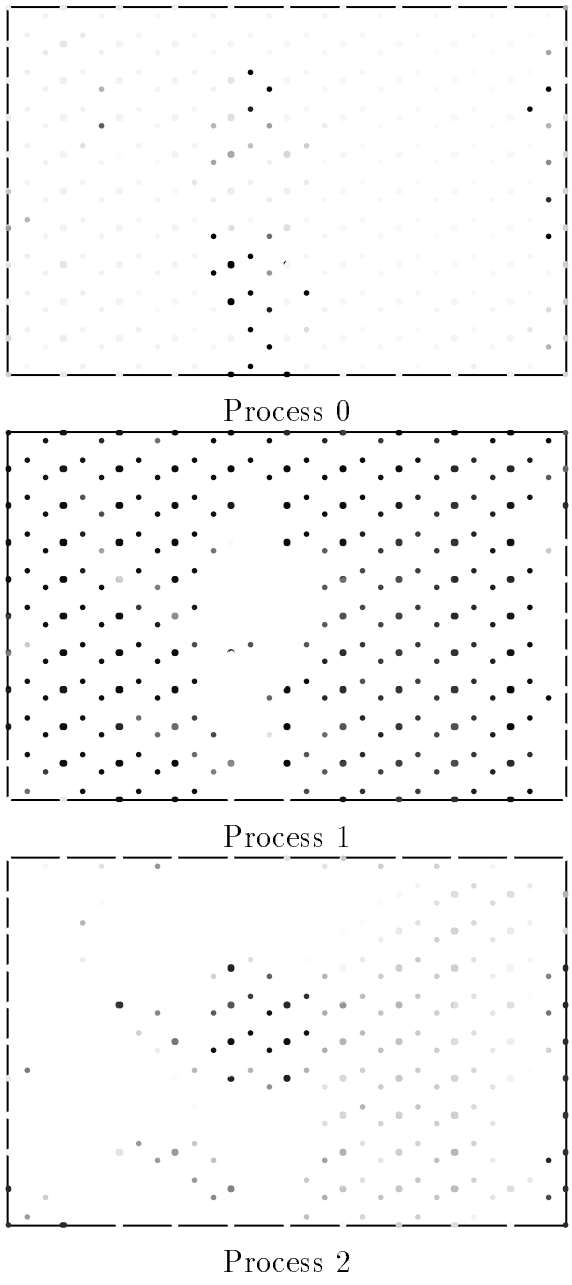


Figure 10.7: The support for the outlier population plus the two motion processes are shown. The grey-level of each point indicates its support for the process, with black representing strong support. Process 0 draws some support from the IMO as well as flow samples near the right image boundary. Process 1 draws support from the background, and Process 2 from the IMO and some points on the right image boundary.

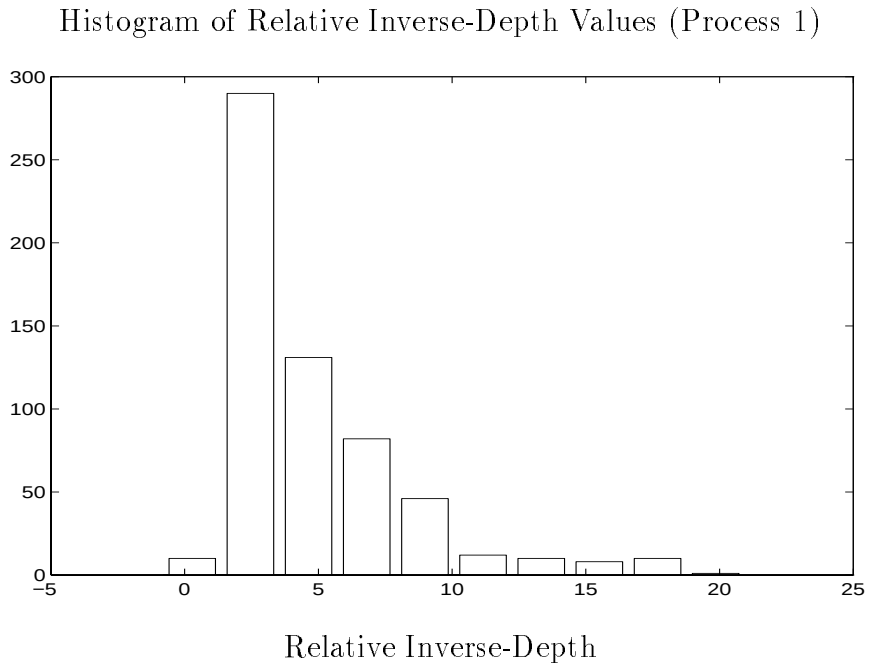


Figure 10.8: The depth values as computed using parameters from Process 1. The values are shown as relative inverse-depth, therefore values near zero indicate relatively distant objects in the scene. There were only two negative values, possibly resulting from outliers.

space using a nonlinear optimization process.

While the subspace methods appear promising (both in theory and practice), more work is necessary. The problems encountered here serve to emphasize the importance of a good initial guess for the motion processes. A temporal integration framework where estimates from a previous frame are carried forward as initial estimates for the current frame is one possible solution.

Chapter 11

Results from Van Sequence

This chapter presents the results from another sequence, captured using a hand-held video camera. The methods for recovering flow and segmenting constraints are identical to those in Chapter 9, but the results have some different features. The image sequence, which is 20 frames long, was taken from an automobile coming to a stop at an intersection. The car on the left side of the image, shown in Figure 11.1, and the pedestrian are IMOs.

Seven patches are used to model the flow for objects in the scene: street-light (Light), pedestrian (Person), car (Car), van (Van), road (Road), CN Tower[1] and adjacent buildings (CNT), and the tree in the upper-right corner of the image (Tree). As with the floor patch in the forklift sequence, the flow for the road is modelled with a rational flow model. The patches for the van, the CN Tower, the tree and the car are all affine models. The pedestrian and the street-light are each modelled using the constant-flow model. (Chapter 9 contains a description of the constant, affine, and rational flow models.) Recovered flow samples for the frame shown in Figure 11.1 are plotted in Figure 11.2.

The estimated observer translation for this frame is $\hat{T} = [0.0390\ 0.0114\ 0.9992]^T$. This gives an FOE of [25.0 7.3]. Only one motion process is recovered for this sequence, even though both the car and pedestrian are moving independently of the

[1] A communications tower nearly 2000 feet tall, situated in downtown Toronto.

Figure 11.1: A frame from the van sequence. There are seven patches: road, tree, CN Tower, van, street-light, pedestrian, and car. The last two are independently moving objects. Each patch is marked, along with the FOE.

observer. This can be explained. The flow for the pedestrian can be projected (nearly) through the FOE. The fact that this flow goes towards the FOE instead of away from it will not affect the final result, which is that constraints generated by combining the pedestrian patch with the road patch will be consistent with the translation associated with the observer motion. The car in the initial frames also provides flow that projects close enough to the FOE to suffer the same fate. Recall that subspace methods fail when the translational component of the flow is either zero or directed along a line passing through the FOE. Conditions under which the subspace constraints will fail were discussed in Section 4.5. A specific case occurs when the image of an IMO is concentrated around a plane defined by the relative translations of the observer with respect to the static environment and with respect to the IMO. Assuming that this plane also contains the nodal point of the camera, and that any rotational motion of the IMO is orthogonal to the plane, then subspace methods will fail to discriminate the IMO. This case can arise easily for an observer and IMO's moving on a flat surface, such as a road.

Despite the failure of segmentation based on the subspace constraints, it is still possible to identify objects that are moving independently in this sequence. Once estimates for $\vec{T}$ and $\vec{\Omega}$ (egomotion) have been recovered, it is possible to recover estimates of depth structure in the image. It is possible to predict how this depth structure will evolve for a static scene, and, by comparing this to the recovered structure, identify IMOs.

11.1 Depth estimates

The motion field equation (Eqn. 2.2) can be rewritten as

$$\vec{u}(\vec{x}) = R(\vec{x}) \left(\frac{f||\vec{T}||}{X_3(\vec{x})} \hat{T} + \vec{\Omega} \times \vec{x} \right) \quad . \tag{11.1}$$

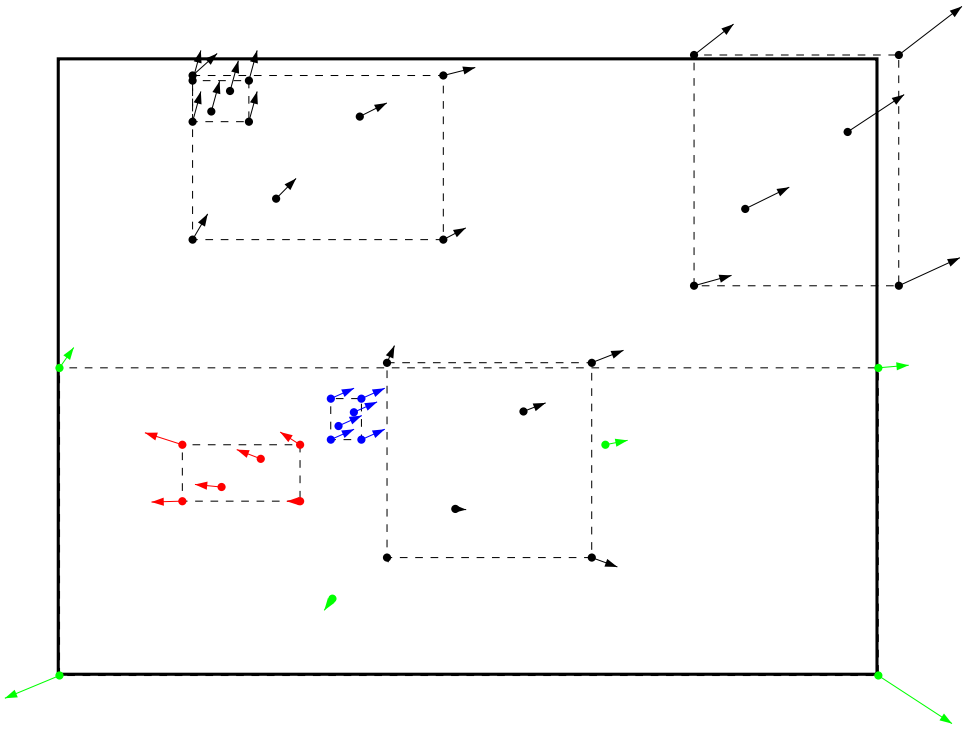


Figure 11.2: This figure shows the optic flow recovered from the frame shown in Figure 11.1. Each dashed box indicates a patch described by an affine, rational or constant flow model. For each box, six samples of the flow have been plotted, according to the model used to that patch. The green flow vectors are from the road patch. The red flow vectors are from the moving car, and the blue flow vectors are from the pedestrian.

In this equation $\hat{T} = \vec{T}/||\vec{T}||$ is translational direction, and $||\vec{T}|| \neq 1$ (in general). Refer to the quantity

$$\frac{1}{D} = \frac{f||\vec{T}||}{X_3}$$

as *relative inverse-depth* (RID). If the translational direction and the rotation are known, then RID can be recovered from optic flow as discussed in Section 6.5. If one also knew $||\vec{T}||$ and f, then it would be possible to recover absolute depth. A method will be developed in which, given RID estimates, it is possible to estimate focal length and identify objects which may be moving independently. To do this a model will be developed for the evolution of RID assuming $\vec{\Omega} = 0$ and that $\hat{T}$ is known for each frame.

11.2 Evolution of Relative Inverse-Depth in Time

Assume an image sequence with n frames in which a number of objects are identified. Let d_{ij} be the (absolute) distance to the jth object in the ith frame. The translational motion for this frame is $\vec{T}_i$, and it is assumed that $\vec{\Omega}_i = 0$ for all frames in the sequence. Knowing an object's depth[2] for a given frame, one can derive the corresponding depth value for the next frame: $d_{(i+1)j} = d_{ij} - ||\vec{T}_i||\hat{T}_{i_z}$. Assume that $\hat{T}_z > 0$ for an observer moving forwards, and that $d_{ij} > 0$ means that the object is in front of the observer. This is rewritten as

$$\frac{d_{(i+1)j}}{||\vec{T}_{i+1}||} = \frac{||\vec{T}_i||}{||\vec{T}_{i+1}||}\left(\frac{d_{ij}}{||\vec{T}_i||} - \hat{T}_{i_z}\right) . \tag{11.2}$$

Define

$$D_{ij} = \frac{d_{ij}}{f||\vec{T}_i||}$$

so that $1/D_{ij}$ is the measured RID. By recursive solution of Eqn. 11.2 one arrives at the relation

$$D_{ij} = v_i\left(D_j - \frac{1}{f}\sum_{k=1}^{i-1}\frac{\hat{T}_{k_z}}{v_k}\right), \qquad i > 1 , \tag{11.3}$$

[2]By "depth" I refer to the X_3 component of the point $\vec{X}$ in the scene.

where $D_j = D_{1j}$. The quantity v_i is called *relative velocity* and defined as

$$v_i = \frac{||\vec{T}_1||}{||\vec{T}_i||}.$$

It relates the magnitude of the velocity of the current frame to that of frame 1 (this is as close as one will get to recovering actual velocity). Eqn. 11.3 allows for the recovery of the RID curve for an object knowing its initial RID, focal-length, translational direction, and relative velocity between frames.

In the event that the translational motion is undergoing constant change in speed,[3] then the following relation holds:

$$v_i = 1 + \beta(i-1).$$

The constant β determines this change in speed. Constant speed has $\beta = 0$, increasing speed requires $\beta < 0$ and decreasing speed $\beta > 0$.

11.3 Time-to-Adjacency

It is possible to define *time-to-adjacency* (t_{adj}) as that time when $d_{ij} = 0$. The RID will be infinite at this point. An instantaneous estimate for t_{adj} can be computed as $t_{adj_{ij}} = d_{ij} / \left(||\vec{T}_i|| \hat{T}_i \right)$. This assumes that $||\vec{T}||$ and d are measured in compatible units, such as pixels/frame and pixels. Given an estimate for RID and f, t_{adj} can be recovered in units of frames^{-1}. Figure 11.3 shows simulated RID values for an object. The initial values show the object to be in front of the observer, which is moving forwards with constant translation. The observer is adjacent to the object at $t = 5$ seconds. Notice that the RID goes to $+\infty$ prior to this time, and returns from $-\infty$ afterwards. For $t > 5$ the RID values are negative, indicating that the object is now behind the observer. Burlina *et. al.* [10] use temporal parameters to estimate *time-to-collision* assuming zero (or known) rotation and arbitrarily smooth

[3]Note that this does not imply constant acceleration. Since it is assumed that the translational direction is known, only the magnitude of the translation, *i.e.* speed, is of interest.

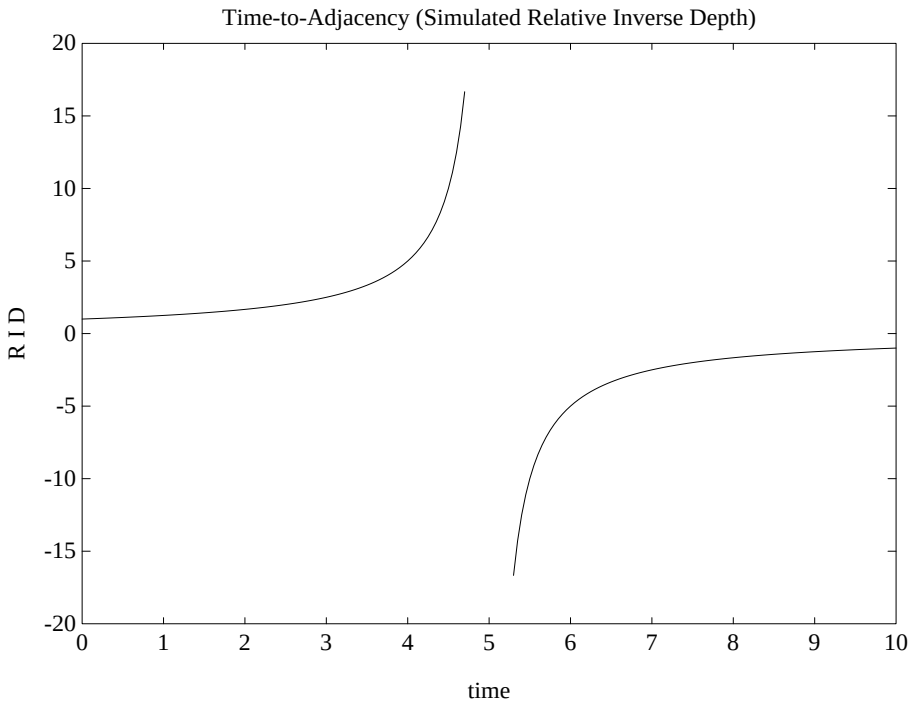


Figure 11.3: Relative inverse-depth has been simulated for an observer moving towards an object with constant translational velocity. The observer is adjacent to the object at $t = 5$ seconds.

translational motion.

11.4 Mixture Model for Relative Inverse-Depth

From Eqn. 11.3 it is observed that once f, $\{v_i\}_{i=1}^n$, $\left\{\hat{T}_{i_z}\right\}_{i=1}^n$ and D_{1j} are set, the RID curve for the jth object is completely determined. It is proposed that, using a mixture model to achieve robustness, it should be possible to estimate the parameters f, $\{v_i\}_{i=1}^n$, and $\{d_{1j}\}_{j=1}^m$ for an image sequence. The model will have two processes, one to model RID curves and the other to account for outliers. Each measurement $1/D_{ij}$ will have a probability of w_{ij} that it belongs to the RID model, and $1-w_{ij}$ that it is an outlier. Application of the EM algorithm allows iterative refinement of the model parameters and the ownership probabilities. IMOs will not evolve according to the RID model as they will have a different value for $\vec{T}$ from that determined for the egomotion. Therefore, it is expected that IMOs will have a higher outlier probability

than objects that are static in the scene.

The objective function minimized in the 'M'-step of the EM algorithm is

$$f_{obj} = \sum_{j=1}^{m} \left\{ w_{1j} \left(\frac{1}{D_{1j}} - \frac{1}{D_j} \right)^2 + \sum_{i=2}^{n} w_{ij} \left(\frac{1}{D_{ij}} - \left[v_i \left(D_j - \frac{1}{f} \sum_{k=1}^{i-1} \frac{\hat{T}_{k_z}}{v_k} \right) \right]^{-1} \right)^2 \right\} ,$$

$$v_i = 1 + \beta(i-1) .$$

This function is nonlinear in the parameters f, β and $\{D_j\}_{j=1}^{m}$. A Polak-Ribiere conjugate-gradient method [72] is used to perform the minimization, with the w_{ij}'s held fixed.

During the 'E'-step, an ownership (probability) is calculated for each data point indicating how well it fits the model. Let D_{ij} be a (measured) data point, and $\hat{D}_{ij}$ the model's predicted value. Define $\Delta D_{ij} = D_{ij} - \hat{D}_{ij}$. Then the probability that the data point belongs to the model is given by

$$p_{model}(\Delta D_{ij}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ \frac{\Delta D_{ij}^2}{2\sigma^2} \right\} ,$$

where σ is a parameter chosen to model the variability in the measurements. Outliers are modelled by a uniform probability distribution

$$p_{outlier}(\Delta D_{ij}) = p_0 .$$

As in Chapter 6, the value of p_0 is chosen so that a data-point has equal probability of belonging to the model and outlier populations when it is a distance of $\rho\sigma$ from the model's prediction, *i.e.* $p_0 = p_{model}(\rho\sigma)$. The parameter ρ is held fixed. The ownership value for a data point is then given by

$$w_{ij} = \frac{m_{model} p_{model}(\Delta D_{ij})}{m_{outlier} p_0 + m_{model} p_{model}(\Delta D_{ij})} .$$

The quantities m_{model} and $m_{outlier}$ are the mixture proportions for the model and outlier processes, respectively.

Each RID curve can be assigned a likelihood value based on the ownership values for points belonging to that curve. Specifically, likelihood is given by $l_j = \prod_{i=1}^{n} w_{ij}$ and log-likelihood by $\log l_j$. Likelihood values can be used as a figure of merit when attempting to determine how well a particular curve fits the overall model. Note that, since $0 \leq w_{ij} \leq 1$, the magnitude of l_j will be affected by the number of frames in the sequence, n.

11.4.1 Determining Relative Inverse-Depth Estimates

RID estimates are determined for each patch in each frame in the van sequence. For each patch, 6 points are chosen in the first frame and tracked throughout the sequence. RID values are recovered for these points in each frame, and fit to a planar model,

$$\frac{f||\vec{T}||}{X_{3_l}} = \hat{n}^T \vec{x}_l \ .$$

One can solve for $\hat{n}$ by solving 6 equations in 3 unknowns:

$$\begin{bmatrix} x_{11} & x_{12} & 1 \\ & \vdots & \\ x_{61} & x_{62} & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{X_{3_1}} \\ \vdots \frac{1}{X_{3_6}} \end{bmatrix} \frac{f}{||\vec{T}||} \ .$$

This can be solved via a linear least-squares method. As discussed in Section 9.2.1, the planar model can be expected to provide a better fit for the rational patches, but may not be as good for the affine and constant velocity patches. Finally, a single depth estimate is generated for each patch for each frame. This is done by evaluating $\hat{n}^T \vec{x}^c$ where $\vec{x}^c$ is the average position of the 6 points. The averaging inherent in this model is important to provide noise immunity, especially as several of the patches contain the FOE, around which RID measurements can be expected to be quite noisy. Figure 11.4 shows the recovered RID values for the van sequence.

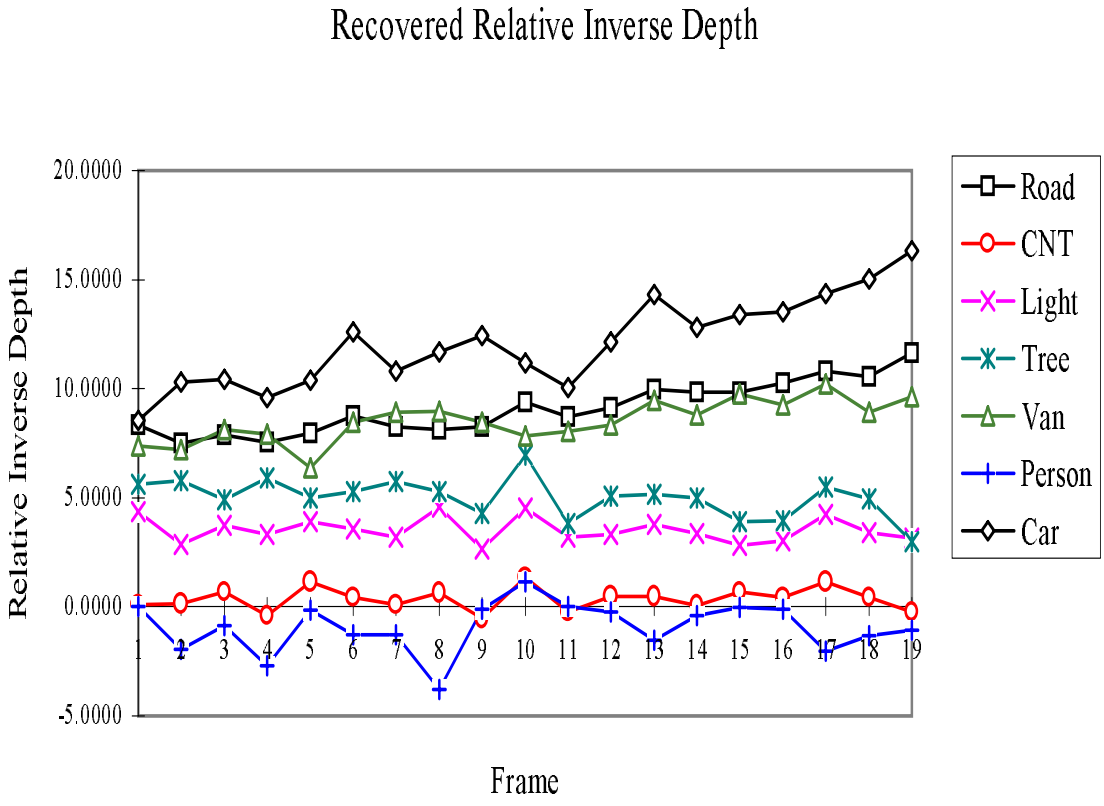


Figure 11.4: Relative inverse depth has been recovered independently for each frame of the sampled flow shown in Figure 11.2. Six points were tracked in each patch, and the recovered depth estimates fitted to a planar model.

11.4.2 Initial Guesses for Parameters

In order to reduce the number of parameters in our model, the constant acceleration model discussed earlier is adopted. This allows replacing $\{v_i\}_{i=1}^n$ with a single parameter, β. This reduction is helpful, as our minimization procedure during the 'M'-step of the algorithm is nonlinear. Our initial choice for this parameter is $\beta = 0$, indicating constant speed.

The initial guess for focal-length, f, is chosen to be large ($f = 1000$ pixels was the chosen value). Experimentation with the algorithm showed that convergence was assisted by choosing f "too large" as opposed to "too small". Choosing f "too large" has the effect of flattening the curves produced by the model. No empirical value for the focal-length of the camera was available.[4] In many cases knowledge of the camera optics should allow a better initial guess for f to be made.

A simple choice for the values of D_j would be to use the measured value, D_{1j}. However, given that this value will determine the shape of an individual RID curve, a different value was used. Each measured RID curve was individually fit to a curve of the form

$$D_i = \frac{a}{b - i} \, .$$

This allows using $a_j/(b_j - 1)$ as the initial value for D_j instead of using D_{1j}. The initial estimates thus chosen allowed for faster convergence of the optimization procedure.

Finally, the values of $\left\{\hat{T}_{i_z}\right\}_{i=1}^n$ are already known from our determination of ego-motion from the sequence.

11.5 Results: RID Mixture Model

Figure 11.5 shows the fitted RID curves superimposed on the measured ones. The fit can be seen to be quite good. Table 11.1 shows that a large improvement is made in the objective function. The pedestrian is shown to have a negative depth value,

[4]The camera was fitted with a zoom lens, and the exact focal length was not recorded at the time the image sequence was captured.

Parameter		Initial	Final
D_0	Road	7.472	7.998
	CNT	3.149	3.293
	Light	3.717	3.646
	Tree	7.462	7.651
	Van	5.780	5.246
	Person	-0.931	-0.581
	Car	9.555	9.459
f (pixels)		1000	239.2
β		0.0	0.0252
f_{obj}		221.1	16.90

Table 11.1: This table reports the results for the RID mixture model when applied to the van sequence. The results quoted here are for $\sigma = 0.4$ and $\rho = 2.0$.

$D_0 < 0$ and the RID curve labelled "Person" stays negative over all frames. This fact alone indicates that the pedestrian is moving independently, since it is impossible for an object to have a negative depth *and* be visible in the image.

Table 11.2 shows the likelihood values for each patch with respect to the model. The patches corresponding to the pedestrian and car are seen to have very small likelihoods, indicating that they are probably moving independently.

The tree also scores poorly in terms of the likelihood function. This can be explained in terms of the optic flow recovered for the tree. The affine model is used for this flow, but since the tree is significantly non-planar, the resulting flow has a higher degree of error. This can be observed by warping the image with the inverse of the affine transform and viewing the resulting sequence—the tree is roughly stabilized but still exhibits motion. By contrast, performing the same operation for any other patch results in the object of interest being quite well stabilized. One observes, therefore, that errors in recovery of optic flow will have a strong impact on this method. It should also be noted that attempting to fit depth values to a planar surface, as described above, will lead to poor results for the tree patch.

Another source of error occurs when the points tracked for each patch move outside the region. This happens, for example, with the road patch. Referring to Figure 11.2, 3 of the corner points will move outside the image in the next frame. This means that

j	l_j	$\log l_j$
Road	1.6e-02	-1.79
CNT	6.3e-04	-3.20
Light	1.1e-04	-3.95
Tree	1.0e-17	-17.00
Van	1.0e-05	-4.98
Person	4.3e-30	-29.37
Car	1.1e-16	-15.95

Table 11.2: This table shows the likelihood values for each patch tracked in the image. The pedestrian and car have low likelihoods, suggesting that they are moving independently. The tree also has a low value, probably due to the poor fit of an affine flow model to this type of surface. The results quoted here are for $\sigma = 0.4$ and $\rho = 2.0$.

the flow transformation will be extrapolated for these points. For the case of affine or rational flow, this extrapolation may lead to errors in the recovered RID values. While the likelihood for the road is very good, it is not perfect.

The value of β listed in Table 11.1 is approximately 0.025. This indicates that the observer velocity is decreasing, which is consistent with the conditions under which the image sequence is captured. Figure 11.6 shows the recovered values for the v_i parameters, corresponding to the curves shown in Figure 11.5.

The values of the ownership weights for the individual data points are shown in Figure 11.7. They are plotted on a logarithmic scale, *i.e.* $\log w_{ij}$. For the road, all the data points are in good agreement with the model. On the other hand, while some data points for the car are also in good agreement, others fit the model very poorly giving the car a poor overall likelihood. As noted above, one should expect poorer results for the tree patch since it is not well-modelled by a planar surface. In each frame it is possible to find points within the tree patch that lie on a planar surface, but there is no guarantee that the same plane will be recovered for this patch in each frame.

Finally, it is worthwhile considering the effect of σ on fitting the model. This value will depend on the expected error in the flow estimates. Figure 11.8 shows the resulting RID curves fitted to the van sequence with $\sigma = 0.3$. The curves do

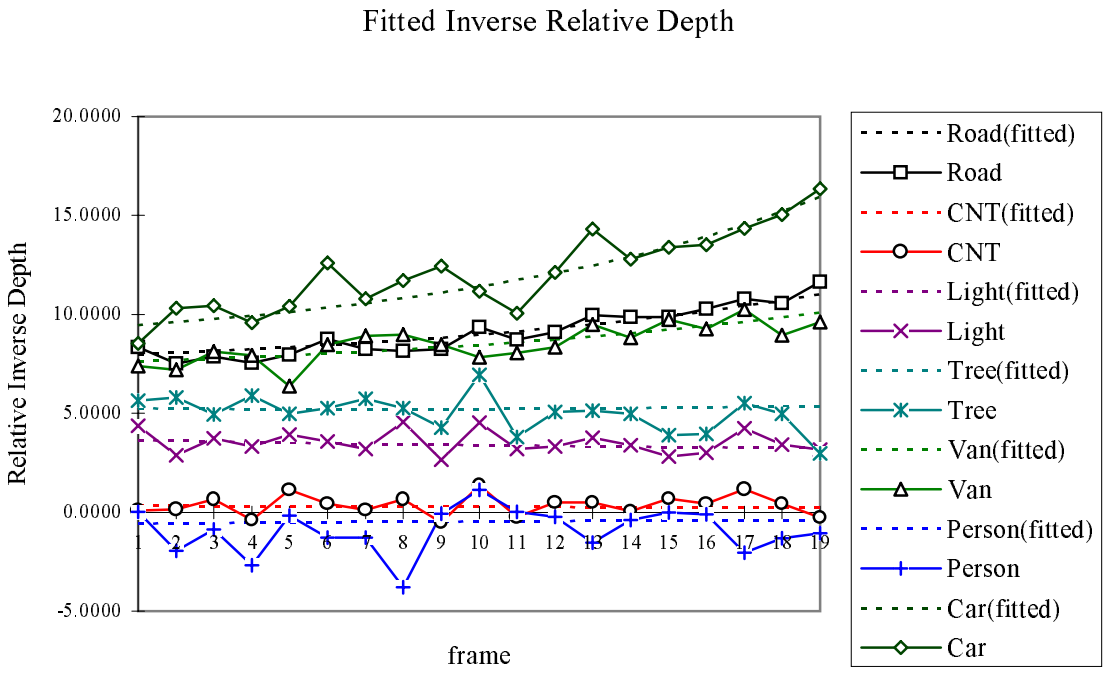


Figure 11.5: Smooth curves were fitted to each curve in Figure 11.4. In this plot $\sigma = 0.4$.

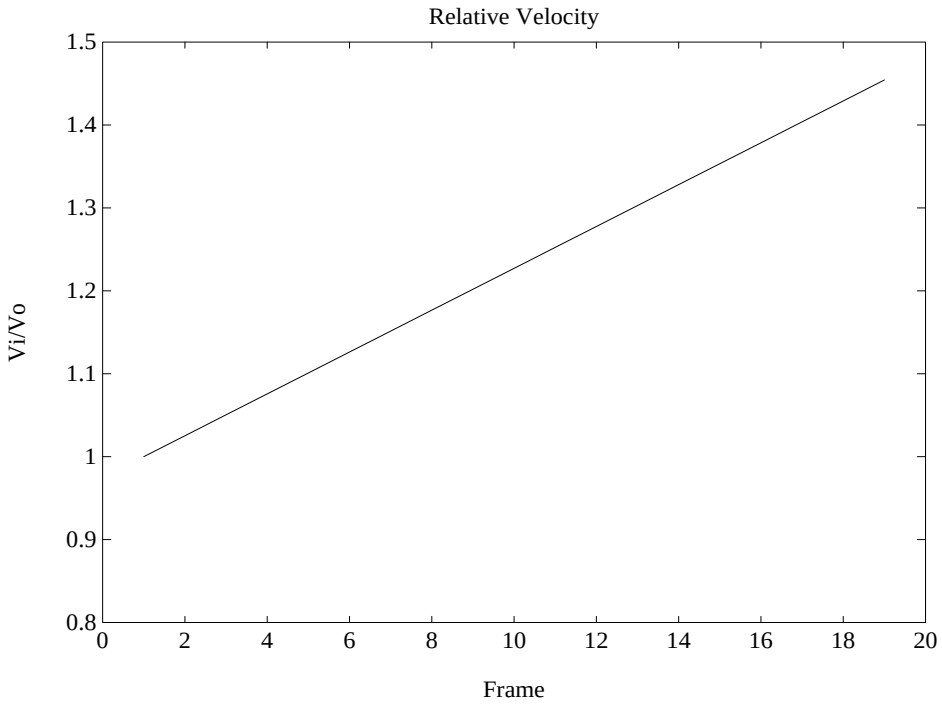


Figure 11.6: This figure shows the recovered v_i parameters. The slope of the graph is β, and in this case indicates a constant deceleration.

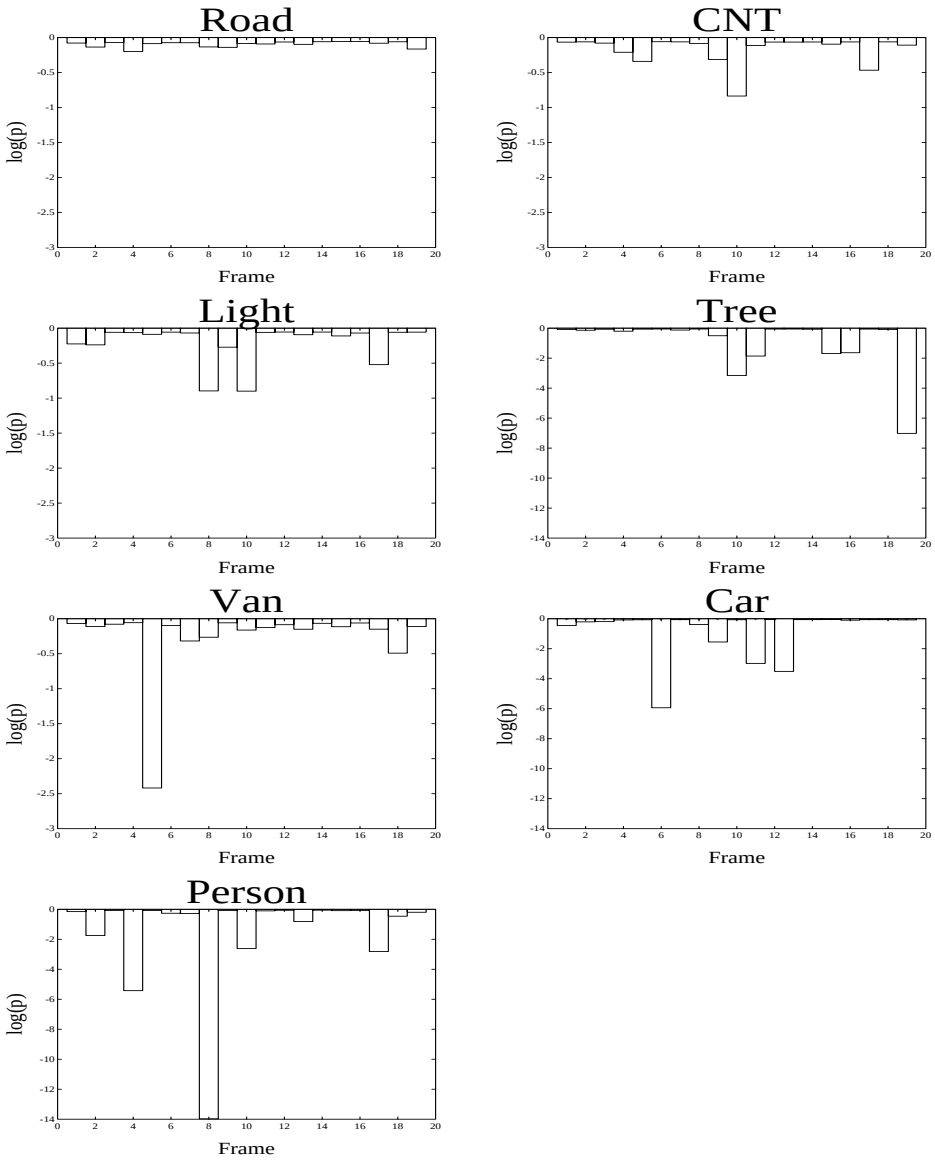


Figure 11.7: This set of figures shows the ownership of data points for each patch over the entire sequence. The ownership probabilities are plotted as $\log p$ and therefore range from 0 for $p = 1$ to $-\infty$ for $p = 0$. The patches corresponding to the road are the best, followed closely by those for the CN Tower, the light, and the van. The tree patch doesn't fare as well, but this is expected since the tree is close and poorly approximated by a planar surface. The patch for the car also shows low ownership probabilities, as does that for the person. (Note the different scale used for the Car, Tree, and Person ownership plots.)

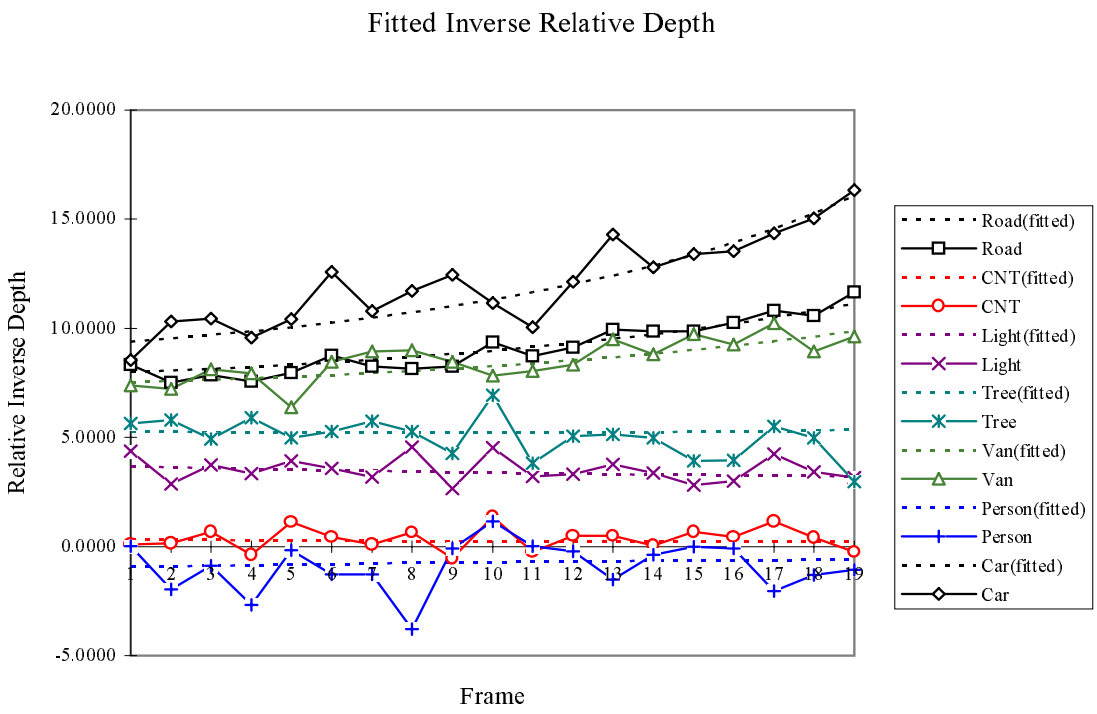


Figure 11.8: This is the same as Figure 11.5, except $\sigma = 0.3$.

not appear to change substantially. The new values for f and β are, respectively, 226.9 pixels and 0.0278. In Table 11.3 observe that the ordering of the likelihood values remains the same. As σ decreases one expects to see an effect on the curves produced since more points will be labelled as outliers and thus will have less input into determining curve parameters.

The data presented here do not satisfy the condition $\vec{\Omega} = 0$. In fact, for the frame shown in Figure 11.1 the rotation is $[0.40\ 0.32\ 1.41]^T$ (this has been scaled by f to give a sense of the rotational magnitude in pixels). The measured rotation is $||f\vec{\Omega}|| = 1.469$ pixels/frame, which is quite significant by comparison to the optic flow magnitude in many of the patches. It has still been possible to recover useful information about the image sequence. It should be possible to modify Eqn. 11.3 to account for the effect of rotation and include the effect of $\vec{\Omega}$ in the model.

j	l_j	$\log l_j$
Road	2.0e-03	-2.70
CNT	1.4e-07	-6.85
Light	2.3e-09	-8.64
Tree	2.8e-34	-33.55
Van	3.5e-11	-10.46
Person	0.0e-00	$-\infty$
Car	1.4e-33	-32.85

Table 11.3: This table shows the likelihood values for each patch as in Table 11.2, except that a smaller value for σ has been used. The value shown for the car is due to numerical underflow during computation. The results shown here are for $\sigma = 0.3$ and $\rho = 2.0$.

11.6 Summary

In this chapter a new method for identifying IMOs based on recovered relative inverse-depth is described. Segmentation of linear and bilinear subspace constraints is not always sufficient to identify IMOs, but in this case it is possible to use the expected evolution of RID curves to determine which objects are part of the stationary environment. The concept of mixture models was used to perform a robust estimation of model parameters and identify outliers in the data. These outliers may indicate the presence of IMOs whose RID curves evolve differently from those belonging to objects in the stationary environment. This is due to the different relative velocity between the observer and environment, as opposed to that between the observer and the IMOs.

This work could be reformulated in terms of the affine/projective structure from motion work [51, 22] cited in Chapter 3. This work assumed a rigid world with a moving observer, much the same as the model developed for the evolution of RID in time. By applying the concept of mixture models to matched image features (in this case, image locations for which optic flow has been recovered) it should be possible to detect those points which violate the rigidity assumption. An advantage of using affine or projective structure is that it becomes unnecessary to estimate focal length in the model. A version of this using only orthographic projection and affine structure, and

not based on mixture models, has been proposed by Lawn & Cipolla [54]. Violation of rigidity has been suggested and used by others [81, 85].

Chapter 12

Contributions & Future Directions

The preceding chapters have discussed issues relating to the detection of independent object motion in image sequences which contain image motion due to egomotion. The approach combines the use of robust statistical methods with constraints on the underlying 3-D motions. Only one previous work [83] combines these two characteristics, but the theory and implementation is completely different. This thesis makes a number of novel contributions to this area of vision research. The present chapter will summarize these contributions and will also suggest avenues for new work related to this problem.

12.1 Contributions

12.1.1 Application of Mixture Models to 3-D Motion Segmentation

There have been few previous approaches that involve 3-D motion directly, and none of these involve the concept of mixture models. It is proposed that mixture models are a natural and powerful method of representing derived constraints on 3-D motion. The availability of the EM algorithm to perform simultaneous segmentation and parameter estimation is a decided advantage to this particular representation. The EM algorithm, although nonlinear, is simple to implement and guarantees an

improvement in the likelihood function on each iteration. Its nonlinear nature requires that initial guesses be reasonably close, but it is shown that there are methods for generating these guesses.

Mixture models assume that the number of underlying processes is known in advance, since the problem of testing for the number of processes is generally unsolved. Since the number of motion process will not usually be known in advance[1] I have proposed a new method of examining structure inherent in the combined constraints to spawn new motion processes when necessary. This method for determining the number of processes is domain specific, but the concept should be generalizable to other problems where underlying structure is known, or assumed to be known.

The approach to identification of IMOs is two-fold:

- segmentation by clustering linear & bilinear constraints on 3-D motion, and
- segmentation by studying evolution of depth structure and determining outliers which may be the result of IMOs.

The latter approach is necessary to handle situations in which the subspace constraints do not provide enough information to perform the identification of independent motion. In summary, this thesis presents a new method for motion recovery based on robust statistical clustering of constraints on 3-D motion, as well as robust estimation of the evolution of depth structure with respect to egomotion.

12.1.2 Development of Effect of IMO Boundaries on Subspace Constraints

The subspace methods used to derive the constraints on 3-D translation and rotation assume rigidity of the environment. Independent motion violates this assumption, making it necessary to understand the effect of generating constraints across IMO

[1] In examining a sequences of images one can assume that the number of motion processes in any particular frame will be the same as in the previous one. This assumption will fail in the event that an object moves in or out of the field of view. The latter case should be manageable through tracking of objects. The problem of starting the system from scratch can be considered similar to the case where a new object enters the field of view.

boundaries. It is not sufficient to use discontinuities in the underlying flow as a guide, since not all such discontinuities represent IMO boundaries. Constraints generated across IMO boundaries can be expected to be erroneous. In fact, Chapters 8, 9 and 11 all present data in which this has happened. In many cases it may be possible to proceed without special consideration, but it is best to understand the underlying theory that governs this case.

This thesis presents an analysis of what happens to a linear constraint generated across IMO boundaries. This analysis shows that the resulting constraint is the average of the two constraints that would be expected if only the motion due to egomotion or the IMO were present, plus a term which depends on the exact geometry of the flow points used to calculate the constraint. Knowledge of the expected form of outlier constraints is useful for later stages of analysis.

12.1.3 Dealing with Anisotropic Nature of Constraint Noise

The subspace methods are exact in the absence of noise. It has been shown that the addition of isotropic noise into the optic flow leads to a highly non-isotropic noise distribution in the resulting constraints. As a result, linear methods used to recover translational direction suffer from an inherent bias which is accentuated by small fields of view. This thesis presents an analysis that shows that this bias can be removed if the noise covariance matrix is known up to a scale factor (or can be estimated). The constraint vectors can be rescaled prior to estimating the translational direction. Results are presented in which an estimate of noise covariance is used to reduce considerably the bias in the estimates.

The presence of this bias is not unique to the subspace methods. Kanatani [48] suggests an iterative method in which the bias is removed by successive subtraction of an estimated covariance matrix. It is proposed that the method presented herein is an improvement as it is not iterative, it can be proven exact in a close-form analysis and it requires only knowledge of the covariance matrix up to a scale factor. This result is of importance to motion estimation problems in general.

12.1.4 Observations on the Effect of Fixation on Motion Segmentation

It is observed that fixating a background point improves the SNR of subspace constraints when a relative-error model is assumed for the optic flow (see Section 8.3). This, in turn, improves ability to perform segmentation of IMOs. It was noted that the subspace methods can be related to the generation of constraints by other methods. This suggests that fixation may have a more general purpose with respect to vision. Fixation is generally assumed to be necessary for stabilization of images on the retina in order to prevent blur. Specifically, the opto-kinetic and vestibulo-ocular reflex are assumed to serve this purpose [11]. It may be that fixation serves to enhance detection of IMOs by reducing the dynamic range of visual motion, in order to better enhance flow variations due to depth that are necessary for recovery of motion parameters. Any mechanism which enhances detection of independent motion (which may be prey or predator in some cases) would be expected to have value in an evolutionary sense. While this is only speculation, the observation may prove important in understanding how the brain processes visual information. Thomas *et. al.* [80] have suggested that fixation by a moving observer assists in avoiding collisions with static objects. The analysis of fixation presented in this thesis involves the noise properties of the flow, and as such is different than previous work by Ballard [4].

12.1.5 Interpretation of Constraint Clusters (BruteSac)

The concept of robust statistical clustering of linear and linear subspace constraints to perform motion segmentation is an important contribution of this work. As was shown in Chapter 7 mere segmentation of linear constraints into great circles may be insufficient for this task. It is further necessary to define localized clusters of linear constraints on the surface of the unit sphere, and to decide how to interpret them. The "BruteSac" method for generating hypotheses about underlying translational motions through pair-wise combination of constraint clusters was introduced, as well as methods for ordering these hypothesis based on their likelihood. These hypotheses

may form suitable input into higher-level motion interpretation systems in which contextual information or knowledge of constraints formed at the boundaries of IMOs are taken into account.

12.2 Future Directions

There are a number of directions for future work that can be suggested from this thesis. The first suggestion involves extending the method to the case of discrete displacement. There are a number of algorithms available for automatically tracking feature correspondences in image sequences [21]. A robust segmentation method that works on displacements of arbitrary size would make the method more general. Work is currently under way to extend subspace methods to the case of discrete displacement [88]. Any such work could be expected to have much in common with methods based on the essential matrix. It may be useful to consider the application of the EM algorithm to essential matrix generation in the presence of IMOs.

Chapter 7 considered the problem of combining the information inherent in constraint clusters in order to hypothesize about the underlying motion processes. This thesis presents a method in which clusters are spawned and merged as needed in order to represent the constraints in the data set. There may be advantages to considering instead a fixed number of "cluster-detectors." These detectors would tile the surface of the unit sphere with cluster distributions of the form of Eqn. 7.1. Each would have location and orientation vectors which remain fixed. The EM algorithm is then used to assign ownership of constraints to clusters, and perhaps allow the distribution parameters of any given cluster to vary within a certain range in order to allow for a better fit. The "BruteSac" methods of Chapter 7 could then be used to hypothesize about motions existing in the data, only considering clusters which had attracted some preset level of constraint ownership.

Still with respect to the clustering of constraints, it would be worthwhile to compare the methods described in this thesis with ones based on the concept of "self-organization" [52, 75, 38, 53]. Artificial neural networks which are capable of unsu-

pervised learning have been used to cluster data: the network is given the data set without *a priori* knowledge of which data belongs to which class, and "learns" to classify the data in a statistically meaningful way. These networks tend to classify data as belonging entirely to one class or another, although it may be possible to change this characteristic. (It is quite reasonable to allow for the possibility that a motion constraint may be consistent with more than one hypothesized motion.)

Recent work in computational vision has explored the use of context information in vision [23, 73, 41, 42]. There is reason to assume that motion interpretation could benefit from top-down contextual information. Consider, for example, the task of detecting IMOs in the forklift sequence. Assume the XZ plane is parallel to the floor (track the floor in any event, so it is possible to apply the appropriate transform to validate this assumption). The Y-axis will be vertical. IMOs can be expected (most probably) to move in the XZ-plane, *e.g.* vehicles, people. There may be some things that move primarily in the Y-direction, *e.g.* objects being hoisted by a crane. This may have lower *a priori* probability than objects which move in the XZ-plane. There may also be *a priori* knowledge of egomotion from transducers attached to the robot's locomotion system. This information may not be accurate enough for general navigation, but could be expected to provide a good initial guess for the egomotion parameters. This would also give us a head-start on the problem of predicting how a static environment would evolve. Developing the appropriate models for priors with respect to motion is an area that deserves attention.

With respect to the work in Chapter 11 there are two suggestions. The first concerns Eqn. 11.3. This equation assumes that there is no rotational component involved in the motion. For the image sequence considered this is not an unreasonable assumption, but may not be appropriate for all sequences. If an estimate for $\vec{\Omega}$ is available for each frame of the sequence, it should be straightforward to add a correction for rotation. This should improve the recovery of IMO information from RID curves.

The methods of Chapter 11 to robustly fit RID curves include the task of estimating the focal length of the camera. As was pointed out in that chapter's summary, this

is not strictly necessary. It should be possible to reformulate the problem to robustly track RID for image points while representing image structure using an affine or projective basis. This would decouple the method from the recovery of the camera's intrinsic parameters. This would generalize the method and make it more powerful for detecting deviations from rigidity.

12.3 Conclusion

Given the importance of motion to vision in general, the interpretation of visual motion is a highly significant problem. Vision systems designed to work in dynamic environments where navigation and collision avoidance are issues will need to be able to recover information about egomotion and IMOs within the visual field. This thesis has presented statistical methods for robust segmentation of monocular image sequences based on underlying 3-D motion. This segmentation is performed by clustering local constraints that involve translation alone, or translation together with rotation. There do exist situations in which these constraints will be insufficient to perform the required segmentation, in which case robust tracking of the evolution of depth structure together with an assumption of rigidity can be used to identify independent object motion. Results from synthetic and real image sequences were presented. The story does not end here. There remains much work to be done in this area, and some suggestions for directions this work might take have also been presented.

Appendix A

List of Symbols

$\vec{X}$: point in 3-D, camera-centred coordinates

$\vec{x}$: projection of $\vec{X}$ onto image plane

f : focal-length of imaging system

X_3 : component of $\vec{X}$ directed along the optical-axis ($\hat{z}$)

$\vec{V}$: velocity of 3-D point, defined as $\frac{d\vec{X}}{dt}$

$\vec{u}$: velocity of projected image point (in the image plane), defined as $\frac{d\vec{x}}{dt}$; also called the *motion field* component at $\vec{x}$

$\vec{v}$: depth-scaled projected velocity

$P(\vec{x})$: projection operator, onto the plane perpendicular to $\vec{x}$

$R(\vec{x})$: transformation from $\vec{V}$ to $\vec{u}$

$\vec{u}_T$: translational component of $\vec{u}$

$\vec{u}_\Omega$: rotational component of $\vec{u}$

$\vec{T}$: 3-D translational velocity

$\vec{\Omega}$: 3-D rotational velocity

$I(\vec{x},t)$: image intensity as a function of image position and time

$\nabla_{\vec{x}}$: spatial gradient operator with respect to image position

$R(\vec{x},t)$: filtered image response

$\rho(\vec{x},t)$: amplitude response of filtered image

$\varphi(\vec{x},t)$: phase response of filtered image

E : the *essential matrix*

$[T]_\times$: matrix cross-product operator for vector $\vec{T}$

$\vec{a}$, B : 3×1 vector and 3×3 matrix which define a *bilinear constraint*

K : number of sample points used to construct a *linear constraint*

$\vec{\tau}$: *linear constraint* on $\vec{T}$; a normalized 3×1 vector

w : weighting factor for $\vec{\tau}$

$\vec{c}$: interpolation coefficient vector for constructing $\vec{\tau}$

D : matrix constructed from $\vec{\tau}$ vectors in order to solve for $\vec{T}$

$E(\vec{T})$: objective function to be minimized in solving for $\vec{T}$

$\mathrm{E}(\ldots)$: expectation operator

I_3 : 3×3 identity matrix

λ : eigenvalue of D

$\tilde{D}$: D matrix with noise term added

$\hat{D}$: rescaling of $\tilde{D}$ to remove bias in estimates of $\vec{T}$

R : 3×3 rotation matrix

$||\ldots||$: euclidean norm of vector

$\chi(\vec{x})$: characteristic function indicating if image position $\vec{x}$ belongs to background

m or M : number of distributions in a mixture model

π_i : mixing proportion of ith distribution in a mixture

$p(\ldots)$: probability density function

σ : standard deviation, used in various distribution functions

s_{ij} : probability that ith observation belongs to jth distribution in a mixture model

γ : normalizing factor for spherical gaussian probability density function

$L(\ldots)$: likelihood function

$f(\vec{T}, \vec{\Omega})$: objective function minimized while solving for $\vec{T}$ and $\vec{\Omega}$ using *bilinear constraints*

$\vec{L}$: "location vector" for defining location of a cluster of linear constrints on the surface of the unit sphere

$\vec{L}_\perp$: a vector perpendicular to both $\vec{L}$ and $\vec{T}$

ρ : relative noise in optic flow

$\vec{n}$: noise component of optic flow vectors

C : covariance matrix for optic flow noise

α_i : parameters used in constant and affine flow models, and rational displacement model, for optic flow estimation

F : matrix constructed from image position of flow sampling points, used to construct coefficient vector $\vec{c}$

$\hat{n}$: unit vector normal to a surface

D : relative inverse depth

d : distance

β : constant acceleration factor

v_i : relative velocity

t_{adj} : time-to-adjacency

ΔD : difference between predicted and estimated relative inverse-depth

Appendix B

Glossary of Terms

IMO: independently moving object. (Page 3)

Egomotion: the apparent motion in an image caused by the observer moving in a static environment. (Page 4)

Motion Field: the motion of the projection of a scene point on the image plane, caused by the relative motion between the point and the observer. (Page 4)

Optic Flow: a practical attempt at measuring the motion field through analysis of a sequence of images. (Page 4)

Focal Length: for a planar receptor surface and a pinhole camera, this is the perpendicular distance from the image plane to the nodal point of the imaging system; for perspective projection this quantity defines where a point in the scene is imaged on the image plane. (Page 7)

Perspective Projection: an imaging model in which all light rays go through a single point. (called the nodal point) of the imaging system; this line of projection determines where on the imaging receptor a scene point projects to. (Page 8)

Orthographic Projection: an imaging model in which all light rays travel parallel to the optic axis of the imaging system, and strike the image plane at right angles. (Page 8)

FOE: focus-of-expansion. (Page 12)

Essential Matrix: a matrix which can be recovered by measuring point correspondences on a rigid body between pairs of images in a sequence; in the absence of noise it uniquely defines translational direction and rotation of the observer between the two views. (Page 30)

SNR: signal-to-noise ratio. (Page 48)

PDF: probability density function. (Page 55)

EM-Algorithm: the "expectation and maximization" algorithm is an iterative method for simultaneously solving for distribution parameters and ownership probabilities in a statistical mixture model. (Page 60)

DOG: difference-of-Gaussians. (Page 82)

RID: relative inverse-depth. (Page 130)

Bibliography

[1] Gilad Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans Pattern Analysis & Machine Intelligence*, 7(4):384–401, 1985.

[2] S. Ayer, P. Schroeter, and J. Bigün. Segmentation of moving objects by robust motion parameter estimation over multiple frames. In Jan-Olof Eklundh, editor, *Lecture Notes in Computer Science, Vol.801 Computer Vision—ECCV'94*, pages 316–327, Berlin Heidelberg, May 1994. Springer-Verlag.

[3] Ali Azarbayejani, Bradley Horowitz, and Alex Pentland. Recursive estimation of structure and motion using relative orientation constraints. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 294–299, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[4] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 48:57–86, 1991.

[5] John Barron, David Fleet, and S. Beauchemin. Performance of optical flow techniques. In *Proceedings of the 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Champaign, Il.*, pages 236–242, Los Alamitos, California, June 1992. IEEE Computer Society Press.

[6] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. on Computer Vision, ICCV-93*, pages 231–236, Berlin, Germany, May 1993.

[7] Michael J. Black. *Robust incremental optical flow.* PhD thesis, Yale University, Department of Computer Science, 1992.

[8] Michael J. Black and Allan Jepson. Estimating multiple independent motions in segmented images using parametric models with local deformations. In *Proceedings of the Workshop on Motion of Non-Rigid and Articulated Objects*, pages 220–227, Austin, Texas, November 1994.

[9] Andrew Blake and Andrew Zisserman. *Visual reconstruction.* The MIT Press, Cambridge, Massachusetts, 1987.

[10] Philippe Burlina and Rama Chellappa. Time-to-x: Analysis of motion through temporal parameters. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle Washington*, pages 461–468, Los Alamitos, California, June 1994. IEEE Computer Society Press.

[11] R.H.S. Carpenter. *Movements of the eyes.* Pion Ltd., London, 1977.

[12] João Costeira and Takeo Kanade. A multi-body factorization method for motion analysis. Short version of technical report CMU-CS-TR-94-220. Submitted to ICCV95.

[13] Niels da Vitoria Lobo. *Computing egomotion, shape and detecting independent motion from image motion.* PhD thesis, University of Toronto, Department of Computer Science, 1992.

[14] Trevor Darell and Alexander Pentland. Robust estimation of a multi-layered motion representation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 173–177, Princeton, New Jersey, October 1991.

[15] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[16] J.E. Dennis Jr. and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[17] R. Deriche, Z. Zhang, Q. T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In Jan-Olof Eklundh, editor, *Lecture Notes in Computer Science, Vol.801 Computer Vision—ECCV'94*, pages 567–576, Berlin Heidelberg, May 1994. Springer-Verlag.

[18] Zoran Durić, Azriel Rosenfeld, and Larry S. Davis. Egomotion analysis based on the frenet-serret motion model. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 703–712, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[19] A. Mark Earnshaw and Steven D. Blostein. Unbiased estimation of camera translation direction from optical flow using linear constraints. In preparation.

[20] John D. Enderle and J. Wolfe. Time-optimal control of saccadic eye movements. *IEEE Transactions on Biomedical Engineering*, BME-34(1):43–55, 1987.

[21] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* The MIT Press, Cambridge, Massachusetts, 1993.

[22] Olivier D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Lecture Notes in Computer Science, Vol.801 Computer Vision—ECCV'92*, pages 563–578, Berlin Heidelberg, May 1992. Springer-Verlag.

[23] Jacob Feldman, Allan Jepson, and Whitman Richards. Is perception for real? In progress.

[24] David J. Fleet. *Measurement of Image Velocity.* Kluwer Academic Publishers, Boston, Massachusetts, 1992.

[25] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[26] David J. Fleet and Allan D. Jepson. Stability of phase information. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 52–60, Princeton, New Jersey, October 1991.

[27] William T. Freeman and Edward H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[28] Watson Fulks. *Advanced calculus: An introduction to analysis.* John Wiley & Sons, New York, 3rd edition, 1978.

[29] Eleanor J. Gibson, James J. Gibson, Olin W. Smith, and Howard Flock. Motion parallax as a determinant of perceived depth. *Journal of Experimental Psychology*, 58(1):40–51, 1959.

[30] J.J. Gibson. *The perception of the visual world.* Houghton Mifflin, Boston, Massachusetts, 1950.

[31] Peter Hallett. Personal communication.

[32] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: the approach based on influence functions.* John Wiley & Sons, Inc., New York, 1986.

[33] K. J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 156–162, Princeton, New Jersey, October 1991.

[34] Simon Haykin. *Adaptive Filter Theory.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[35] David J. Heeger. Optical flow from spatiotemporal filters. In *Proc. Int. Conf. on Computer Vision, ICCV-87*, pages 181–190, 1987.

[36] David J. Heeger and Greg Hager. Egomotion and the stabilized world. In *Proc. Int. Conf. on Computer Vision, ICCV-88*, pages 435–440, 1988.

[37] H. Helmholtz. *Treatise on physiological optics*. Dover, New York, 1910.

[38] John Hertz, Anders Krogh, and Richard G. Palmer. *Introduction to the Theory of Neural Computation*. Sante Fe Institute Studies in the Science of Complexity. Addison-Wesley Publishing Company, Redwood City, California, 1991.

[39] Berthold Klaus Paul Horn. *Robot vision*. The MIT Press, Cambridge, Massachusetts, 1986.

[40] Michael Jenkin and Allan D. Jepson. Detecting floor anomalies. In *Proceedings of the British Machine Vision Conference*, September 1994.

[41] Allan Jepson and Whitman Richards. A lattice framework for integrating vision modules. *IEEE Transactions on Systems, Man, and Cybernetics*, 1992.

[42] Allan Jepson and Whitman Richards. What makes a good feature? in preparation, 1993.

[43] Allan D. Jepson and Michael J. Black. Mixture models for optical flow computation. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 760–761, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[44] Allan D. Jepson and David J. Heeger. Subspace methods for recovering rigid motion, part II: Theory. Research in Biological and Computational Vision RBCV-TR-90-36, University of Toronto, November 1990.

[45] Allan D. Jepson and David J. Heeger. A fast subspace algorithm for recovering rigid motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 124–131, Princeton, New Jersey, October 1991.

[46] Allan D. Jepson and David J. Heeger. Linear subspace methods for recovering translational direction. In L. Harris and M. Jenkin, editors, *Spatial Vision in Humans and Robots*. Cambridge University Press, 1993. See also: Research in Biological and Computational Vision, Department of Computer Science, University of Toronto, RBCV-TR-90-40, Apr. 1992.

[47] Kenichi Kanatani. *Geometric computation for machine vision*. Clarendon Press, Oxford, England, 1993.

[48] Kenichi Kanatani. Renormalization for unbiased estimation. In *Proceedings of the 4th International Conference on Computer Vision*, pages 599–606, Berlin, Germany, May 11–14 1993.

[49] Jan J. Koenderink. Optic flow. *Vision Research*, 26(1):161–180, 1986.

[50] Jan J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.

[51] Jan J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8(2):377–385, 1991.

[52] Teuvo Kohonen. *Self-Organization and Associative Memory*. Springer Series in Information Science. Springer-Verlag, Berlin, second edition, 1988.

[53] Arun D. Kulkarni. *Artificial Neural Networks for Image Understanding*. Van Nostrand Reinhold, New York, 1994. ISBN 0-442-00921-6; LofC QA76.87.K84 1993.

[54] Jonathan Lawn and Roberto Cipolla. Robust egomotion estimation from affine motion parallax. Technical Report CUED/F-INFENG/TR 160, Department of Engineering, University of Cambridge, January 1994.

[55] S. Lehman and L. Stark. Multipulse controller signals I: Pulse width and saccadic duration. *Biological Cybernetics*, 48:1–4, 1983.

[56] S. Lehman and L. Stark. Multipulse controller signals II: Time optimality. *Biological Cybernetics*, 48:5–8, 1983.

[57] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[58] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London B*, 208:385–397, 1980.

[59] Q.-T. Luong and T. Viéville. Canonic representations for the geometries of multiple projective views. In Jan-Olof Eklundh, editor, *Lecture Notes in Computer Science, Vol.801 Computer Vision—ECCV'94*, pages 589–599, Berlin Heidelberg, May 1994. Springer-Verlag.

[60] W. James MacLean. Early prediction of saccadic amplitude. Master's thesis, University of Toronto, Department of Electrical Engineering, 1989.

[61] David Marr. *Vision*. W. H. Freeman and Company, New York, 1982.

[62] S. J. Maybank. Properties of essential matrices. *International Journal of Imaging Systems and Technology*, 2:380–384, 1990.

[63] Suzanne P. McKee, Gerald H. Silverman, and Ken Nakayama. Precise velocity discrimination despite random variations in temporal frequency and contrast. *Vision Research*, 26(4):609–619, 1986.

[64] Geoffrey J. McLachlan and K. E. Basford. *Mixture models: Inference and applications to clustering*. Marcel Dekker Inc., New York, 1988.

[65] Radford M. Neal and Geoffrey E. Hinton. A new view of the EM algorithm that justifies incremental and other variants. Submitted to Biometrika, 1993.

[66] Shahriar Negahdaripour and Chi-Ho Yu. A generalized brightness change model for computing optical flow. In *4th International Conference on Computer Vision*, pages 2–11, Berlin, Germany, May 11–14 1993.

[67] Randal C. Nelson. Qualitative detection of motion by a moving observer. *International Journal of Computer Vision*, 7(1):33–46, 1991.

[68] Randal C. Nelson and J. Aloimonos. Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273, 1988.

[69] Gary D. Paige. The influence of target distance on eye movement responses during vertical linear motion. *Experimental Brain Research*, 77:585–593, 1989.

[70] Gary D. Paige and David L. Tomko. Eye movement responses to linear head motion in the squirrel monkey I. basic characteristics. *Journal of Neurophysiology*, 65(5):1170–1182, 1991.

[71] Gary D. Paige and David L. Tomko. Eye movement responses to linear head motion in the squirrel monkey II. visual-vestibular interactions and kinematic considerations. *Journal of Neurophysiology*, 65(5):1183–1196, 1991.

[72] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, England, 3rd edition, 1992.

[73] Allan D. Jepson W. Richards. What is a percept? Center for Cognitive Science Occasional Paper #43, Massachusetts Institute of Technology, April 1991.

[74] J. H. Rieger and D. T. Lawton. Processing differential image motion. *J Opt Soc Am A*, 2(2):354–359, 1985.

[75] David E. Rumelhart and James L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. The MIT Press, Cambridge, MA, 1986.

[76] Harpreet S. Sawhney, Serge Ayer, and Monika Gorkani. Model-based 2D & 3D dominant motion estimation for mosaicing and video representation. Submitted to ICCV '95.

[77] D. Sinclair. Motion segmentation and local structure. In *Proceedings of the 4th International Conference on Computer Vision*, pages 366–373, Berlin, Germany, May 11–14 1993.

[78] Ajit Singh. Incremental estimation of image flow using a Kalman filter. *Journal of Visual Communication and Image Representation*, 3(1):39–57, 1992.

[79] Richard Szeliski. *Bayesian Modeling of Uncertainty in Low-level Vision*. Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[80] Inigo Thomas, Eero Simoncelli, and Ruzena Bajcsy. Spherical retinal flow for a fixating observer. In *Proceedings of the Workshop on Visual Behaviours, Seattle Washington*, pages 37–44, Los Alamitos, California, June 1994. IEEE Computer Society Press.

[81] William B. Thompson, Pamela Lechleider, and Elizabeth R. Stuck. Detecting moving objects using the rigidity constraint. *IEEE Trans Pattern Analysis & Machine Intelligence*, 15(2):162–166, 1993.

[82] Carlo Tomasi and Takeo Kanade. Factoring image sequences into shape and motion. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 21–28, Princeton, New Jersey, October 1991.

[83] P. H. S. Torr. Outlier detection and motion segmentation. Ph.D. thesis, in preparation, 1994.

[84] P. H. S. Torr and D. W. Murray. Stochastic motion clustering. In Jan-Olof Eklundh, editor, *Lecture Notes in Computer Science, Vol.801 Computer Vision—ECCV'94*, pages 328–337, Berlin Heidelberg, May 1994. Springer-Verlag.

[85] Shimon Ullman. *The interpretation of visual motion*. The MIT Press, Cambridge, Massachusetts, 1979.

[86] Alessandro Verri and Tomaso Poggio. Motion field and optic flow: Qualitative properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):490–498, 1989.

[87] John Y. A. Wang and Edward H. Adelson. Layered representation for motion analysis. In *Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York*, pages 361–366, Los Alamitos, California, June 1993. IEEE Computer Society Press.

[88] Zhengyan Wang and Allan Jepson. A new closed-form solution for absolute orientation. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle Washington*, pages 129–134, Los Alamitos, California, June 1994. IEEE Computer Society Press.

[89] Joseph Weber and Jitendra Malik. Robust computation of optical flow in a multi-scale differential framework. In *4th International Conference on Computer Vision*, pages 12–20, Berlin, Germany, May 11–14 1993.

[90] Juyang Weng, Thomas S. Huang, and Narendra Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, 1989.