

Direct Passive Navigation

by

Shahriar Negahdaripour

BSME Massachusetts Institute of Technology

(1979)

MSME Massachusetts Institute of Technology

(1980)

Submitted to the Department of

Mechanical Engineering in

Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

In Mechanical Engineering

at the

Massachusetts Institute Of Technology

November 1986

©Massachusetts Institute of Technology

Signature of Author_____

Department of Mechanical Engineering

November 7, 1986

Certified by_____

Professor Berthold K. P. Horn

Thesis Supervisor

Accepted by_____

Ain A. Sonin

Chairman, Departmental Graduate Committee

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

MAR 09 1987

LIBRARIES

ARCHIVES

Direct Passive Navigation

by

Shahriar Negahdaripour

Doctoral Thesis

submitted to the Department of Mechanical Engineering
in November, 1986 in partial fulfillment of the
requirements for the degree of Doctor of Philosophy

Abstract

One of the primary tasks of a computer vision system is to reconstruct uniquely three-dimensional physical properties of a scene such as motion and shape from two-dimensional images of the scene. This thesis addresses the problem of recovering the motion of the camera directly from the time-varying images without the need to estimate the velocities of image points or establish point correspondence. The problem is usually under-constrained; yet, robust results can be obtained in special but not very restrictive cases. Some new results concerning the uniqueness of the solution are presented.

Thesis Supervisor: Professor Berthold K. P. Horn

Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I dedicate this to my parents, the two dearest individuals, who made everything possible.

Professor Berthold K.P. Horn, a pioneer in the area of computer vision, deserves my deepest gratitude for introducing the topic to me and supervising the work. He also made it possible for me to spend five months in Hawaii during which time the first draft of the thesis was written.

Professor Patrick Winston deserves a lot of credit for maintaining the MIT Artificial Intelligence Laboratory as one of the most exciting and thought-provoking research centers in the field.

I am thankful to Dr. Alan Yuille for collaboration in the earlier developments of the uniqueness results in the case of quadratic surfaces presented in Chapter 6. Professors Shimon Ullman, Tomaso Poggio, Ellen Hildreth, and Michael Brady kindly reviewed and commented on the earlier drafts of the papers on the results derived for planar surfaces.

I am grateful to Professors Leonard A. Gould and Derek Rowell (two members of my thesis committee), Drs. Philippe Brou and James Little, and Michael Gennert who reviewed the drafts of the thesis. My special thanks go to Michael Gennert who patiently went over the hairy equations. Because of this, I hold him responsible for any undetected errors in the equations (send bug reports to michaelg@oz); however, this will not be necessary since he did a thorough job.

My colleagues at the MIT Artificial Intelligence Laboratory deserve a lot of credit for many helpful discussions and suggestions during the course of my research.

I owe many thanks to Professor Edward J. Weldon and the other members of the Waikiki Surfing and Computer Vision Society, Department of Electrical Engineering, University of Hawaii, who made my trip to Hawaii a very memorable one.

This work was completed with financial support from the System Development Foundation, the Advanced Research Projects Agency of the Department of Defense under office of Naval Research, and the National Science Foundation.

Contents

Abstract	2
Acknowledgments	3
Contents	4
List of Figures	8
List of Tables	10
1 Introduction	11
1.1 Previous Work	13
1.2 Motivation	18
1.3 Goal of the Thesis	20
1.4 Summary of Results and Novel Contributions	21
1.5 Organization of the Thesis	23
2 Direct Passive Navigation: Problem Formulation	24
2.1 Perspective Projection	25
2.2 Rigid Body Motion	25
2.3 Motion Field for the Case of Rigid Body Motion	26
2.4 Optical Flow	27
2.5 Distinction Between Motion Field and Optical Flow	27
2.6 Computing Optical Flow	29

2.6.1	Imposing Smoothness Constraint	30
2.6.2	Parametrization Approach	32
2.7	Using Optical Flow to Recover 3-D Motion	33
2.7.1	Instability of the Solution Based on a Local Analysis	34
2.8	The Alternative: Direct Methods	35
2.9	Brightness Change Constraint Equation	35
2.10	Summary	36
3	Ill-Posed Nature of the Motion Vision Problem	38
3.1	Pure Rotation	39
3.2	Known Motion	40
3.3	Known Depth	40
3.4	Unknown Depth and Motion	41
3.4.1	Parametrization Approach	42
3.4.2	Regularization Approach	43
3.4.3	Depth Positivity as a Constraint	44
3.5	Summary	45
4	Planar Surfaces	46
4.1	A Frontal Plane	47
4.2	A Plane at any Arbitrary Orientation	48
4.2.1	Iterative Methods	49
4.2.1.1	First Scheme	50
4.2.1.2	Second Scheme	51
4.2.1.3	Division of Labor	52
4.2.1.4	Choosing an Initial Condition	53
4.2.2	Uniqueness	54
4.2.3	Closed-Form Solution	57
4.2.3.1	Essential Parameters of Planar Surfaces	57
4.2.3.2	Recovering Essential Parameters	59
4.2.3.3	Recovering Motion and Surface Parameters	61
4.3	Some examples	64
4.3.1	A Comparison of the Two Iterative Schemes	65
4.3.2	A Case of the Unique Solution for Planar Surfaces	65
4.3.3	Sensitivity to Image Noise: Iterative Methods	72
4.3.4	A Comparison of the Analytical and Iterative Methods	72
4.3.5	Sensitivity to Modeling Error and Image Noise: A Saddle Point	72
4.4	Summary	73
5	Quadratic Surfaces	81
5.1	Recovering Motion and Surface Parameters	83
5.2	An Iterative Solution Procedure	84
5.3	Division of Labor	85

5.4 An Example	86
5.5 Summary	86
6 Uniqueness of Motion	90
6.1 Surface Presentation	92
6.2 Essential Parameters of an n^{th} Order Surface	93
6.3 Computing the Essential Parameters	95
6.4 Plan of Action	96
6.5 Recovering Motion and Surface Parameters	98
6.5.1 Coordinate Transformation	98
6.5.2 The Case of $t_i \neq 0$ and $d_{ij} \neq 0$	101
6.5.2.1 The Case of $t_3 = 0$	101
6.5.2.2 Ambiguities of $t_3 = 0$	102
6.5.2.3 The Case of $t_3 \neq 0$	104
6.5.2.4 Ambiguities of $t_3 \neq 0$	104
6.5.2.5 Information in the Higher-Order Terms of the Motion Field	106
6.5.3 The case of $t_i = 0$ or $d_{ij} = 0$	109
6.6 Ambiguities in the Motion of Smooth Surfaces	111
6.6.1 Surfaces That Give Rise to an Ambiguity	112
6.6.2 Critical Surface Pairs	113
6.7 Relationship Among Multiple Solutions	115
6.7.1 Case one: Three-Fold Ambiguity of Hyperboloids of One Sheet	118
6.7.2 Case two: Two-Fold Ambiguity of Hyperboloids of One Sheet	121
6.8 Imposing Depth Positivity Constraint to Resolve Ambiguity	124
6.9 Some Experimental Results	129
6.9.1 An Example of a Three-Fold Ambiguity	129
6.9.1.1 Simulation Results	133
6.9.2 An Example of a Two-Fold Ambiguity	135
6.9.2.1 Simulation Results	137
6.9.3 The Ambiguity of a Second-Order Flow Analysis: Example one	138
6.9.3.1 Simulation Results	140
6.9.4 The Ambiguity of a Second-Order Flow Analysis: Example two	140
6.9.4.1 Simulation Results	142
6.9.5 A Non-Ambiguous Case	142
6.9.5.1 Simulation Results	143
6.10 Summary	143
7 Depth Positivity as a Constraint to Recover Motion	155
7.1 Pure translation or Known Rotation	156
7.1.1 Feature Points	157
7.1.2 Constraints Imposed by Brightness Gradient Vectors	158
7.1.2.1 Locating Focus of Expansion using Gradient Vectors	165
7.2 Unknown Rotation	169
7.3 Selected Examples	170

7.3.1 Example One: Focus of Expansion in the Image	171
7.3.2 Example Two: Focus of Expansion at Infinity	171
7.3.3 Example Three: Unknown Rotation	171
7.4 Summary	177
8 Application of Regularization to Motion Vision	178
8.1 Regularizing an Ill-Posed Problem	179
8.2 Imposing Surface Smoothness	180
8.2.1 Solution Through Variational Principles	181
8.2.2 Physical Interpretation of the Regularization Functionals	183
8.3 Recovering Motion and Depth: General Formulation	186
8.3.1 Recovering Motion	186
8.3.2 Recovering Depth	187
8.3.2.1 Finite Difference Approximation	189
8.3.3 Choice of the Regularization Parameter	191
8.4 An Iterative Scheme	192
8.4.1 Comments on Experimental Results	193
8.4.2 Known Motion	193
8.5 Summary	195
9 Summary and Concluding Remarks	197
Appendix	203
A: Non-Robustness of Certain Analytical Methods	203
B: Surface Properties Relevant to Ambiguities in Motion Vision	204
C: Relating Image and Surface Taylor Coefficients	212
D: Distance from a Point to a Conic Section	213
E: Point with Least Sum-Squared Distance from n Lines	216
F: Some Properties of Dyadic Products	217
Bibliography	219

List of Figures

2.1	Viewer-centered coordinate system	25
2.2	Corresponding iso-brightness contours in an image sequence	28
4.1	A simulation converging to the true solution using the first scheme	66
4.2	A simulation converging to the dual solution using the first scheme	67
4.3	A simulation converging to the true solution using the second scheme	68
4.4	A simulation converging to the dual solution using the second scheme	69
4.5	Results of a simulation where the solution is unique	70
5.1	Results of simulations with noise-less and noisy data.	87
6.1	The direction of translation and the two asymptotic lines of the surface form the characteristic lines of motion-structure ambiguity. The ambiguity in the interpretation of a motion field can be resolved using a large field of view	127
6.2	A motion field with three rigid body interpretations	128
6.3	The images of the boundaries of three hyperboloids of one sheet responsible for a three-fold ambiguity. The ambiguity in a motion field is usually restricted to a small region of the image. For a sufficiently large field of view, the ambiguity can be resolved by matching the depth discontinuity boundaries with the conic section of the true surface	131
6.4	A motion field with three rigid body interpretations in a local region. The discontinuity in the motion field is consistent with the depth discontinuity boundaries of the true surface	134

6.5	The boundaries of discontinuity for two hyperboloids of one sheet that give rise to the same motion field when viewed by an observer moving perpendicular to the viewing direction	136
6.6	The ambiguity in the interpretation of a motion field is usually restricted to local regions. Three surfaces undergoing different motions can give rise to motion fields that are similar up to second-order terms	139
6.7	An example of a two-fold ambiguity when the analysis is restricted to a local region of the image	141
7.1	True and spurious (assumed) FOE's define the positive and negative half-planes	160
7.2	Permissible and forbidden regions	161
7.3	Relationship between the size of the permissible region and the relative position of an image point with respect to the FOE constraint line	163
7.4	Constant probability loci	164
7.5	Expected shape of the negative depth cluster	165
7.6	Determining the location of the FOE from a FOE constraint line	167
7.7	Determining the location of the FOE from several FOE constraint lines	168
7.8	Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example one	172
7.9	Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example two	173
7.10	Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example three	175
B-1	Surfaces with negative, zero, and positive Gaussian curvature	211
B-2	A Surface consisting of points with negative, zero, and positive Gaussian curvature	212

List of Tables

4.1 Unique Solution of Planar Surfaces; Results of the First Iterative Scheme	75
4.2 Iterative Scheme -vs- Close-Form Method for Planes	76
4.3 Simulation Results of the Analytical Method for Planes; the true surface is quadratic with a saddle point	77
6.1 (a) Simulation Results of the Iterative Scheme after 120 Iterations	145
(b) Simulation Results of the Analytical Method	146
6.2 (a) Simulation Results of the Iterative Scheme after 100 Iterations	147
(b) Simulation Results of the Analytical Method	148
6.3 (a) Simulation Results of the Iterative Scheme after 60 Iterations	149
(b) Simulation Results of the Analytical Method	150
6.4 (a) Simulation Results of the Iterative Scheme after 60 Iterations	151
(b) Simulation Results of the Analytical Method	152
6.5 (a) Simulation Results of the Iterative Scheme after 100 Iterations	153
(b) Simulation Results of the Analytical Method	154

Introduction



Even though our visual system only registers two-dimensional images of the world around us it routinely provides us with a three-dimensional description of the environment almost instantaneously. This enables us to successfully interact with our environment even in situations that are new to us.

The task of reconstructing the three-dimensional physical properties of a scene from its two-dimensional images arises in object recognition, locomotion, robot navigation and path planning, and other industrial applications where a machine interacts actively with an environment with the aid of a vision system. The goal is to determine the relative or absolute depth of each point in the scene as well as its motion. This information can be used to perform segmentation using depth discontinuity and subsequently to build a description of the scene including the spatial arrangement and motion of objects in the scene.

There are many approaches in machine vision for recovering three-dimensional information about scenes. For example, one may use stationary cues as in stereopsis or shape from shading or texture. In stereo analysis, two images of the same scene taken from different viewpoints are used to recover the absolute depth map of the scene. The main difficulty is establishing correspondences between the two images of a three-dimensional point. Since the viewing geometry is known, the corresponding image points in two frames are restricted to lie on the so-called epipolar lines and, therefore, solving the correspondence problem is restricted to a one-dimensional search. In the shape from shading or texture methods, only the surface orientation can be recovered.

Alternatively, one may exploit motion cues to recover shape. The motion of an observer in a fixed environment or, equivalently, the motion of an object with respect to

a stationary observer induces the so-called optical flow that is the apparent motion of brightness patterns in the image. In many cases, the optical flow is a good estimate of the motion field which is the projection of the resulting three-dimensional velocity field onto the image. In such cases, the optical flow contains rich information about the three-dimensional structure of the scene and the relative motion between the observer and the scene and can be exploited to reconstruct the scene structure and the relative motion. Only the relative depth of the points or objects in the scene can be recovered since the motion field is not changed when depth and translational motion are multiplied by the same scale factor. Further, only the relative motion between the observer and the objects in the scene can be recovered.

In passive navigation, the objective is to recover the motion of an observer (camera) from the time-varying image of the stationary scene. It is well-known that when motion is known the environmental layout can be recovered as long as the translational component of motion is non-zero. Structure from motion is the problem of recovering the scene from the time-varying imagery. Once this is known, the motion can then be determined rather easily. Therefore, both problems address the same issue even though the objectives are different.

The methods considered so far to recover three-dimensional motion or structure depend on the visual input used. Four distinct types of approaches have been pursued, namely, discrete, differential, least-squares, and direct.

In the discrete approach, the displacements of selected features or points in a sequence of frames are used to recover their three-dimensional configuration and motion. To do this, dominant features or tokens such as grey-level corner points that are likely to be found in consecutive images are identified, a correspondence between these elements in successive images is then established and, finally, the three-dimensional motion and configuration of these isolated features is recovered. The minimum number of points required to recover three-dimensional motion depends on the number of image frames and the type of projection assumed (Meiri [1980]). For example, a minimum of five points in two perspective frames or four points in three views is usually sufficient to recover the motion uniquely. The results are less noise-sensitive when more points are used. This, however, requires the detection and matching of more prominent points or features that are likely to have matches in the next frames. These methods may be appropriate if the scene consists of polyhedral objects where dominant features such as corners can be detected relatively easily; however, they are not applicable when one is viewing objects with smooth surfaces lacking distinct feature points.

In the differential approach, the velocity of an image point (or velocities of several image points) and its (their) derivatives are used to recover the instantaneous translational and rotational velocities of the camera relative to the scene (or, conversely, the velocities of the objects in the scene relative to the viewer) as well as the local structure of the object viewed. The popular route to accomplish this task is based on the assumption

that the optical flow is an accurate estimate of the motion field. This, along with some other assumptions made about the surface viewed, is then used to formulate a well-posed problem to recover motion and the surface parameters.

In the least-squares approach, the optical flow at every image point is used. The solution is determined by minimizing the discrepancy between the optical flow data and that predicted by the solution over the whole region of the image.

Unfortunately, the available methods for determining optical flow do not give a good estimate of the true motion field since optical flow is not uniquely defined. More precisely, of the two components of this vector field, only the component in the direction of the brightness gradient can be computed from time-varying images. This is called the aperture problem. In order to estimate the component in the direction perpendicular to the brightness gradient, either additional image information or a priori constraints are necessary or some heuristic assumptions have to be made. For example, one may require that the solution be smooth since the optical flow varies smoothly in a region that corresponds to the image of a single smooth object. Usually, a priori knowledge of the type of surface viewed or the observer motion is required to compute an optical flow field consistent with the true motion field.

While the computation of the optical flow may allow for subsequent image segmentation or the estimation of three-dimensional motion and spatial configuration of objects in the scene, it is not necessary as an intermediate step for recovering motion and structure. Considering the fact that all existing methods for computing the local optical flow either incorporate heuristic assumptions or require a priori knowledge of the scene or motion and that they are computationally expensive, one can make an even stronger argument against using optical flow as a means of recovering motion.

In the direct approach, the image brightness derivatives or, more precisely, the constraint between the temporal and spatial gradients of the image function are used to recover the instantaneous motion of the viewer and the scene structure. There is no need to compute optical flow or establish correspondence; only the spatial and temporal derivatives of the brightness function that are estimated easily from a time-varying image sequence are needed. This approach is attractive not only computationally but also in the sense that the readily available information, the time-varying imagery, is used directly to estimate the desired output, the scene structure and motion.

1.1 Previous Work

Scientists have been studying the underlying mechanisms responsible for different functions of the biological vision systems for many centuries. Probably, one of the most vital functions of a biological system, at least in the case of humans, is the capability to estimate depth from the two-dimensional retina images. This allows us to interact with our surroundings even in unfamiliar environments.

It is well-known that motion as well as stereo provides cues for recovering depth. Psychologists have been interested in three-dimensional motion perception for several decades. Some of the earlier results of the studies or experiments on biological systems can be found in the psychology literature (e.g., Helmholtz [1925], Gibson [1950], Wallach & O'Connell [1953], Gibson [1957], Gibson et al. [1957], Gibson & Gibson [1957]). While it is very important to know how biological systems work or perform different visual functions from a scientific viewpoint, a computer vision system need not necessarily mimic a biological system.

Apparently, Helmholtz [1925] is the first to report that motion parallax contains rich information about depth of objects in a scene. Motion parallax arises when an observer moves relative to a scene. When the observer fixates on a point in the scene as he moves, the other stationary points in the scene seem to move with respect to the fixation point. The points further away than the fixation point seem to move with the viewer while the points closer to the viewer (than the fixation point) move in the opposite direction.

Even though the basic mathematics of image formation has been known to physicists for a long time, it was only a few decades ago that the underlying relationship between three-dimensional motion perception and two-dimensional image motion was investigated. Apparently, Gibson et al. [1957] gives the first mathematical study of motion perception using image motion field. They discovered that when an observer moves relative to an environment, the resulting motion field gives information on both the scene structure (described by Helmholtz [1925] as the motion parallax) and the observer motion (this was termed motion perspective by Gibson [1950]). They observed that apparent angular velocity of fixed points in the scene (the apparent rotational motion that results from the translation of the observer) is inversely proportional to the distance of the points from the viewer. Gordon [1965] studied the flow field generated by a moving car and developed the resulting mathematical equations. He observed that when the driver fixates on the foreground, the background appears to translate as well as rotate forward around it. When the viewer fixates on the background, the foreground seems to rotate in the opposite direction. He concluded that the difference between the angular velocities of the foreground and background were responsible for the resulting motion illusion. Hay [1966] presented a mathematical formulation of the relationship between the object motion and the resulting image motion field. His study focused on finding different object motions that can produce the same two-dimensional image motion field. This led to the discovery of the two-fold ambiguity associated with planar surfaces.

Koenderink & van Doorn [1975] studied the important effect of the local surface structure and the viewer motion on the resulting motion field as well as the invariant properties of the two-dimensional velocity field. They distinguish between propriospecific inputs that are determined through the observer motion and exterospecific inputs that are governed by the environmental layout irrespective of the movement of the viewer (Koenderink & van Doorn [1976,1981]). They hypothesized that organisms that rely

on their visual sense are sensitive only to the exterospecific component, unless they use the propriospecific component to monitor their own eye movements. For the case of planar surfaces, they show how the flow field is separated into two components, solenoidal (source-free) and lamellar (vorticity-free) fields, where the former is propriospecific and the latter is exterospecific.

Ullman [1979] incorporated the rigidity assumption in the interpretation of rigid body motion. More specifically, he stated that, given several interpretations of the configuration of points in space undergoing three-dimensional motion, one should choose the one (provided that it exists) consistent with rigid body motion. He proposed an algorithm for reconstructing the three-dimensional configuration of a discrete set of points from their two-dimensional images by determining the "most rigid" interpretation, one that minimizes the departure from rigidity as measured by the sum of distances between each two points as they undergo motion (Ullman [1984]). He also showed that three-dimensional motion can be recovered uniquely from three orthographic views of four non-coplanar points (Ullman [1983]).

In order to recover three-dimensional motion, one usually exploits a geometric constraint between the perceived motion in the image and the underlying three-dimensional motion of objects in the scene.

In the discrete approach, the governing equation between the position vector of a point in space before and after motion is exploited. The projection equation (either perspective or orthographic) is used to establish the constraint on the position of the corresponding image point in consecutive frames (before and after motion). Using the constraint equations from several points, the three-dimensional configuration of these points and their motion can be recovered uniquely. In most cases, correspondence of a minimum of five points in two perspective views or four points in three views are sufficient to recover motion uniquely even though the resulting non-linear equations are solved iteratively (for example, Roach & Aggarwal [1980], Barnard & Thompson [1980], Webb & Aggarwal [1982], and Dreschler & Nagel [1982]). Similarly, the velocities of five image points may be used to recover motion (for example, Prazdny [1979]). Assuming rigidity constraints, Mitiche [1984] presents a formulation that uses the velocity of four image points to recover motion. In either case, uniqueness cannot be shown. Fang & Huang [1984] show that nine points that do not lie on a second-order surface passing through the viewing point are sufficient to recover the motion uniquely (they observed that six points chosen randomly are sufficient in most cases). By ignoring some of the constraints, closed-form solutions have been obtained when a minimum of eight points in two frames are available (Longuet-Higgins [1981], Tsai & Huang [1984]). Tsai & Huang [1984] show that seven points are sufficient to determine the motion parameters uniquely from two perspective views; however, these points cannot lie on either two planes one passing through the viewing point or a cone containing the viewing point. For the case of planar surfaces, Tsai et al. [1982] found a closed-form solution from the correspondence of eight

image points (they also show that, at most, two solutions are possible). Longuet-Higgins [1984] presents a closed-form solution using the velocity of five image points. Buxton et al. [1984] show how the results obtained by Longuet-Higgins [1984] can be used to recover 3-D motion from the 2-D motion of eight edge features. When the motion is known, only three edge points are needed to recover the orientation of the planar surface.

The feature point matching schemes are computationally attractive; however, they are noise sensitive since information from a small portion of the image is used. Further, they require the identification and matching of prominent features or grey-level corners in a sequence of images. The existing methods for detecting grey-level corners as features and for solving the correspondence problem are non-robust and computationally expensive. Also, it is hard to find good feature points or corners in images of smooth objects.

Jain [1983] suggests a method for determining the location of the focus of expansion (where the axis of translation intersects the image plane) without establishing correspondence; however, it seems that it works best if matching points in two frames are employed and that the results deteriorate with non-matching points. Tsai [1983] shows how motion can be recovered using correspondence in two views of two planar conic sections; the two conic sections need not reside on the same planar surface patch. Yen & Huang [1983] present a method using correspondence of straight lines in three frames. Kanatani [1985a,1985b] has proposed a method using line and surface integrals for planar contours. Aloimonos & Basu [1986a] proposed a method using a four-camera set-up and contour correspondence. Aloimonos & Basu [1986b] present other results relevant to the case of a moving observer with respect to planar surfaces using binocular or monocular cues without establishing point correspondence. These methods seem more practical since it is easier to establish correspondence between contours in consecutive frames than between points.

In the differential formulation, the motion field and its derivatives are used to recover motion and the local structure of the scene. Here, it is assumed that a good estimate of the motion field in a local region is available. Longuet-Higgins and Prazdny [1980] suggested a method that uses the motion field and its first and second derivatives at a single point in order to recover motion and the local structure of the surface viewed (tangent plane and curvature parameters at the point of interest) from a set of 12 non-linear equations. Through an image coordinate transformation and elimination, they showed that the solution to a cubic equation may be used to compute the unknown parameters in closed form. Since they did not study the uniqueness of the solution of the cubic equation, it was concluded that, in general, three solutions are possible. Waxman and Ullman [1983] presented a new formulation by defining their own 12 observables in terms of the motion field and its first and second derivatives. Through an image coordinate transformation, they showed that the solution can be obtained in closed form for each possible transformation. The transformation angle, however, is determined through an iterative method from a non-linear equation. They illustrated uniqueness through a numer-

ical simulations. A closed-form solution to the 12 non-linear equations has been derived by Waxman et al. [1986]. They claim that, for these surfaces, two types of ambiguities, surface-related and motion-related, may occur. For example, surfaces with a negative Gaussian curvature in the local region of interest may give rise to a three-fold ambiguity in the solution or when the translational motion, the local surface normal, and the line of sight are in the same plane, a two-fold ambiguity is expected. They state that these ambiguities are instantaneous and can be resolved by incorporating temporal coherence (consistency in the solution obtained at different time instances). Negahdaripour & Yuille [1986] have shown that, for the case of locally quadratic surfaces, only two rare situations are potentially ambiguous (most of the the ambiguities observed by Waxman and colleagues are the shortcomings of a local second-order analysis of the image motion field). We will show that these ambiguities, in most cases, may be resolved if one imposes the constraint that the surface should lie on one side of the image plane; that is, the depth values computed for the whole image should have the same sign. Unfortunately, differential methods do not allow for errors in the data. More precisely, considering that the motion field cannot be estimated accurately and free of noise, any method based on information in the second-order derivatives of the flow field is bound to fail.

More robust algorithms have been proposed by using the velocity field from the whole or a relevant part of the image region through a least-squares formulation. It seems that these methods work best for either purely rotational or purely translational motion. Ballard and Kimball [1981] suggest a method using Hough transforms; however, they assume that depth is known. Bruss and Horn [1983] use a least-squares formulation to derive a set of non-linear equations that can be solved for the motion parameters. They give closed-form solutions for simple translational motion and simple rotational motion. Adiv [1985b] proposes segmenting the image into regions corresponding to planar patches on the surface. For each patch, he first computes the motion and surface parameters from the velocity field and then uses some criteria for combining patches with similar velocities and surface parameters. His method seems robust but requires extensive computation as well as several thresholds that have to be selected heuristically. Several methods for locating the focus of expansion and consequently determining the three-dimensional motion have been proposed. These methods exploit the well-known fact that the image velocity vectors radiate from the FOE (or point toward the FOC for a departing motion). Prazdny [1981] shows how to decompose the optical flow into its rotational and translational components. The translational component is extracted implicitly by locating the focus of expansion (or contraction for a departing motion). Lawton [1983] has incorporated a search process to locate the "best" feature points that are used to determine the location of the FOE. Jerian and Jain [1984] give a detailed presentation of the well-known properties of the FOE and suggest how a hypothesis and test approach may be employed to recover motion and structure by first locating the FOE.

Aloimonos and Brown [1984] present results based on using the brightness gradients

directly to recover motion. They consider the cases of a purely translational motion and a purely rotational motion. The case of a purely rotational motion is easily solved from three linear equations. For the purely translational motion, they derive a nonlinear (or linear if shape is known) constraint equation in terms of the second derivatives of the brightness function. They propose least-squares as well as Hough transform formulations. Negahdaripour & Horn [1985a,1985b,1986a,1986b] show how to recover the motion of an observer relative to a planar surface directly from image brightness derivatives iteratively as well as in closed form.

1.2 Motivation

The ultimate goal in computer vision is to design a system that enables a machine to interact with its environment successfully. To do this, the raw data in the form of two-dimensional images of the scene are first recorded by a camera system. Depending on the application, some type of processing is then performed on these images in order to extract three-dimensional information (such as motion, shape, and spatial arrangement of objects in the environment). Finally, in most cases, a higher level processor interprets the output from the first stage in order to present the information in a symbolic form. The final output is used to assist the machine in performing some task in the environment. This information has to be obtained in real time with as little computation as possible.

In the best cases, a vision system should consist of many simple modules that can essentially be studied independently of each other. For example, a module that is tuned to motion cues will require time-varying imagery to reconstruct the scene structure and a module tuned to binocular cues requires only stationary images. Analyzing or interpreting scenes does not necessarily have to be done by a single module but perhaps by the interaction of all the appropriate modules even though it is not yet clear how these modules should interact. Further, each module may consist of several sub-modules, each tuned to different cues, that have to interact. For example, to reconstruct three-dimensional motion and structure from monocular information, moving feature points, contours and surface markings, texture, and intensity variations all provide valuable information. Since these data are processed differently, it will be necessary to have different modules that are turned on when the appropriate cues present themselves in the visual field. Again, it is not yet clear how different sub-modules should interact when several of them become active. This, however, should not stop us from studying different modules or sub-modules independently provided that we are aware of their limitations when the appropriate input is not present in the visual field.

The problem of reconstructing motion and the scene structure from time-varying imagery has received a lot of attention in the last decade; however, no method has yet been suggested that uses the readily available data, that is, the time-varying image, directly. Most of the existing methods, based on the discrete, differential, or least-squares approaches, require some type of input that either is hard to compute or cannot be

computed accurately and reliably from image information.

Some of the disadvantages of the discrete methods are that

- (1) they require the detection of dominant feature points in consecutive frames,
- (2) they require the matching of the feature points in different frames,
- (3) they are noise-sensitive since information from a small portion of the image is used,
- (4) they are restricted to scenes that consist of polyhedral objects where feature points can be identified easily since it is difficult to find prominent features in the images of scenes with smooth objects, and
- (5) only the three-dimensional configuration of the feature points can be estimated; in order to obtain a full description of the scene, surface interpolation has to be performed.

The corner detection and the correspondence problems are difficult to solve. In general, these require heuristics either to establish criteria for selecting a point as a feature point (or grey-level corner) or to establish correspondence, especially when several candidate solutions are possible. The problem becomes even more difficult when more points are needed to reduce the sensitivity to noise. Usually, there is a trade-off between the number of feature points and the accuracy of the correspondence made; it is easier to establish correspondence when feature points are sparse. Another disadvantage of these methods is that it is important to use feature points that are well-separated or spread over the image in order to obtain reliable results since neighboring points tend to have similar velocities (or displacements). When several objects are present in the scene, it is likely that well-separated feature points belong to different objects that undergo different motions. Therefore, it may become important to segment the image first to guarantee that the feature points belong to either the same object or objects that undergo similar motion.

Differential methods use the optical flow and its first and second derivatives to recover motion and structure. These methods are non-robust since they use derivatives of an optical flow that is noisy and inaccurate. In addition, only local information is used. Several papers (for example, Ullman [1983], Jerian & Jain [1984], Adiv [1985a], Horn & Weldon [1986]) show why local information can give rise to ambiguities in the solution and, therefore, why methods that use local information are almost guaranteed to fail.

Methods based on a least-squares formulation require the motion field over the whole or relevant part of the image to recover three-dimensional motion. Unfortunately, no method is available that gives a sufficiently accurate estimate of the true motion field. From time-varying images, only the component of the optical flow (estimate of the motion field) in the direction of the brightness gradient may be estimated. To compute the full flow field, additional constraints, assumptions, or information is needed. In general, the

computed optical flow is not the same as the true motion field and, therefore, will not give a reasonable estimate of the three-dimensional motion or shape.

To summarize, while several methods have been suggested that may work under ideal conditions (that is, when the appropriate visual input is available and the data is noise free), no method exists that uses the readily available data, the brightness function in time-varying imagery, directly to recover three-dimensional motion and the scene structure. In developing a robust method, preferably based on a least-squares formulation, one should choose a presentation of the image data that is immediately accessible (or efficiently computable) from the time-varying imagery. This important issue is often overlooked in the analysis of the motion vision problem by the simplistic assumption that accurate measurement of the optical flow is available. This motivated a detailed investigation of alternative methods that do not require the computation of the optical flow as an intermediate step to solving the motion vision problem.

1.3 Goal of the Thesis

In this thesis, the problem of direct passive navigation is investigated. Passive navigation is the problem of recovering the motion of a viewer in a stationary environment from the changing images. In direct passive navigation, the brightness derivatives are used directly. There is no need either to compute optical flow or to solve a correspondence problem. This has the following advantages:

- (1) **Efficiency:** Only the brightness derivatives that are estimated from the readily available data, the time-varying imagery, are used. In many cases, it is not even necessary to save the images since the only information required is contained in certain moments of brightness derivatives that are computed from an initial pass over the image.
- (2) **Accuracy:** More accurate results are obtained when no heuristic assumptions are made and a least-squares formulation is used to exploit information from the whole image. In comparison, the optical flow based methods require heuristic assumptions in order to compute an optical flow that is then used to recover the motion parameters. In the discrete approach, several intermediate steps such as corner detection and correspondence problems require heuristics or setting threshold limits that have to be readjusted for different images.
- (3) **Simplicity in Implementation:** No intermediate computation is required. Therefore, only one module performs all the necessary computations. The other common approaches, discrete, differential, or least-squares require the solution of several intermediate problems. For example, in the discrete approach, features have to be identified first and matched next before the three-dimensional motion can be estimated. Similarly, the differential and least-squares methods require the computation of the optical flow as a first step.

It is important to use a least-squares formulation in order to exploit the information from the whole or relevant region of the image. This is due to the fact that image information from a local region can be highly ambiguous. Also we need a large field of view to recover certain motion parameters with good accuracy.

1.4 Summary of Results and Novel Contributions

A new formulation for solving the problem of passive navigation (structure from motion) is presented. It is based on using the brightness derivatives that are computed easily from time-varying images. There is no need to solve the optical flow problem as an intermediate step. No correspondence needs to be established.

First, the brightness change constraint equation in the case of rigid body motion,

$$E_t + \mathbf{v} \cdot \boldsymbol{\omega} + d \mathbf{s} \cdot \mathbf{t} = 0,$$

is derived. This equation is a constraint relationship between the velocity of the viewer, $\boldsymbol{\omega}$ and \mathbf{t} , the scene structure, $d = 1/Z$ (Z is the distance from the viewer to a point in the scene along the optical axis), the time derivative of the image brightness, E_t , and some moments of the image brightness gradient, \mathbf{v} and \mathbf{s} . It shows how the motion of a viewer in the scene imposes a constraint on the resulting time-varying imagery.

In general, motion cannot be recovered using the brightness change equation alone; more constraints are necessary. It is shown that the rotational parameters can be determined from three linear equations if there is no translation (the solution to this problem was first reported by Aloimonos & Brown [1984]). When the depth values for a minimum of six points are known, motion can be recovered by solving six linear equations. In practice, every available data point should be used. Similar equations have to be solved when the surface is a frontal plane; that is, all the points lie on a plane parallel to, but at an unknown distance from, the image plane. When the surface is planar with an unknown orientation, the motion parameters can be recovered from eight non-linear equations derived from a least-squares formulation. Two iterative methods for solving these equations are given and some simulation results are presented to show the convergence properties of these schemes as well as the sensitivity of the solution to noise in the data. A closed-form solution to this problem is also derived and it is explained why the analytical solution gives the same estimate as the iterative method. This is in contradiction with the results obtained by Scott [1986] using real images. He reports that his implementation of the iterative method gives better estimates of three-dimensional motion than that obtained from the closed-form solution. When the surface of the scene is quadratic, the motion parameters can be determined from 11 non-linear equations iteratively. A closed-form solution can be derived; however, it is generally non-robust in the presence of very little noise in the data. Through several examples, it is shown that the iterative method gives more robust and accurate estimates than the analytical one.

No assumptions have to be made about the surface of the scene if the motion is purely translational; only physical constraints are sufficient to recover motion directly from brightness gradient vectors. The translation of the viewer can be recovered simply by incorporating explicitly in the formulation the constraint that depth values are positive for all image points. This constraint means that the surface of the scene has to be in front of the camera for it to be seen. This method can be extended to the case when the rotation is not known; however, it requires more computation and can be non-robust because of the inherent ambiguities in the solution when data are noisy.

Recently, regularization theory has been used to formulate several under-constrained vision problems as a well-posed problem in the calculus of variations (see Poggio & Torre [1984] for a review). This requires imposing smoothness on the solution. In the present context, this is smoothness of the surface of the scene. It is shown that the problem of direct passive navigation can be formulated as a well-posed problem using regularization theory. Applying the necessary conditions for optimality and methods of variational principles, the problem reduces to solving a set of differential and algebraic equations iteratively. Unless a good initial guess of the solution is available, the method requires extensive computation and, in general, does not converge to an acceptable solution.

An important issue in recovering three-dimensional motion from time-varying images is whether or not the solution is unique. It is shown that ambiguities in the interpretation of a given motion field occur rarely and are restricted to planar surfaces and certain quadratic surfaces. The ambiguity of planar surfaces is at most two-fold (apparently, this was first observed by Hay [1966]). Some quadratic surfaces may give rise to a two-fold or three-fold ambiguity in very special cases; for example, certain hyperboloids of one sheet can give rise to three interpretations or certain cylindrical surfaces can give rise to two interpretations. These results are more general than those previously derived by Longuet-Higgins & Prazdny [1980], Waxman et al. [1986], and Negahdaripour & Yuille [1986] and are consistent with recent results derived by Horn [1986b] using a general representation of surfaces. Previous results derived by Tsai & Huang [1984] seem to be only some of the degenerate cases as shown by Horn [1986b]. It is shown that the correct interpretation can be identified when the field of view is sufficiently large. The minimum requirement on the size of the field of view can be determined analytically and is derived for a special case that gives rise to a three-fold ambiguity. An equally important result is the relationship among multiple interpretations, when they exist. This is useful if the shape and motion are determined from an iterative method; an iterative method can converge to only one solution per simulation. These results have practical applications not only in motion vision applications but other applications where the relative distance and orientation of two camera stations are to be determined from the images taken at the two stations, for example, as in photogrammetry, camera calibration, or stereo analysis.

1.5 Organization of the Thesis

In Chapter 2, the brightness change constraint equation is derived. This constraint equation shows how the motion of the viewer imposes a constraint on the temporal/spatial variation of the brightness of every image point. In Chapter 3, it is shown that the problem of recovering motion from time-varying images is generally underconstrained; however, a unique solution can be obtained in special cases. For example, the rotational motion parameters can be recovered from three linear equations if there is no translation. When the depth values for a minimum of six points are known, the motion is determined from six linear equations. In practice, every available data point should be used. Similar equations have to be solved when the surface is a frontal plane; that is, all the points are on a plane parallel to but at an unknown distance from the image plane. The results presented in Chapter 4 include two solution methods, iterative and closed-form, for the case of planar surfaces. Numerical simulations are also presented to demonstrate the convergence properties of the iterative schemes and the sensitivity of the solution to noise in the data. It is shown that the problem generally has two solutions unless the translation is in the direction perpendicular to the surface. In Chapter 5, an iterative method is proposed for estimating the motion parameters when some portion of the scene can be represented as a quadratic surface. This method requires the solution of 11 non-linear equations. In Chapter 6, the uniqueness of the solution of the motion vision problem for curved surfaces is studied. First, it is shown that only certain quadratic surfaces can give rise to an ambiguity in the solution; the ambiguity may be either a two-fold or a three-fold one. Then, the relationship among multiple solutions, when they exist, is derived. It is shown that, with a large field of view, the correct interpretation, when more than one exists, can be identified by imposing the depth positivity constraint. In Chapter 7, the positivity of depth is incorporated explicitly in a method for recovering the translation of the viewer from the brightness gradient vectors. It is shown how the method may be extended to recover both the rotational and translational vectors. In Chapter 8, a formulation based on the regularization method is presented. Finally, a summary of the results, the conclusions, and recommendations for future work are presented in Chapter 9.

Direct Passive Navigation: Problem Formulation



In most cases, the motion of an observer with respect to a stationary scene will give rise to time-varying imagery. An exception is, for example, provided by images of flat diffuse objects with no texture or surface markings. Other scene events, like moving shadows due to the motion of the light source, can produce time-varying imagery not related to the observer motion; however, the variations due to changes in shading are usually negligible in comparison to those due to motion. We expect then that image brightness variations are most affected by the observer motion and the scene structure. The motivation behind the approach we use here is that the brightness variations, both spatial and temporal, can be estimated easily from a time-varying image sequence. These variations can then be used to estimate the motion of the viewer as well as the scene structure.

In this chapter, we derive the brightness change constraint equation for the case of rigid body motion. This equation shows how, under favorable conditions, the spatial and temporal brightness variations are constrained by the observer motion and the scene structure at each point in the image. Consequently, each point in the image provides one constraint on the motion parameters and the scene structure. In later chapters, we show how viewer motion and scene structure can be determined using the constraint equations from the whole image or some region of interest.

We will first review some details relating to perspective projection, rigid body motion, the motion field, and optical flow. We present some selected examples from a variety of methods proposed in the literature for computing a unique optical flow from time-varying images and for estimating three-dimensional motion from optical flow. We describe the inherent difficulties associated with computing the local optical flow field and the shortcomings of methods for estimating 3-D motion from optical flow. We then

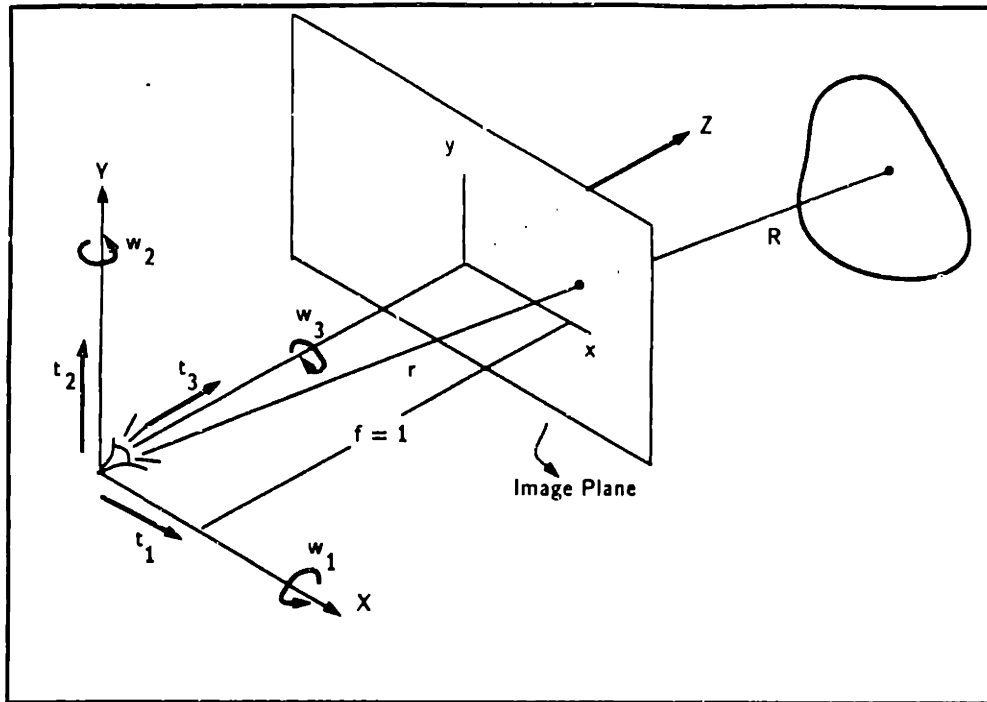


Figure 2-1. Viewer-centered coordinate system.

derive the brightness change constraint equation that is the basis of the work presented in later chapters on recovering three-dimensional motion from time-varying imagery. Vector notation is used in order to keep the resulting equations as compact as possible.

2.1 Perspective Projection

Let the center of projection be at the origin of a Cartesian coordinate system. Without loss of generality, it is assumed that the effective focal length is unity. The image is formed on the plane $z = 1$; that is, the optical axis lies along the z -axis (see figure 2-1). Let \mathbf{R} be a point in the scene. Its projection in the image is \mathbf{r} , where

$$\mathbf{r} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}.$$

The z -component of \mathbf{r} is clearly equal to one since $\mathbf{r} \cdot \hat{\mathbf{z}} = 1$.

2.2 Rigid Body Motion

When an observer moves relative to a rigid environment with translational velocity \mathbf{t} and rotational velocity $\boldsymbol{\omega}$, a point \mathbf{R} in the environment seems to move relative to the observer with velocity

$$\mathbf{R}_t = -\boldsymbol{\omega} \times \mathbf{R} - \mathbf{t},$$

where \mathbf{R}_t denotes the 3-D instantaneous velocity of the point \mathbf{R} . Since $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$, we can write

$$\mathbf{R}_t = -(\mathbf{R} \cdot \hat{\mathbf{z}})(\boldsymbol{\omega} \times \mathbf{r}) - \mathbf{t}.$$

2.3 Motion Field for the Case of Rigid Body Motion

As a point in the scene moves, the point that it projects onto in the image also moves. The *motion field* is the vector field that is induced in the image plane by the motion of the observer with respect to the scene. It is a purely geometric concept which is uniquely defined in terms of the observer motion and the scene structure.

The motion field equation can be determined for any type of projection, perspective (central) or orthographic (parallel), by differentiating the appropriate projection equation with respect to time. In this study, only perspective projection is considered.

The velocity \mathbf{r}_t of an image point \mathbf{r} is given by

$$\mathbf{r}_t = \frac{d}{dt} \left(\frac{\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \right) = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{R}_t - \frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} (\mathbf{R}_t \cdot \hat{\mathbf{z}}) \mathbf{R}.$$

Since $\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = (\mathbf{c} \cdot \mathbf{a})\mathbf{b} - (\mathbf{a} \cdot \mathbf{b})\mathbf{c}$, this equation can be written in the compact form

$$\mathbf{r}_t = \frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} (\hat{\mathbf{z}} \times (\mathbf{R}_t \times \mathbf{R})).$$

The vector \mathbf{r}_t is parallel to the xy -plane since $\mathbf{r}_t \cdot \hat{\mathbf{z}} = 0$. Further, $\mathbf{r}_t = 0$ if $\mathbf{R}_t \parallel \mathbf{R}$, as expected.

In the case of rigid body motion, one can substitute the expression for \mathbf{R}_t into the above equation to obtain

$$\mathbf{r}_t = -\frac{1}{(\mathbf{R} \cdot \hat{\mathbf{z}})^2} \left(\hat{\mathbf{z}} \times ((\boldsymbol{\omega} \times \mathbf{R}) \times \mathbf{R} + \mathbf{t} \times \mathbf{R}) \right).$$

Using $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$ we arrive at the perspective motion field equation for the case of rigid body motion

$$\mathbf{r}_t = -\left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\mathbf{r} \times \boldsymbol{\omega} - \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{t} \right) \right) \right).$$

There is an inherent ambiguity here since the same motion field results when distance and the translational velocity are multiplied by the same constant factor. This can be seen easily from the above equation since the same image plane velocity is obtained if one multiplies both $Z = (\mathbf{R} \cdot \hat{\mathbf{z}})$ and \mathbf{t} by some constant factor. Therefore, the translational motion and the depth values can be recovered only up to a scale factor from the image motion field. This will be referred to as the *scale-factor ambiguity* of the motion vision problem.

2.4 Optical Flow

Brightness patterns in the image move when the objects that give rise to them move. *Optical flow* is defined as the apparent (or perceived) motion of the image brightness patterns. More precisely, an optical flow field is a vector field that shows how the brightness patterns in the image at one instant of time can be transformed into the patterns at the next instant of time. As we see later, optical flow is generally non-unique.

The brightness of the image of a particular patch of a surface depends on many factors. For example, it may vary with the orientation of the patch or the direction of the light source. In many cases, however, it remains at least approximately constant as the surface moves in the environment (Horn [1986a]). Let $E(\mathbf{r}, t)$ denote the brightness of a point \mathbf{r} in the image, the projection of point \mathbf{R} on the surface, at time t . At some later time $t + dt$, the point \mathbf{R} will project onto the point $\mathbf{r} + d\mathbf{r}$ in the image. Assuming that the image brightness of the patch remains constant we can write

$$E(\mathbf{r} + d\mathbf{r}, t + dt) = E(\mathbf{r}, t).$$

If the image brightness function varies smoothly, the left-hand side of the above equation can be expanded in a Taylor series form,

$$E(\mathbf{r} + d\mathbf{r}, t + dt) = E(\mathbf{r}, t) + \frac{\partial E}{\partial \mathbf{r}} \cdot d\mathbf{r} + \frac{\partial E}{\partial t} \cdot dt + O(e) = E(\mathbf{r}, t),$$

where $O(e)$ denotes the higher order terms in $d\mathbf{r}$ and dt . The above equation after canceling $E(\mathbf{r}, t)$ from both sides and dividing by dt simplifies to

$$\frac{\partial E}{\partial \mathbf{r}} \cdot \frac{d\mathbf{r}}{dt} + \frac{\partial E}{\partial t} = 0.$$

Using the notation $E_{\mathbf{r}}$ for the spatial derivative and E_t for the time derivative of the brightness, the above equation can be written

$$E_{\mathbf{r}} \cdot \mathbf{r}_t + E_t = 0.$$

This equation is sometimes referred to as the *image brightness continuity equation* derived under the constant brightness assumption. Assuming brightness constancy, optical flow can be defined as any vector \mathbf{r}_t that satisfies the brightness continuity equation.

2.5 Distinction Between Motion Field and Optical Flow

The distinction between the motion field and optical flow is an important one that is rarely made; that is, it is commonly assumed that optical flow is the same as the motion field.

The motion field is the vector field obtained by projection of the velocities of points in the scene. It is determined entirely by the scene structure and the motion of the observer

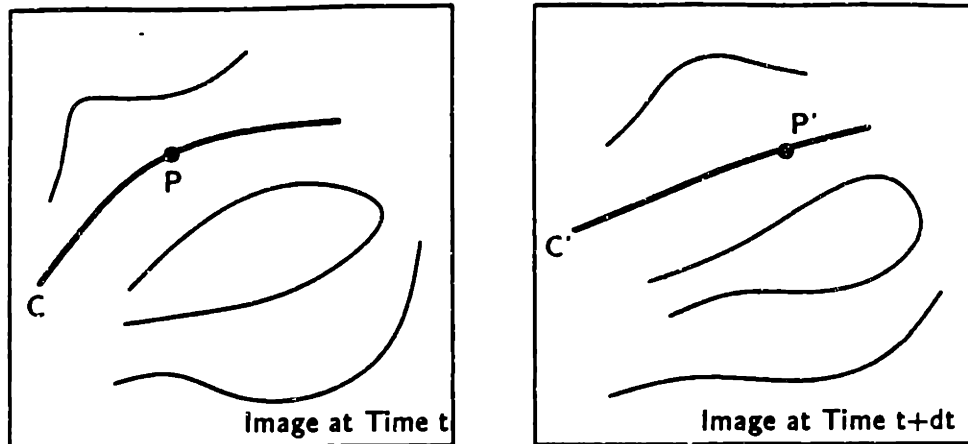


Figure 2-2. Corresponding iso-brightness contours in an image sequence.

relative to the scene. The motion field can be found by determining the velocities of the intersection points with the image of rays connecting the projection center to points in the environment. Knowing the motion field, in most cases, allows us to recover 3-D motion uniquely; however, the best we can do is to compute a particular optical flow from an image sequence. It is commonly assumed that this is an accurate estimate of the motion field. Optical flow is any vector field that is consistent with the brightness change equation,

$$E_r \cdot r_t + E_t = 0.$$

There are an infinite number of optical flows, since, if r_t is an optical flow, so is

$$r'_t = r_t + f(x, y) (E_r \times \hat{z})$$

for any $f(x, y)$, where \hat{z} is a unit vector perpendicular to the image plane. First, note that $r'_t \cdot \hat{z} = r_t \cdot \hat{z} = 0$, as it should. Furthermore, we have

$$E_r \cdot r'_t + E_t = E_r \cdot r_t + f(x, y) (E_r \cdot (E_r \times \hat{z})) + E_t = E_r \cdot r_t + E_t = 0.$$

To explain this graphically, consider the two images in figure 2-2. Suppose that C' is a contour of constant brightness in the second image corresponding to contour C in the first image. It is not easy to decide which point P' on C' corresponds to a particular point P on C since the contour changes shape as viewed by a moving observer (because of perspective geometry). In fact, there are many possible ways to establish correspondence between points on contours C and C' . This ambiguity has been referred to as the *aperture problem*. In terms of the iso-brightness contours, any vector field that transforms contour C into contour C' is an acceptable optical flow.

Typically, out of the infinite number of possible optical flows, one chooses a particular one using some heuristic criterion. This may be some form of smoothness criterion or minimization of some penalty function. One may hope to obtain a vector field that approximates the apparent motion of the brightness patterns in the image as judged by a human observer. Under unfavorable circumstances, this may be very different from the true motion field. Consider a perfectly smooth and uniform rotating billiard ball or the motion of shading on a stationary object due to a moving light source (these examples have been taken from Horn [1986a], pp.280-281). In the first case, the curved surface of the ball gives rise to shading or brightness variations in the image. The shading in the image, however, does not move with the surface, and so the image does not change with time. The most logical interpretation may then be a zero optical flow everywhere. In the second case, the motion of shading gives rise to a nonzero optical flow field even though there is no relative motion between the viewer and the object.

To avoid the type of ambiguities described above, we need to assume that the surface of the scene has strong texture or surface markings and the lighting condition remains unchanged from one image to the next one. Then the image brightness variations are affected primarily by the observer motion and the scene structure.

2.6 Computing Optical Flow

The optical flow constraint equation can be written

$$E_r \cdot r_t = -E_t.$$

This equation imposes one constraint on the two components of optical flow. Only the component in the direction of the brightness gradient is fixed:

$$r_t \cdot g = -\frac{E_t}{\|E_r\|},$$

where g denotes the unit vector in the gradient direction,

$$g = \frac{E_r}{\|E_r\|}.$$

We cannot, however, determine the component in the direction perpendicular to the brightness gradient; that is, along an *iso-brightness contour*. In order to estimate this component, we need additional information or we have to impose heuristic constraints. An example is the “minimal” optical flow, obtained by retaining only the component in the direction of the brightness gradient. Alternatively, we may impose a smoothness assumption through minimization of some criterion function or some type of polynomial fit. We will briefly describe two common approaches for computing a unique optical flow which ideally serves as the best estimate of the true motion field.

2.6.1 Imposing Smoothness Constraint

The motion field varies smoothly in most parts of the image. The exception is at an occluding boundary where a depth discontinuity gives rise to a discontinuity in the velocity field. Therefore, the motion field is smooth in areas of the image where depth is continuous. Based on this fact, we may require that the optical flow should vary smoothly in neighboring regions. Smoothness can be imposed through minimizing a functional that is a measure of the required degree of smoothness. For example, we may choose to minimize the integral of the square of the magnitude of the gradient of the optical flow. The gradient of the optical flow is

$$\nabla \mathbf{r}_t = \frac{\partial \mathbf{r}_t}{\partial \mathbf{r}} = \begin{pmatrix} \frac{\partial(\mathbf{r}_t \cdot \hat{\mathbf{x}})}{\partial x} & \frac{\partial(\mathbf{r}_t \cdot \hat{\mathbf{x}})}{\partial y} & 0 \\ \frac{\partial(\mathbf{r}_t \cdot \hat{\mathbf{y}})}{\partial x} & \frac{\partial(\mathbf{r}_t \cdot \hat{\mathbf{y}})}{\partial y} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

So we minimize

$$E_s = \iint_I \|\nabla \mathbf{r}_t\|_2^2 dx dy.$$

Here $\|\cdot\|_2^2$ denotes the Euclidean or Frobenius norm of a matrix which is the sum of the square of all the elements of the matrix (Korn & Korn [1968], pp. 404). To make sure that the error in the optical flow constraint equation is small we should also minimize

$$E_c = \iint_I (E_r \cdot \mathbf{r}_t + E_t)^2 dx dy.$$

All together, we minimize

$$J = E_s + kE_c,$$

where the parameter k weights the error due to each term. It is chosen to be large when the data are accurate and small when brightness values are noisy. This is the basis of the method for computing optical flow proposed by Horn & Schunck [1981].

Hildreth [1983] investigated a similar formulation but incorporated different measures of smoothness. The computation was restricted to be along the zero-crossing boundaries (where the convolved image with the Laplacian of a Gaussian operator crosses zero). The problem is formulated as that of minimizing, along a zero-crossing contour, the functional

$$J = E_s + kE_c,$$

where

$$E_c = \int_{\partial} \left(\mathbf{r}_t \cdot \mathbf{g} + \frac{E_t}{\|E_r\|} \right)^2 dl.$$

As before, \mathbf{g} is the unit vector in the direction of the brightness gradient (along a zero-crossing, this is the same as the normal to the zero-crossing) and dl is the arc length

along the zero-crossing. Again, E_s is some measure of smoothness. For example, the local variations in the magnitude of the velocity vectors are minimized by choosing

$$E_s = \int_{\partial} \left\| \frac{\partial \mathbf{r}_t}{\partial t} \right\|^2 dl,$$

or the local variations in the direction of the velocity vector are minimized if we choose

$$E_s = \int_{\partial} \left(\frac{\partial \phi}{\partial t} \right)^2 dl,$$

where ϕ is the direction of the velocity vector. It is worth mentioning that Rougée et al. [1986] have derived results very similar to those of Horn & Schunck [1981] and Hildreth [1983] using formulations based on theories in dynamic systems and estimation.

Other methods based on variations of the smoothness assumption have also been proposed (for example, Fennema & Thompson [1979], Prager & Arbib [1983], Paquin & Dubois [1983], Davis et al. [1983], Yashida [1983], Anandan [1984]). Cornelius and Kanade [1983] incorporate temporal as well as spatial smoothness in their formulation by introducing the additional term E_b into the error function J given in Horn & Schunck [1981], where

$$E_b = \iint_I \|\nabla E_t\|^2 dx dy.$$

The methods discussed above use a linear constraint equation in the velocity space; that is,

$$E_r \cdot \mathbf{r}_t + E_t = 0.$$

Nagel [1983a,1983b] suggested a method that incorporates second order effects in order to obtain a better estimate around edges and corners. In his formulation, he uses the constraint equation,

$$E_r \cdot \mathbf{r}_t + \frac{1}{2} \mathbf{r}_t^T E_{rr} \mathbf{r}_t + E_t = 0,$$

where

$$E_{rr} = \begin{pmatrix} \frac{\partial^2 E}{\partial x^2} & \frac{\partial^2 E}{\partial x \partial y} & 0 \\ \frac{\partial^2 E}{\partial x \partial y} & \frac{\partial^2 E}{\partial y^2} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Here, $()^T$ denotes the transpose operation.

The true motion field is uniquely defined in terms of the observer motion and the scene structure. It seems, therefore, more appropriate to incorporate some knowledge of the motion or the scene structure than to impose some measure of smoothness blindly in order to compute an optical flow. This does not imply that the flow field computed using the smoothness assumption has no practical application. It may, for example,

be used to perform image segmentation since discontinuities in the motion field usually correspond to depth discontinuities. Even though a smooth optical flow field will not be discontinuous, it is likely to undergo sharp changes at the occluding boundaries. Hence, it may be still possible to obtain an estimate of the location of these boundaries from the resulting velocity field; however, the accuracy of the results will depend on the degree of smoothness that is imposed by the smoothness term.

2.6.2 Parametrization Approach

The observer motion and the scene structure uniquely define the motion field. If we assume a polynomial form for the surface of the scene locally, we can then express the motion field corresponding to the image of the local region in a polynomial form. It is then possible to estimate the coefficients of the motion field from the measurements of the component of the flow in the direction of the brightness gradient. Consequently, the full flow field can be computed for the local region accurate up to measurement errors and the validity of the parametric model of the surface of the scene.

For planar surfaces, the motion field has a quadratic form with eight independent coefficients. Murray & Buxton [1984] and Waxman & Wohn [1984] exploited this to estimate the coefficients of the motion field for planar surfaces from the measurements of the component of the optical flow normal to edge segments or boundaries, v_g . The formulation involves minimizing the function

$$E = \int_{\partial} (\mathbf{r}_t \cdot \mathbf{g} - v_g)^2 dl,$$

where

$$\mathbf{r}_t = \begin{pmatrix} a_1 + a_2x + a_3y + \frac{1}{2}a_7x^2 + a_8xy \\ a_4 + a_5x + a_6y + a_7xy + \frac{1}{2}a_8y^2 \\ 0 \end{pmatrix}.$$

For curved surfaces that can be approximated locally by a quadratic surface, Wohn [1984] gives an extension of this approach for estimating the coefficients of the local motion field up to the second-order terms. In this case, 12 coefficients have to be determined; that is,

$$\mathbf{r}_t = \begin{pmatrix} a_1 + a_2x + a_3y + \frac{1}{2}a_7x^2 + a_8xy + \frac{1}{2}a_9y^2 \\ a_4 + a_5x + a_6y + \frac{1}{2}a_{10}x^2 + a_{11}xy + \frac{1}{2}a_{12}y^2 \\ 0 \end{pmatrix}.$$

We will present some results to show that we cannot hope to obtain an accurate estimate of 3-D motion based on velocity measurements in a small region of the image (small field of view). Therefore, it seems unlikely that the optical flow that is computed based on a parametric model in a small neighborhood can give an accurate estimate of 3-D motion.

2.7 Using Optical Flow to Recover 3-D Motion

In Passive Navigation, the goal is to recover the motion of an observer from a sequence of time-varying images. To do this, we exploit a constraint equation between the available data, the 3-D motion parameters, and the scene structure.

A common approach has been to assume that an accurate estimate of the motion field is available at a few isolated image points (discrete methods), in a local region (differential methods), or in the whole image (least-squares methods). The unknown parameters are then determined in a way most consistent with the available data. For example, by eliminating Z from the two components of the motion field,

$$\mathbf{r}_t = -\left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\mathbf{r} \times \boldsymbol{\omega} - \frac{1}{Z}\mathbf{t}\right)\right)\right),$$

we can derive a non-linear constraint equation in terms of the six motion parameters (three components of each of $\boldsymbol{\omega}$ and \mathbf{t}). This includes five independent unknowns since only the direction of the translation vector can be determined because of the scale-factor ambiguity. Ideally, the unknown motion parameters can be determined using the velocity of five image points (for example, Prazdny [1979]). Another method, first suggested by Longuet-Higgins & Prazdny [1980], is to use the optical flow and its *first* and *second* derivatives at a single point. This gives 12 non-linear constraint equations in terms of 11 unknown parameters, six motion parameters and six surface parameters (the local tangent plane and curvature parameters) that can be recovered only up to the scale-factor ambiguity. Analytical solutions for solving these equations have been derived (Longuet-Higgins & Prazdny [1980], Waxman & Ullman [1983], Waxman et al. [1986], Negahdaripour & Yuille [1986]). Unfortunately, methods that rely on the information from a small region of the image do not allow for noise in the data and, therefore, are not practical for implementation. Instead, we need to exploit the information from the whole image region (as in Bruss & Horn [1983] or Adiv [1985b]). We can formulate a least-squares problem where we minimize the discrepancy between the optical flow data and that predicted by the solution

$$\iint_I \left\| \mathbf{r}_t + \left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\mathbf{r} \times \boldsymbol{\omega} - \frac{1}{Z}\mathbf{t}\right)\right)\right) \right\|^2 dx dy.$$

Even though the least-squares methods give more robust results, they rely on the simplistic assumption that a good estimate of the motion field is available over the whole image region. Unfortunately, the available methods for determining local optical flow require a lot of computation. Furthermore, as shown earlier, they use heuristic assumptions and do not generally give a sufficiently accurate estimate of the true motion field over the whole image. This is a big disadvantage since errors as small as a few percent in the motion field can give rise to estimates of 3-D motion that are drastically different from the actual motion.

2.7.1 Instability of the Solution Based on a Local Analysis

We start with Ullman's depth scaling proposition in the case of orthographic projection. For non-planar surfaces he shows that, given an orthographic motion field and a possible rigid body motion interpretation, there is a class of possible solutions all related to the first one through depth scaling (see Ullman [1983] for details). For planar surfaces, the ambiguity is doubled since the orthographic motion field allows two distinct solutions, each resulting in a class of solutions related through depth scaling. For a sufficiently small patch of the surface, large viewing distances, or small viewing angles, perspective and orthographic projections are almost indistinguishable. Since two different objects can have very similar or identical motion fields under orthographic projection, slight errors in the computed motion field may have large effects on the interpreted 3-D motion and shape. This implies that the local interpretation of perspective motion field can easily become unstable. For example, Ullman shows that under perspective projection, errors as small as 3.2% for a one degree visual angle or 6% for twice the visual angle can result in many interpretations that are related through depth scaling. He concludes that it is possible to construct drastically different objects whose motion fields within a region are almost identical.

Similar conclusions can be drawn from the results summarized in Jerian and Jain [1984]. They present examples where instantaneous rotation about a certain axis can be interpreted as instantaneous translation in a direction roughly perpendicular to the plane projection of the axis of rotation. This is because such rotations cause image points to move on hyperbolic paths that look linear in small regions, especially near the center of the image. Our simulation results based on the implementation of several iterative schemes using optical flow as input indicate that, even with perfect data, very different solutions can be obtained where rotation is offset by an appropriate translation. For example, if the true solution is translation with velocity v along positive x -axis (positive y -axis), the algorithm may converge to a solution that is a rotation by approximately v/r in the positive y -axis (negative x -axis), where r is the viewing distance. In general, many solutions are obtained where an incorrect rotation is offset by an appropriate translation in a direction roughly perpendicular to the plane projection of the axis of rotation. The problem becomes even more acute when noise is added to the data to simulate real situations.

Adiv [1985a] presents a theoretical derivation of these results through an error analysis. Again, it is concluded that very different 3-D motions can give rise to very similar perspective motion fields in a local region. Finally, we cite the formulation proposed by Mitiche [1984] based on a rigidity assumption. He shows how the positions and optical velocities of four image points can be used to derive six non-linear equations in terms of selected motion parameters. His simulation results indicate, once again, that although the correct solution can be obtained (in the noise-free case) using good initial guesses of the solution from an iterative method, other numerically acceptable solutions are system-

atically obtained using different initial guesses. He reports that the accuracy of results are rather sensitive to small perturbations of image positions and optical velocities.

These studies point to the instability of the solution of the algorithms that use the optical flow data from a local region to recover 3-D motion (and/or structure). Unless a very accurate estimate of the optical flow is available, we cannot hope to obtain a reliable estimate of 3-D motion.

2.8 The Alternative: Direct Methods

Optical flow can give important information about the spatial arrangement of the objects viewed. Discontinuities in the optical flow can help to segment the image into regions corresponding to different objects. In these applications, only a qualitative rather than a quantitative description of the optical flow is of importance; it is not too critical to obtain a velocity field that is linear, quadratic, or cubic within the boundaries of discontinuity. Therefore, any algorithm that imposes some degree of smoothness on the velocity field can be used both to detect these boundaries and to obtain a solution that captures the qualitative behavior of the true solution within the boundaries of discontinuity. This, however, is not sufficient to obtain an accurate estimate of 3-D motion. We simply cannot afford to use an optical flow which is computed through imposing heuristic constraints to recover 3-D motion since small errors in the computed velocity field have significant effects on the interpreted motion and structure. In addition, computing local optical flow is expensive. This has motivated the investigation of alternative methods for estimating 3-D motion that do not depend on the computation of the optical flow as an intermediate step.

The approach undertaken in this study is based on the belief that much can be learned about 3-D motion simply by studying the variations in image brightness patterns over the whole image. More precisely, we can use the image brightness values (actually brightness gradients) directly to recover 3-D motion (direct passive navigation). In order to do so, we need to derive a constraint equation in terms of the motion parameters, the scene structure, and the brightness derivatives.

2.9 Brightness Change Constraint Equation

If the surface of the scene has sufficient texture or surface markings and the lighting condition remains relatively uniform during successive images, the variations in the image brightness are affected primarily by the motion of the viewer with respect to its environment. It is then reasonable to assume that the brightness of a particular patch on the surface of the scene remains approximately constant from one image to the next image. Therefore, we will write

$$E_r \cdot r_t + E_t = 0.$$

This assumption implicitly states that the true motion field satisfies the optical flow constraint equation. Note that we do not need to assume that optical flow is the same as the motion field. Then we can substitute the expression for the motion field into this equation to arrive at

$$E_t - E_r \cdot \left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\mathbf{r} \times \boldsymbol{\omega} - \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \mathbf{t} \right) \right) \right) = 0.$$

Using the simplification

$$E_r \cdot (\hat{\mathbf{z}} \times (\mathbf{r} \times \mathbf{t})) = (E_r \times \hat{\mathbf{z}}) \cdot (\mathbf{r} \times \mathbf{t}) = ((E_r \times \hat{\mathbf{z}}) \times \mathbf{r}) \cdot \mathbf{t}$$

and similarly

$$E_r \cdot \left(\hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\mathbf{r} \times \boldsymbol{\omega} \right) \right) \right) = \left(((E_r \times \hat{\mathbf{z}}) \times \mathbf{r}) \times \mathbf{r} \right) \cdot \boldsymbol{\omega}$$

we obtain

$$E_t - \left(((E_r \times \hat{\mathbf{z}}) \times \mathbf{r}) \times \mathbf{r} \right) \cdot \boldsymbol{\omega} + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} ((E_r \times \hat{\mathbf{z}}) \times \mathbf{r}) \cdot \mathbf{t} = 0.$$

To make this constraint equation more compact, let us define

$$c = E_t, \quad \mathbf{s} = (E_r \times \hat{\mathbf{z}}) \times \mathbf{r}, \quad \text{and} \quad \mathbf{v} = \mathbf{r} \times \mathbf{s}.$$

Finally, we arrive at the *brightness change constraint equation for the case of rigid body motion*,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} (\mathbf{s} \cdot \mathbf{t}) = 0.$$

This equation shows how the motion of the viewer $\{\boldsymbol{\omega}, \mathbf{t}\}$ with respect to a stationary scene (described by $Z = \mathbf{R} \cdot \hat{\mathbf{z}}$) imposes a constraint on the spatial/temporal changes of the brightness value at every point in the image. Consequently, it can be used to recover the motion parameters and the scene structure as we show in the remainder of this thesis.

2.10 Summary

In this chapter, we have reviewed some details about perspective projection, rigid body motion, the motion field and optical flow. We presented selected examples (from a variety of proposed methods in the literature) based on two common approaches for computing a unique optical flow field. We also described three common approaches, discrete, differential, and least-squares, for recovering 3-D motion from the motion field. We explained why methods based on local measurements of optical flow (or even the motion field) are non-robust and that the least-squares methods require a lot of computation just to obtain a particular optical flow over the whole image. Since the available methods for computing a particular optical flow depend on heuristics and do not generally give a sufficiently accurate estimate of the true motion field over the whole image, we conclude that we need to consider alternative methods that do not rely on the computation of the

optical flow as an intermediate step. We propose using image brightness derivatives to recover 3-D motion simply because the information readily available to us is a sequence of images. From these, we can estimate the temporal and spatial derivatives of the image brightness function easily.

We derived the brightness change constraint equation for the case of rigid body motion. This equation shows how the observer motion and the scene structure impose a constraint on the spatial and temporal variations of the brightness at a particular image point. It was derived under the assumption of brightness constancy; that is, the brightness of an image point (or small patch) remains approximately constant as viewed by the observer who is in motion relative to the scene. This assumption is reasonable when the surface of the scene has sufficient texture (or surface markings) and the lighting condition remains relatively uniform between successive images.

In order to estimate the spatial derivatives accurately, the image brightness should vary smoothly over the image or within the occluding boundaries. In practice, because they are band-limiting, imaging devices cut off or attenuate the high frequency components of the image. As a result, images are analytical functions with bounded derivatives. Therefore, even in regions of sharp texture changes on the object, image grey-level values vary relatively smoothly. To estimate the temporal derivative accurately we often need to assume small relative motion between image frames. This imposes a constraint on the required sampling rate.

In the remainder of this thesis, we will show how we can recover motion by exploiting the brightness change constraint equation over the whole image or in some image region of interest.

Ill-Posed Nature of the Motion Vision Problem

Motion vision is the problem of recovering the 3-D motion and structure of objects in a scene from the time-varying images of the scene. The structure of the scene is usually defined in terms of the relative distance from the viewer to the points on the surface of the scene. We can consider either a moving observer and a stationary scene or a stationary viewer in a scene with moving objects. In general, both the viewer and the objects in the scene can be in motion. In any case, only the relative motion between the viewer and the scene (or objects in the scene) up to the scale-factor ambiguity can be recovered.

In some applications, the primary goal is to recover the structure of the scene (or objects in the scene). This has been commonly referred to as the problem of *structure from motion*. In other applications, we need to recover the motion of the viewer, the problem known as *passive navigation*. In either case, a similar problem has to be solved. Once the scene structure is known, the motion parameters can be determined rather easily. Similarly, if the motion is known, it is not difficult to recover the structure of the scene. In most applications, neither motion nor structure is known a priori and, therefore, the goal is to determine both simultaneously from the information in the time-varying imagery. Here, we address the problem of passive navigation; however, most of the results are applicable if we need to recover structure as well. Our goal is to recover motion directly from time-varying images (direct passive navigation) so that we do not need to compute optical flow as an intermediate step.

In the previous chapter, we derived the brightness change constraint equation for the case of rigid body motion,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z} (\mathbf{s} \cdot \mathbf{t}) = G.$$

As defined in Chapter 2, $c = E_t$ is the time derivative of image brightness function and

$\mathbf{s} = (E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r}$ and $\mathbf{v} = \mathbf{r} \times \mathbf{s}$ are vectors of some moments of brightness gradient $E_{\mathbf{r}}$. This equation shows how the instantaneous motion of the viewer and the scene structure impose a constraint on the spatial/temporal derivatives of the image brightness function at each image point. Given the measurements of the image brightness derivatives, $E_{\mathbf{r}}$ and E_t , we want to recover the instantaneous motion of the viewer, $\boldsymbol{\omega}$ and \mathbf{t} , and the structure of the surface of the scene, namely, the depth function Z .

Each point in the image provides one constraint equation. Suppose we have n data points and, therefore, n constraint equations. We seek to recover $n + 6$ unknowns; that is, n depth values (Z for each image point) and six motion parameters (the three components of each of $\boldsymbol{\omega}$ and \mathbf{t})—or rather $n + 5$ unknowns since the depth values and the translational motion can be recovered only up to a scale factor. It follows immediately that we cannot recover motion and surface structure uniquely from the image brightness constraint equation alone; we need more constraints.

A reasonable approach to resolve this problem is to identify physical constraints that make the problem solvable. We will consider several approaches in order to formulate a well-posed problem. First, we consider some special cases.

3.1 Pure Rotation

In the special case when the motion is pure rotation, the brightness change equation reduces to

$$c + \mathbf{v} \cdot \boldsymbol{\omega} = 0.$$

Since Z drops out of the equations, it becomes clear that the surface cannot be recovered in the absence of translational motion.

The rotational vector, though, can be recovered rather easily. The brightness change equation is linear in $\boldsymbol{\omega}$ and there are three unknowns to solve for. In theory, three image points are all that are needed to recover rotational motion. In practice, a least-squares formulation is most appropriate in order to obtain robust results. We choose $\boldsymbol{\omega}$ that minimizes

$$J = \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})^2 dx dy.$$

The solution is determined from

$$\boldsymbol{\omega} = -\left(\iint_I \mathbf{v}\mathbf{v}^T dx dy\right)^{-1} \left(\iint_I c\mathbf{v} dx dy\right).$$

The product $\mathbf{v}\mathbf{v}^T$ known as a *dyadic product* is a singular 3×3 matrix. The solution to this problem exists, that is, the 3×3 matrix in the above equation is invertible, if the image points do not lie on a straight line.

This solution was first proposed by Aloimonos & Brown [1984]. Horn and Weldon [1986] have investigated this problem in detail. They show that the rotational parameters can be recovered robustly from these equations; however, they point out that a large field

of view is needed to estimate the z -component of ω (the component along the optical axis) accurately.

3.2 Known Motion

If the motion parameters are known, it is possible, in theory, to solve for the surface structure (depth value for each point in the scene) from

$$Z = -\frac{\mathbf{s} \cdot \mathbf{t}}{c + \mathbf{v} \cdot \boldsymbol{\omega}},$$

except when the denominator is zero. In practice, this estimate is noisy because of noise in the images, quantization errors, and errors in estimating brightness derivatives using finite difference methods. Further, at image points where the numerator and/or denominator are/is small, the estimate of depth values will not be very accurate. In general, a smoothing operator (for example, local averaging or convolution with a Gaussian operator) can be applied to the solution to obtain a more reasonable surface representation (this is addressed in more detail in Chapter 8).

3.3 Known Depth

Suppose the depth values are known (for example, from stereo or range data) at n non-collinear data points ($n \geq 6$). Then the brightness change equation for these points,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) = 0,$$

becomes linear in the remaining unknown parameters, namely, the elements of the 3-D vectors $\boldsymbol{\omega}$ and \mathbf{t} . A least-squares estimate of these parameters can be obtained by minimizing the sum-of-squared-error function

$$J = \sum_{i=1}^n (c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}))^2.$$

The extremum of J is given by

$$\frac{\partial J}{\partial \boldsymbol{\omega}} = 0 \quad \text{and} \quad \frac{\partial J}{\partial \mathbf{t}} = 0.$$

Performing the above differentiation we obtain

$$\sum_{i=1}^n \mathbf{v} \left(c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) \right) = 0 \quad \text{and} \quad \sum_{i=1}^n \frac{1}{Z} \mathbf{s} \left(c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) \right) = 0.$$

Any 3-D column vector can be written as a 3×1 matrix and, therefore, $(a \cdot b) = a^T b$. Using this, the above equations can be written

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{d}_1 = \sum_{i=1}^n (c\mathbf{v}), \quad \mathbf{d}_2 = \sum_{i=1}^n \left(\frac{1}{Z} c\mathbf{s}\right),$$

$$\mathbf{M}_1 = \sum_{i=1}^n (\mathbf{v}\mathbf{v}^T), \quad \mathbf{M}_2 = \sum_{i=1}^n \left(\frac{1}{Z} \mathbf{v}\mathbf{s}^T\right), \quad \text{and} \quad \mathbf{M}_4 = \sum_{i=1}^n \left(\frac{1}{Z^2} \mathbf{s}\mathbf{s}^T\right).$$

This is a linear matrix equation that can be solved easily for $\boldsymbol{\omega}$ and \mathbf{t} . The solution can be expressed analytically in several forms; for example, it can be written in terms of the inverse of 3×3 matrices \mathbf{M}_i , $i = 1, \dots, 4$, in the symmetrical form

$$\boldsymbol{\omega} = (\mathbf{M}_2^{-1} \mathbf{M}_1 - \mathbf{M}_4^{-1} \mathbf{M}_2^T)^{-1} (\mathbf{M}_4^{-1} \mathbf{d}_2 - \mathbf{M}_2^{-1} \mathbf{d}_1),$$

$$\mathbf{t} = (\mathbf{M}_1^{-1} \mathbf{M}_2 - \mathbf{M}_2^{-T} \mathbf{M}_4)^{-1} (\mathbf{M}_2^{-T} \mathbf{d}_2 - \mathbf{M}_1^{-1} \mathbf{d}_1),$$

or in the asymmetrical form

$$\mathbf{t} = (\mathbf{M}_4 - \mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{M}_2)^{-1} (\mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{d}_1 - \mathbf{d}_2),$$

$$\boldsymbol{\omega} = -\mathbf{M}_1^{-1} (\mathbf{d}_1 + \mathbf{M}_2 \mathbf{t}).$$

In general, these equations can be solved more easily by elimination. A unique solution can be obtained if we exclude the degenerate cases, for example, when the points lie on a straight line.

3.4 Unknown Depth and Motion

In general, when neither the surface nor the motion is known we have to impose constraints on either the motion or the type of the surface. Since our goal is to recover motion, we impose the constraints on the surface. The remainder of this study is concerned with how such constraints will allow us to formulate a well-posed problem to recover 3-D motion.

We first consider parametric models. More precisely, we approximate the surface of the scene by a polynomial; examples are a frontal plane (that is, the points on the surface of the scene are at the same depth from the viewer), planar surfaces with unknown orientation, or quadratic surfaces (we distinguish between a frontal plane and a plane at any orientation since the problem can be solved more easily in the former case). These can be viewed as zero-order, first-order, and second-order surfaces. In general, the true surface need not have the assumed form. Therefore, this is only an attempt to find a

surface of a certain parametric form (or degree of differentiability) that is most consistent with the data.

We will then show how we can employ the *regularization method* to formulate a well-posed problem. This approach has been successfully adopted in several early vision problems that are under-constrained. Some examples are edge detection (Poggio et al. [1985b]), computing optical flow (Horn & Schunck [1981], Hildreth [1983]), surface reconstruction (Grimson [1981], Terzopoulos [1984a]), and shape from shading (Ikeuchi & Horn [1981], Horn & Brooks [1985]). In these problems, the smoothness is imposed through minimizing a functional that is termed the *regularizing functional*. The choice of the particular functional governs the degree of smoothness.

3.4.1 Parametrization Approach

As before, a viewer-centered coordinate system is chosen. The optical axis is along the Z -axis and the image plane is parallel to the XY -plane. The focal length is assumed to be unity so that the image is formed on the plane $Z = 1$.

The visible surface of the scene may be defined by the function $f(\mathbf{R}) = 0$, where $\mathbf{R} = (X, Y, Z)$ is a point on the surface. Alternatively, the surface may be represented by the function $Z = Z(X, Y)$, where Z is the depth of a point on the visible portion of the surface of the scene.

Let us assume that the surface of the scene is smooth. More precisely, it has continuous derivatives up to some degree n in a local neighborhood. If we take this to be a region around the line of sight, then the surface $Z(X, Y)$ may be expressed by a Taylor series expansion in this region,

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y + \frac{1}{2}\left(\frac{\partial^2 Z}{\partial X^2}\right)_0 X^2 + \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 XY + \frac{1}{2}\left(\frac{\partial^2 Z}{\partial Y^2}\right)_0 Y^2 + \dots + \frac{1}{n!}\left(\frac{\partial^n Z}{\partial X^n}\right)_0 X^n + \dots + \frac{1}{n!}\left(\frac{\partial^n Z}{\partial Y^n}\right)_0 Y^n.$$

Alternatively, we can express Z as a function of the image coordinates. It is more convenient to write the Taylor series expansion in terms of $d = 1/Z$ when image coordinates are used; that is, we will write

$$d = d_0 + \left(\frac{\partial d}{\partial x}\right)_0 x + \left(\frac{\partial d}{\partial y}\right)_0 y + \frac{1}{2}\left(\frac{\partial^2 d}{\partial x^2}\right)_0 x^2 + \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 xy + \frac{1}{2}\left(\frac{\partial^2 d}{\partial y^2}\right)_0 y^2 + \dots + \frac{1}{n!}\left(\frac{\partial^n d}{\partial x^n}\right)_0 x^n + \dots + \frac{1}{n!}\left(\frac{\partial^n d}{\partial y^n}\right)_0 y^n.$$

Let us retain only up to the second-order terms of the expansion. Then we have

$$d = d_0 + \left(\frac{\partial d}{\partial x}\right)_0 x + \left(\frac{\partial d}{\partial y}\right)_0 y + \frac{1}{2}\left(\frac{\partial^2 d}{\partial x^2}\right)_0 x^2 + \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 xy + \frac{1}{2}\left(\frac{\partial^2 d}{\partial y^2}\right)_0 y^2.$$

It can be shown that the coefficients of the Taylor series expansion of Z in terms of the object coordinates are related to those of d in terms of the image coordinates as follows (see Appendix C for proof):

$$d_o = \frac{1}{Z_o}, \quad \left(\frac{\partial d}{\partial x}\right)_o = -\frac{1}{Z_o} \left(\frac{\partial Z}{\partial X}\right)_o, \quad \left(\frac{\partial d}{\partial y}\right)_o = -\frac{1}{Z_o} \left(\frac{\partial Z}{\partial Y}\right)_o,$$

$$\left(\frac{\partial^2 d}{\partial x^2}\right)_o = -\left(\frac{\partial^2 Z}{\partial X^2}\right)_o, \quad \left(\frac{\partial^2 d}{\partial x \partial y}\right)_o = -\left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_o, \quad \text{and} \quad \left(\frac{\partial^2 d}{\partial y^2}\right)_o = -\left(\frac{\partial^2 Z}{\partial Y^2}\right)_o.$$

The above equation can be written in a more compact form using vector notation. As before, let $\mathbf{r} = (x, y, 1)$ denote a point in the image. Then we can write

$$d = \frac{1}{2} \mathbf{r}^T \mathbf{D} \mathbf{r},$$

where

$$\mathbf{D} = \begin{pmatrix} \left(\frac{\partial^2 d}{\partial x^2}\right)_o & \left(\frac{\partial^2 d}{\partial x \partial y}\right)_o & \left(\frac{\partial d}{\partial x}\right)_o \\ \left(\frac{\partial^2 d}{\partial x \partial y}\right)_o & \left(\frac{\partial^2 d}{\partial y^2}\right)_o & \left(\frac{\partial d}{\partial y}\right)_o \\ \left(\frac{\partial d}{\partial x}\right)_o & \left(\frac{\partial d}{\partial y}\right)_o & 2d_o \end{pmatrix}.$$

This is a second-order Taylor series representation of the surface in some neighborhood around the line of sight. Note that there are only six surface parameters (or three if the surface is planar) to be determined. In the general case, the number of unknowns is governed by the order of the Taylor series expansion. In theory, it is then possible to determine these parameters as well as the six motion parameters from measurements in the whole or relevant region of the image plane.

Note that the parametric model need not be of polynomial form. This particular form has been chosen so that the resulting equations are solved more easily. In general, a well-posed problem can be formulated as long as the surface Z (or equivalently $d = 1/Z$) can be represented by a finite number of unknown parameters \mathbf{a} ; that is, $Z = Z(\mathbf{a}, X, Y)$ or $d = d(\mathbf{a}, x, y)$. The accuracy of the estimate of the motion parameters depends on whether or not the particular parametric model is an appropriate representation of the surface of the scene.

3.4.2 Regularization Method

When the local variations in depth are small relative to the absolute distance of the object from the viewer, it is reasonable to assume that the depth values vary smoothly. The discontinuities in depth usually correspond to occluding boundaries of different objects in the scene when it consists of more than one object. Therefore, it is natural to assume that the surface of the scene varies smoothly within occluding boundaries. In most cases,

however, the locations of these boundaries are not known a priori. Thus, we first need to segment the image into regions of smoothly-varying depth values.

Smoothness can be imposed in several ways. Parametric models impose some degree of smoothness on the solution that is directly related to the assumed model. Alternatively, it can be imposed by requiring the solution to be "regular". The measure of "regularity" (smoothness) of the surface is governed by the choice of an appropriate functional known as the *regularizing functional*. Not only is the solution required to be consistent with the data, it also should minimize the departure from some prespecified measure of smoothness. Mathematically, the problem is formulated as that of minimizing the functional

$$e = E_c + \lambda E_s,$$

where E_c is a norm that is a measure of the discrepancy between the observed and predicted data and E_s is the smoothness term. The coefficient λ , the *regularization parameter*, is a weighting factor that controls the degree of regularization of the solution and its difference from the true solution. When the data are noisy, it is better to choose a larger λ in order to require a higher degree of regularity; this is obtained at the expense of allowing a larger discrepancy between the observed and predicted data. When the data are accurate, then λ is chosen to be relatively small in order to guarantee that the solution is consistent with the data.

The choice of E_s reflects the physical constraints of the problem. Suppose we seek the surface $Z(X, Y)$ that minimizes the functional

$$E_s = \iint_I \|\nabla Z\|^2 dX dY = \iint_I \left(\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 \right) dX dY$$

with the appropriate boundary condition on $Z(X, Y)$. This corresponds to finding a surface with the least local variations in depth that satisfies the prescribed boundary conditions. We can instead minimize some measure of the local surface curvature; for example, the integral of the square of the magnitude of the Hessian of the surface,

$$E_s = \iint_I \|\nabla \nabla Z\|^2 dX dY = \iint_I \left(\left(\frac{\partial^2 Z}{\partial X^2} \right)^2 + 2 \left(\frac{\partial^2 Z}{\partial X \partial Y} \right)^2 + \left(\frac{\partial^2 Z}{\partial Y^2} \right)^2 \right) dX dY.$$

Again, the appropriate boundary conditions have to be imposed.

3.4.3 Depth Positivity as a Constraint

Depth values can only be positive since points on the surface have to be in front of the camera to be imaged. In many cases, this constraint is all that is needed to recover 3-D motion uniquely. Unfortunately, the problem is relatively difficult to solve when the motion consists of both the rotation and translation of the observer relative to the scene. We show how 3-D motion may be recovered either when it is purely translational or

when the rotational component is known. This method can be extended to the general case when neither translation nor rotation is known; however, the procedure may not be robust because of the inherent ambiguities in the solution when data are noisy.

3.5 Summary

In general, the only information available to us is a sequence of images; hence, the only constraint is provided by the brightness change constraint equation. In some cases, this is all that is needed to recover the motion parameters. For example, when the motion consists of the rotation of the viewer, the three motion parameters can be determined easily from three linear equations. When the depth values are known, at least at six image points that do not lie on a straight line in the image, the motion parameters can be determined by solving six linear equations. In general, we cannot recover 3-D motion and structure using the brightness change constraint equation alone; we need more constraints to formulate a well-posed problem. These constraints can be imposed in terms of some assumptions about the structure of the scene. In the remaining chapters, we consider several cases. We will show how to recover motion when the surface of the scene is planar or quadratic (or can be approximated as such). We will also show that translational motion can be recovered from the image brightness gradient vectors using the positivity of depth as a constraint. Finally, we will investigate the regularization approach and formulate the problem of recovering motion from time-varying images as a problem in calculus of variations.

Planar Surfaces

A particularly impoverished scene is one consisting of a single planar surface. The equation for such a surface is

$$\mathbf{R} \cdot \mathbf{n} = 1,$$

where $\mathbf{n}/\|\mathbf{n}\|$ is a unit vector orthogonal to the planar surface and $1/\|\mathbf{n}\|$ is the perpendicular distance of the plane from the origin. The above equation can be viewed as the Taylor series expansion of the surface up to first order terms around the fixation point (the equation of the tangent plane at the fixation point). To show this, let us write the Taylor series expansion of the surface in a neighborhood around the line of sight retaining the first order terms,

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y.$$

If $Z_0 \neq 0$, this can be written as

$$-\frac{1}{Z_0} \left(\left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y - Z \right) = 1.$$

It is easy to see that we must have

$$\mathbf{n} = \frac{1}{Z_0} \left[-\left(\frac{\partial Z}{\partial X}\right)_0, -\left(\frac{\partial Z}{\partial Y}\right)_0, 1 \right]^T.$$

In general, we can write the Taylor expansion of the surface around any point $\mathbf{R}_0 = (X_0, Y_0, Z_0)$ on the surface; that is,

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 (X - X_0) + \left(\frac{\partial Z}{\partial Y}\right)_0 (Y - Y_0).$$

By collecting terms we obtain

$$Z = \left(Z_0 - \left(\frac{\partial Z}{\partial X} \right)_0 X_0 - \left(\frac{\partial Z}{\partial Y} \right)_0 Y_0 \right) + \left(\frac{\partial Z}{\partial X} \right)_0 X + \left(\frac{\partial Z}{\partial Y} \right)_0 Y.$$

Again, this is in the form

$$\mathbf{R} \cdot \mathbf{n} = 1,$$

where

$$\mathbf{n} = \frac{1}{\left(Z_0 - \left(\frac{\partial Z}{\partial X} \right)_0 X_0 - \left(\frac{\partial Z}{\partial Y} \right)_0 Y_0 \right)} \left[-\left(\frac{\partial Z}{\partial X} \right)_0, -\left(\frac{\partial Z}{\partial Y} \right)_0, 1 \right]^T.$$

It is more convenient to express the surface in terms of the image coordinates. Since $\mathbf{R} = \frac{1}{Z}\mathbf{r}$, we can write

$$d = \frac{1}{Z} = \mathbf{r} \cdot \mathbf{n}.$$

This is the equation of a planar surface in terms of the image coordinates. Note that we now have to estimate only three surface parameters, the components of the normal vector \mathbf{n} .

Substituting the equation of the plane into the brightness change constraint equation,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) = 0,$$

we arrive at

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0,$$

where $c = E_t$ is the time derivative of image brightness function and $\mathbf{s} = (\mathbf{E}_r \times \hat{\mathbf{z}}) \times \mathbf{r}$ and $\mathbf{v} = \mathbf{r} \times \mathbf{s}$ are vectors of some moments of brightness gradient \mathbf{E}_r . This is the *brightness change constraint equation for a planar surface*. Note again the inherent ambiguity in the constraint equation. It is satisfied equally well by two planes with the same orientation but at different distances provided that the translational velocities are in the same proportion.

Given the image brightness $E(x, y, t)$ and its spatial and time derivatives, \mathbf{E}_r and E_t , over some region I in the image plane, we want to recover the translational and rotational motions, \mathbf{t} and $\boldsymbol{\omega}$, as well as the plane \mathbf{n} . Negahdaripour & Horn [1986b] present two methods, iterative and closed-form, for estimating the motion parameters when the scene is a planar surface. Before we present these results, we consider a special case.

4.1 A Frontal Plane

The surface of the scene may be approximated by a frontal plane if the absolute distance from the viewer to the surface of the scene is much larger than the depth variations on the surface. For a frontal plane, the normal is along the optical axis and the points on the plane have the same depth value. Let us denote this distance by Z_0 (that is, we have

$n_1 = n_2 = 0$ and $n_3 = \frac{1}{Z_0}$). Since we cannot recover absolute depth due to scale-factor ambiguity, we can assume that $Z_0 = 1$ without loss of generality. Then, for every image point, the brightness change constraint equation reduces to

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{s} \cdot \mathbf{t}) = 0.$$

This is a linear constraint equation in terms of the motion parameters and can be used to determine these parameters rather easily. We can formulate a least-squares problem to minimize the error in the constraint equation over the relevant region of the image plane,

$$\iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega} + \mathbf{s} \cdot \mathbf{t})^2 dx dy.$$

Using necessary conditions for optimality, this reduces to solving the linear matrix equation

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{d}_1 = \iint_I c\mathbf{v} dx dy, \quad \mathbf{d}_2 = \iint_I c\mathbf{s} dx dy,$$

$$\mathbf{M}_1 = \iint_I \mathbf{v}\mathbf{v}^T dx dy, \quad \mathbf{M}_2 = \iint_I \mathbf{v}\mathbf{s}^T dx dy, \quad \text{and} \quad \mathbf{M}_4 = \iint_I \mathbf{s}\mathbf{s}^T dx dy.$$

The solution can be obtained from analytical expressions. For example, it can be written in the symmetric form,

$$\begin{aligned} \boldsymbol{\omega} &= (\mathbf{M}_2^{-1}\mathbf{M}_1 - \mathbf{M}_4^{-1}\mathbf{M}_2^T)^{-1}(\mathbf{M}_4^{-1}\mathbf{d}_2 - \mathbf{M}_2^{-1}\mathbf{d}_1), \\ \mathbf{t} &= (\mathbf{M}_1^{-1}\mathbf{M}_2 - \mathbf{M}_2^{-T}\mathbf{M}_4)^{-1}(\mathbf{M}_2^{-T}\mathbf{d}_2 - \mathbf{M}_1^{-1}\mathbf{d}_1), \end{aligned}$$

where $(^{-T})$ denotes the inverse of the transpose of a matrix, or it can be written in terms of the inverse of \mathbf{M}_1 in the asymmetrical form,

$$\begin{aligned} \mathbf{t} &= (\mathbf{M}_4 - \mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{M}_2)^{-1}(\mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{d}_1 - \mathbf{d}_2), \\ \boldsymbol{\omega} &= -\mathbf{M}_1^{-1}(\mathbf{d}_1 + \mathbf{M}_2\mathbf{t}). \end{aligned}$$

It can also be obtained simply by elimination.

4.2 A Plane at any Arbitrary Orientation

Given the image brightness $E(x, y, t)$ and its spatial and time derivatives, E_r and E_t , over some region I in the image plane, we want to recover the translational and rotational motions, \mathbf{t} and $\boldsymbol{\omega}$, as well as the plane \mathbf{n} using the brightness change constraint equation for a planar surface,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

Ideally, we could do this with image data at just a small number of points. At each point we have one constraint and there are nine unknowns to recover up to the scale-factor ambiguity in the translational velocity and the surface normal.

Image brightness values are corrupted by sensor noise and quantization. These inaccuracies are further accentuated by methods used for estimating brightness derivatives. Thus, it is not advisable to base a method on measurements at just a few points. We instead propose to minimize the error in the brightness constraint equation over the whole region I in the image plane; that is, we wish to minimize

$$J = \iint_I [c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})]^2 dx dy$$

by suitable choice of the translational and rotational motion vectors \mathbf{t} and $\boldsymbol{\omega}$ as well as the normal to the plane \mathbf{n} .

For an extremum of J we must have

$$\frac{\partial J}{\partial \boldsymbol{\omega}} = \mathbf{0}, \quad \frac{\partial J}{\partial \mathbf{t}} = \mathbf{0}, \quad \text{and} \quad \frac{\partial J}{\partial \mathbf{n}} = \mathbf{0}.$$

Thus we obtain

$$\begin{aligned} \iint_I [\mathbf{v}(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r})(\mathbf{s} \cdot \mathbf{t}))] dx dy &= 0, \\ \iint_I [\mathbf{s}(\mathbf{n} \cdot \mathbf{r})(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r})(\mathbf{s} \cdot \mathbf{t}))] dx dy &= 0, \\ \iint_I [\mathbf{r}(\mathbf{s} \cdot \mathbf{t})(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r})(\mathbf{s} \cdot \mathbf{t}))] dx dy &= 0. \end{aligned}$$

These equations comprise nine non-linear (scalar) algebraic equations in terms of the observer motion, \mathbf{t} and $\boldsymbol{\omega}$, and the surface normal \mathbf{n} . They will be referred to as the *planar motion field equations* and have to be solved in order to recover the unknown parameters. Note that we can recover \mathbf{n} and \mathbf{t} up to a scale factor. Therefore, we can set the length of \mathbf{t} or one of the components of \mathbf{t} or \mathbf{n} equal to one so that we solve only eight non-linear equations.

4.2.1 Iterative Methods

Since the planar motion field equations are non-linear, we can improvise an efficient iterative scheme for solving them later). To do so, some observations are in order: The first equation is linear in $\boldsymbol{\omega}$, \mathbf{t} , and \mathbf{n} . The second equation is linear in $\boldsymbol{\omega}$ and \mathbf{t} but quadratic in \mathbf{n} . Finally, the last equation is linear in $\boldsymbol{\omega}$ and \mathbf{n} but quadratic in \mathbf{t} . We will exploit the linearity of these equations to formulate two iterative schemes.

4.2.1.1 First Scheme

The planar motion field equations can be written as

$$\begin{aligned} \left(\iint_I (\mathbf{v}\mathbf{v}^T) dx dy \right) \boldsymbol{\omega} + \left(\iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{v}\mathbf{s}^T) dx dy \right) \mathbf{t} &= - \iint_I c\mathbf{v} dx dy, \\ \left(\iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{s}\mathbf{v}^T) dx dy \right) \boldsymbol{\omega} + \left(\iint_I (\mathbf{r} \cdot \mathbf{n})^2(\mathbf{s}\mathbf{s}^T) dx dy \right) \mathbf{t} &= - \iint_I c(\mathbf{r} \cdot \mathbf{n})\mathbf{s} dx dy, \\ \left(\iint_I (\mathbf{s} \cdot \mathbf{t})^2(\mathbf{r}\mathbf{r}^T) dx dy \right) \mathbf{n} &= - \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})(\mathbf{s} \cdot \mathbf{t})\mathbf{r} dx dy. \end{aligned}$$

In writing the above equation we have used the fact that 3-D column vectors can be written as 3×1 matrices. The first two equations can be grouped in the form

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{d}_1 = \iint_I (c\mathbf{v}) dx dy, \quad \mathbf{d}_2 = \iint_I c(\mathbf{r} \cdot \mathbf{n})\mathbf{s} dx dy,$$

$$\mathbf{M}_1 = \iint_I (\mathbf{v}\mathbf{v}^T) dx dy, \quad \mathbf{M}_2 = \iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{v}\mathbf{s}^T) dx dy, \quad \text{and} \quad \mathbf{M}_4 = \iint_I (\mathbf{r} \cdot \mathbf{n})^2(\mathbf{s}\mathbf{s}^T) dx dy.$$

This can be solved for \mathbf{t} and $\boldsymbol{\omega}$ given the surface normal \mathbf{n} . The last equation can be written

$$\mathbf{N}_4 \mathbf{n} = -\mathbf{g},$$

where

$$\mathbf{N}_4 = \iint_I (\mathbf{s} \cdot \mathbf{t})^2(\mathbf{r}\mathbf{r}^T) dx dy \quad \text{and} \quad \mathbf{g} = \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})(\mathbf{s} \cdot \mathbf{t})\mathbf{r} dx dy.$$

Given the pair of vectors \mathbf{t} and $\boldsymbol{\omega}$, this equation can be solved for the surface normal \mathbf{n} . The motion vectors can be obtained from

$$\begin{aligned} \boldsymbol{\omega} &= (\mathbf{M}_2^{-1}\mathbf{M}_1 - \mathbf{M}_4^{-1}\mathbf{M}_2^T)^{-1}(\mathbf{M}_4^{-1}\mathbf{d}_2 - \mathbf{M}_2^{-1}\mathbf{d}_1), \\ \mathbf{t} &= (\mathbf{M}_1^{-1}\mathbf{M}_2 - \mathbf{M}_2^{-T}\mathbf{M}_4)^{-1}(\mathbf{M}_2^{-T}\mathbf{d}_2 - \mathbf{M}_1^{-1}\mathbf{d}_1). \end{aligned}$$

Since \mathbf{M}_1 is a constant matrix, we can exploit this to write the solution in the form

$$\begin{aligned} \mathbf{t} &= (\mathbf{M}_4 - \mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{M}_2)^{-1}(\mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{d}_1 - \mathbf{d}_2), \\ \boldsymbol{\omega} &= -\mathbf{M}_1^{-1}(\mathbf{d}_1 + \mathbf{M}_2\mathbf{t}). \end{aligned}$$

The surface normal is simply given by

$$\mathbf{n} = -\mathbf{N}_4^{-1}\mathbf{g}.$$

The expression for the solution involves the inversion of 3×3 matrices. In order to rule out the possibility of the degeneracy of the solution, it is usually enough to require that the points do not lie on a straight line in the image. In most cases, the problem is well-formulated since we use the information over the whole image region. Exceptions arise, for example, when the image consists of regions of uniform grey-levels.

All arrays in the above expressions are either 3×3 matrices or vectors of length 3 and, therefore, the solutions for ω , \mathbf{t} , and \mathbf{n} can be computed easily. Actually, most of the indicated matrix inversions do not have to be carried out explicitly since it is computationally cheaper to solve these linear equations by elimination. To summarize:

- (1) We start with an initial guess for \mathbf{n} .
- (2) Using the above equations we solve for \mathbf{t} and ω in terms of the current value of \mathbf{n} and then for \mathbf{n} in terms of the current values of \mathbf{t} and ω .
- (3) After this, we evaluate the improvement in the solution to decide whether to go to the next iteration or stop if the solution has not improved.

Since we can recover depth and translational motion only up to the scale-factor ambiguity we can, after each iteration, either fix the length of the translational vector (e.g., we can set it equal to one) or set one of the parameters of \mathbf{n} or \mathbf{t} equal to a constant (for example, we can set $n_3 = 1$).

4.2.1.2 Second Scheme

The first pair of the planar motion field equations depend linearly on \mathbf{t} and ω ,

$$\begin{aligned} \left(\iint_I (\mathbf{v}\mathbf{v}^T) dx dy \right) \omega + \left(\iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{v}\mathbf{s}^T) dx dy \right) \mathbf{t} &= - \iint_I c\mathbf{v} dx dy, \\ \left(\iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{s}\mathbf{v}^T) dx dy \right) \omega + \left(\iint_I (\mathbf{r} \cdot \mathbf{n})^2(\mathbf{s}\mathbf{v}^T) dx dy \right) \mathbf{t} &= - \iint_I c(\mathbf{r} \cdot \mathbf{n})\mathbf{v} dx dy. \end{aligned}$$

These can be solved for \mathbf{t} and ω in terms of \mathbf{n} . Furthermore, the first and last equations depend linearly on \mathbf{n} and ω ,

$$\begin{aligned} \left(\iint_I (\mathbf{v}\mathbf{v}^T) dx dy \right) \omega + \left(\iint_I (\mathbf{r} \cdot \mathbf{n})(\mathbf{v}\mathbf{s}^T) dx dy \right) \mathbf{t} &= - \iint_I c\mathbf{v} dx dy, \\ \left(\iint_I (\mathbf{s} \cdot \mathbf{t})(\mathbf{r}\mathbf{v}^T) dx dy \right) \omega + \left(\iint_I (\mathbf{s} \cdot \mathbf{t})^2(\mathbf{r}\mathbf{r}^T) dx dy \right) \mathbf{n} &= - \iint_I c(\mathbf{s} \cdot \mathbf{t})\mathbf{r} dx dy. \end{aligned}$$

Given \mathbf{t} , these may be solved for \mathbf{n} and ω . For simplicity, let \mathbf{M}_1 , \mathbf{M}_2 , \mathbf{M}_4 , \mathbf{N}_4 , \mathbf{d}_1 , and \mathbf{d}_2 be as defined earlier and let

$$\mathbf{e}_1 = \mathbf{d}_1, \quad \mathbf{e}_2 = \iint_I c(\mathbf{s} \cdot \mathbf{t})\mathbf{r} dx dy,$$

$$\mathbf{N}_1 = \mathbf{M}_1, \quad \text{and} \quad \mathbf{N}_2 = \iint_I (\mathbf{s} \cdot \mathbf{t})(\mathbf{v}\mathbf{r}^T) dx dy.$$

Then we have

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M} \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{N}_1 & \mathbf{N}_2 \\ \mathbf{N}_2^T & \mathbf{N}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{n} \end{pmatrix} = - \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}.$$

The solution of the above equations is given by

$$\mathbf{t} = (\mathbf{M}_4 - \mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{M}_2)^{-1} (\mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{d}_1 - \mathbf{d}_2),$$

$$\boldsymbol{\omega} = -\mathbf{M}_1^{-1} (\mathbf{d}_1 + \mathbf{M}_2 \mathbf{t}),$$

and

$$\mathbf{n} = (\mathbf{N}_4 - \mathbf{N}_2^T \mathbf{N}_1^{-1} \mathbf{N}_2)^{-1} (\mathbf{N}_2^T \mathbf{N}_1^{-1} \mathbf{e}_1 - \mathbf{e}_2),$$

$$\boldsymbol{\omega} = -\mathbf{N}_1^{-1} (\mathbf{e}_1 + \mathbf{N}_2 \mathbf{n}).$$

As in the first scheme, the matrix inversions do not have to be carried out explicitly since the solution may be obtained by elimination. To summarize:

- (1) We start with an initial guess for \mathbf{n} .
- (2) We solve for \mathbf{t} and $\boldsymbol{\omega}$ in terms of the current value of \mathbf{n} and update \mathbf{t} . We then solve for \mathbf{n} and $\boldsymbol{\omega}$ in terms of the current value of \mathbf{t} and update \mathbf{n} .
- (3) Finally, we evaluate the improvement in the solution to either continue with the next iteration or stop if the solution has not improved.

Again, we can either normalize the translational vector or set one of the parameters of \mathbf{n} or \mathbf{t} equal to one after each iteration.

4.2.1.3 Division of Labor

These methods would not be very attractive if we had to perform integrations over the whole image region I during each iteration in order to compute the matrices and vectors appearing in the equations. Fortunately, this is not necessary. One can see this by writing the equations for the components of the matrices and vectors using the summation convention of tensor calculus (that is, there is an implicit summation over any index that appears twice in an expression):

$$\{\mathbf{M}_1\}_{i,j} = \iint_I v_i v_j dx dy, \quad \{\mathbf{M}_2\}_{i,j} = \left[\iint_I v_i s_j r_k dx dy \right] n_k,$$

$$\{\mathbf{M}_4\}_{i,j} = \left[\iint_I s_i s_j r_k r_l dx dy \right] n_k n_l,$$

$$\begin{aligned} \{\mathbf{d}_1\}_i &= \iint_I cv_i dx dy, & \{\mathbf{d}_2\}_i &= \left[\iint_I cs_i r_j dx dy \right] n_j, \\ \{\mathbf{N}_1\}_{i,j} &= \iint_I v_i v_j dx dy, & \{\mathbf{N}_2\}_{i,j} &= \left[\iint_I v_i s_k r_j dx dy \right] t_k, \\ & & \{\mathbf{N}_4\}_{i,j} &= \left[\iint_I s_k s_l r_i r_j dx dy \right] t_k t_l, \\ \{\mathbf{e}_1\}_i &= \iint_I cv_i dx dy, & \{\mathbf{e}_2\}_i &= \left[\iint_I cs_j r_i dx dy \right] t_j, \end{aligned}$$

and

$$\{\mathbf{g}\}_i = \left[\iint_I cr_i s_j dx dy \right] t_j + \left[\iint_I s_k r_i v_j dx dy \right] t_k w_j.$$

$\mathbf{M}_1 = \mathbf{N}_1$ and $\mathbf{d}_1 = \mathbf{e}_1$ do not depend on ω , t , or \mathbf{n} and so need be computed only once. Also, (cv_i) , $(v_i v_j)$, $(cs_i r_j)$, $(r_k v_i s_j)$, and $(r_k r_l s_i s_j)$ depend only on r , E_r , and E_t , and can be integrated over the image once. This adds up to a set of $3 + 9 + 9 + 27 + 81 = 129$ numbers; however, only 81 numbers have to be stored because of symmetry in $(v_i v_j)$ and $(r_k r_l s_i s_j)$. These accumulated totals represent all the image information needed to solve the planar motion field equations iteratively.

In the first scheme, we only perform 279 multiplications per iteration; the updating of \mathbf{M}_2 , \mathbf{d}_2 , \mathbf{M}_4 , \mathbf{N}_4 , and \mathbf{g} (note that \mathbf{M}_4 and \mathbf{N}_4 are symmetric) requires $27 + 9 + 42 + 42 + 42 = 162$ multiplications and the updating of ω , t , and \mathbf{n} , by comparison, involves 117 multiplications.

In the second scheme, 696 multiplications are carried out at each iteration. We compute the matrices \mathbf{M}_2 , \mathbf{M}_4 and the vector \mathbf{d}_2 required for the first half of the iteration in $27 + 42 + 9 = 78$ multiplications. The same number of multiplications is needed to compute the matrices \mathbf{N}_2 , \mathbf{N}_4 , and the vector \mathbf{e}_2 required in the second half. Further, solving for ω and t takes about 270 multiplications as does solving for ω and \mathbf{n} in the second half of each iterative step.

In an example, we will show that the second scheme has a much better convergence rate, but it requires more computation per iteration.

4.2.1.4 Choosing an Initial Condition

As we show through selected examples from the simulations we have performed, a good estimate of the solution is not necessary for convergence to the solution; the two iterative methods seem to always converge to one of the two possible solutions (the existence of two solutions is shown next) even though we have not presented a proof of convergence mathematically. The rate of convergence of the two iterative schemes seems to depend more on the particular situation that is simulated (namely, the motion of the camera and the orientation of the plane) than on the choice of the initial condition; however,

the convergence rate is slightly slower when the initial condition is near the boundary of the convergence regions of the two possible solutions (this is the region where some perturbation of the initial condition causes the algorithm to converge to one or the other solution). It may be appropriate, however, to assume that the surface is a frontal plane. This, in many instances, gives a very good estimate of the solution. Actually, the solution in the case of a frontal plane is the same as the solution of the first iterative method after one iteration when the initial guess of the surface normal is that it lies along the optical axis.

4.2.2 Uniqueness

It is important to establish whether more than one solution can be obtained. In general, this may occur since an image of uniform brightness could correspond to an arbitrary uniform surface moving in an arbitrary way. Therefore, the brightness gradients or lack of brightness gradients can conspire to make the problem highly ambiguous. What we are interested in here is whether two different planar surfaces can give rise to the same motion field given two different translational and rotational motions of the imaging system. Thus, the question becomes: Given that the brightness change equation is satisfied for the motion \mathbf{t} and $\boldsymbol{\omega}$ and the planar surface \mathbf{n} , is there another motion \mathbf{t}' and $\boldsymbol{\omega}'$ and another planar surface \mathbf{n}' that satisfies the same equation at all points in the region I and for all possible ways of marking the surface? Note that we have to consider a whole image region since the problem is underconstrained if we only have information along a line or at a point in the image. We also have to include the condition that the constraint should be satisfied for all possible surface markings to avoid the kind of ambiguity discussed above where brightness gradients coincidentally line up with the motion field to create ambiguity.

Suppose that two motions and two planar surfaces satisfy the brightness change equation. Then we have

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0 \quad \text{and} \quad c + \mathbf{v} \cdot \boldsymbol{\omega}' + (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0.$$

Subtracting these equations and substituting $\mathbf{v} = \mathbf{r} \times \mathbf{s}$ we obtain

$$(\mathbf{s} \times \mathbf{r}) \cdot (\boldsymbol{\omega} - \boldsymbol{\omega}') + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) - (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0$$

or

$$-\mathbf{r} \cdot ((\boldsymbol{\omega} - \boldsymbol{\omega}') \times \mathbf{s}) + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) - (\mathbf{r} \cdot \mathbf{n}')(\mathbf{s} \cdot \mathbf{t}') = 0.$$

There is a well-known isomorphism between vectors and skew-symmetric matrices that we can use to our advantage here. The skew-symmetric matrix $\boldsymbol{\Omega}$ corresponding to the vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$ is

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix}.$$

The $(i, j)^{\text{th}}$ element of Ω , Ω_{ij} , equals $\omega_k \epsilon_{ijk}$ where ϵ_{ijk} is the permutation symbol. It equals +1 when the ordered set i, j, k is obtained by an even permutation of the set 1, 2, and 3, and it equals -1 when the ordered set is obtained by an odd permutation. It is zero if two or more of the indices are equal. It is easy to show that

$$\omega \times \mathbf{s} = \Omega \mathbf{s}.$$

Using this notation we can write

$$-\mathbf{r}^T(\Omega - \Omega')\mathbf{s} + \mathbf{r}^T(\mathbf{n}\mathbf{t}^T)\mathbf{s} - \mathbf{r}^T(\mathbf{n}'\mathbf{t}'^T)\mathbf{s} = 0$$

or just

$$\mathbf{r}^T \left[-(\Omega - \Omega') + \mathbf{n}\mathbf{t}^T - \mathbf{n}'\mathbf{t}'^T \right] \mathbf{s} = 0.$$

This must hold for all points \mathbf{r} in the image region I and all possible brightness gradients. Therefore, we must have

$$-(\Omega - \Omega') + \mathbf{n}\mathbf{t}^T - \mathbf{n}'\mathbf{t}'^T = \mathbf{0},$$

where the zero on the right hand side represents a 3×3 matrix of zeros. Since Ω is skew-symmetric, we have $\Omega^T = -\Omega$. Taking the transpose of the equation we obtain

$$+(\Omega - \Omega') + \mathbf{t}\mathbf{n}^T - \mathbf{t}'\mathbf{n}'^T = \mathbf{0}.$$

Adding the two equations allows us to eliminate $(\Omega - \Omega')$. We then end up with

$$(\mathbf{n}\mathbf{t}^T + \mathbf{t}\mathbf{n}^T) = (\mathbf{n}'\mathbf{t}'^T + \mathbf{t}'\mathbf{n}'^T).$$

The trace (sum of the diagonal elements) of $\mathbf{n}\mathbf{t}^T$ is just $(\mathbf{n} \cdot \mathbf{t})$. It follows immediately that $(\mathbf{n} \cdot \mathbf{t}) = (\mathbf{n}' \cdot \mathbf{t}')$; however, there are more constraints in the above matrix equation involving the dyadic products of \mathbf{n} and \mathbf{t} as well as of \mathbf{n}' and \mathbf{t}' . Consider the following three possibilities:

- (1) If $\|\mathbf{n}'\| = 0$ or $\|\mathbf{t}'\| = 0$, their dyadic product is a 3×3 matrix of zeros. In this case, the above equation is satisfied if and only if $\|\mathbf{n}\| = 0$ or $\|\mathbf{t}\| = 0$.
- (2) If $\mathbf{n}' \parallel \mathbf{n}$, $\mathbf{t}' \parallel \mathbf{t}$, and $\|\mathbf{n}'\| \|\mathbf{t}'\| = \|\mathbf{n}\| \|\mathbf{t}\|$, then the two sums of dyadic products are equal and the above equation is satisfied.
- (3) If $\mathbf{n}' \parallel \mathbf{t}$, $\mathbf{t}' \parallel \mathbf{n}$, and $\|\mathbf{n}'\| \|\mathbf{t}'\| = \|\mathbf{n}\| \|\mathbf{t}\|$, then the two sums of dyadic products are also equal and the above equation is satisfied.

It turns out that there are no other ways we can satisfy the equation. This can be shown using elementary properties of dyadic products (see Appendix F) or by the inspection of the six components of the above equation (because of symmetry there are only six independent components).

The first case above corresponds to purely rotational motion because either the translational motion is zero or the planar surface is infinitely far away and the translation does not generate a perceptible component of the motion field. The solution is unique in this case because we find $(\Omega - \Omega') = 0$ when we substitute back into the matrix equation. This is nothing new since it has been known for some time that the solution is unique in the cases of purely rotational and purely translational motion (Bruss & Horn [1983]).

In the second case, we find that $\mathbf{n}\mathbf{t}^T = \mathbf{n}'\mathbf{t}'^T$ since the vectors are parallel and the product of their size is constrained by the condition $\mathbf{n} \cdot \mathbf{t} = \mathbf{n}' \cdot \mathbf{t}'$ derived earlier. Thus, once again, we have $(\Omega - \Omega') = 0$. This corresponds to the scale-factor ambiguity in motion vision since we can scale the lengths of the vectors \mathbf{n} and \mathbf{t} as long as the product of their lengths remains unchanged.

The third case is the most interesting. Here we have $\mathbf{t}\mathbf{n}^T = \mathbf{n}'\mathbf{t}'^T$ so that

$$-(\Omega - \Omega') + (\mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T) = 0.$$

Thus, we can write

$$-(\Omega - \Omega')\mathbf{x} + (\mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T)\mathbf{x} = 0$$

for an arbitrary vector \mathbf{x} ; that is,

$$\mathbf{x} \times (\boldsymbol{\omega} - \boldsymbol{\omega}') + \mathbf{x} \times (\mathbf{n} \times \mathbf{t}) = 0$$

for an arbitrary vector \mathbf{x} . We finally arrive at

$$\boldsymbol{\omega} - \boldsymbol{\omega}' + \mathbf{n} \times \mathbf{t} = 0$$

or

$$\boldsymbol{\omega}' = \boldsymbol{\omega} + \mathbf{n} \times \mathbf{t}.$$

To summarize, if we ignore scaling of the normal and the translational velocity we obtain a dual solution given by

$$\mathbf{n}' = \mathbf{t}, \quad \mathbf{t}' = \mathbf{n}, \quad \text{and} \quad \boldsymbol{\omega}' = \boldsymbol{\omega} + \mathbf{n} \times \mathbf{t}.$$

Hay [1966] was the first to show the existence of a dual solution although the result has apparently been independently rediscovered several times since then (Tsai et al. [1982], Waxman & Ullman [1983], Longuet-Higgins [1984], Maybank [1984], Negahdaripour & Horn [1985a]).

This dual solution is not different from the original one in the special case that $\mathbf{n} \parallel \mathbf{t}$; in this case, the solution is unique. Further, if $\mathbf{t} \cdot \hat{\mathbf{z}} = 0$, then $\mathbf{n}' \cdot \hat{\mathbf{z}} = 0$. This corresponds to viewing the planar surface "from the side" and can be considered a degenerate case since the image of the dual plane reduces to a line.

4.2.3 Closed-Form Solution

The motion and plane parameters may be determined analytically in a two-step procedure. We first compute eight intermediate parameters from the brightness derivatives over the whole region of the image plane. We then estimate the motion and plane parameters from the eigenvalue-eigenvector decomposition of a 3×3 matrix whose elements are the intermediate parameters.

For planar surfaces, we have presented two methods, iterative and closed-form, to demonstrate a point. This is the issue of the non-robustness and non-uniqueness of the solution of some analytical methods given in the computer vision literature when the corresponding algorithms are not implemented correctly. For example, in the case of quadratic surfaces, the closed-form solution is generally non-robust, non-unique, and inferior to a solution obtained from an iterative method in the presence of noise in the data (these results are presented in Chapter 5 and Chapter 6). The reader is referred to Appendix A where we address this issue in more detail and explain why the analytical method we will present next gives the best estimate, in the least-squares sense.

4.2.3.1 Essential Parameters of Planar Surfaces

The brightness change equation can be written

$$c + (\mathbf{r} \times \mathbf{s}) \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

Using the identity $(\mathbf{r} \times \mathbf{s}) \cdot \boldsymbol{\omega} = \mathbf{r} \cdot (\mathbf{s} \times \boldsymbol{\omega})$ we obtain

$$c - \mathbf{r} \cdot (\boldsymbol{\omega} \times \mathbf{s}) + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

The isomorphism between vectors and skew-symmetric matrices used in the proof of the existence of a dual solution can be exploited here. If we define

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix},$$

then $\boldsymbol{\Omega}\mathbf{s} = (\boldsymbol{\omega} \times \mathbf{s})$ and we conclude that

$$c - \mathbf{r}^T \boldsymbol{\Omega} \mathbf{s} + (\mathbf{r}^T \mathbf{n})(\mathbf{t}^T \mathbf{s}) = 0 \quad \text{or} \quad c + \mathbf{r}^T (-\boldsymbol{\Omega} + \mathbf{n} \mathbf{t}^T) \mathbf{s} = 0.$$

Let

$$\mathbf{P} = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix} = -\boldsymbol{\Omega} + \mathbf{n} \mathbf{t}^T.$$

We can finally write

$$c + \mathbf{r}^T \mathbf{P} \mathbf{s} = 0.$$

We will refer to $\{p_i\}$ as the *essential parameters of planar surfaces* (a term first used by Tsai & Huang [1984] in a related context) since these parameters contain all the information about the planar surface and motion parameters. The above constraint equation is linear in the elements of \mathbf{P} . Several such equations for different image points can be used to solve for these parameters. We will show how the special structure of \mathbf{P} can be exploited to recover the motion and plane parameters.

Note that the essential parameters are not independent because \mathbf{P} is not an arbitrary 3×3 matrix. It has a special structure since it is the sum of a skew-symmetric matrix and a dyadic product. It takes three parameters to specify ω (and, hence, Ω), three to specify \mathbf{n} , and another three for \mathbf{t} . The matrix \mathbf{P} , however, is unchanged if we replace \mathbf{n} by $k\mathbf{n}$ and \mathbf{t} by $(1/k)\mathbf{t}$ for any nonzero k . Therefore, there are actually only eight degrees of freedom not nine.

Equivalently, we can say that there is one constraint on \mathbf{P} . Since $\Omega^T = -\Omega$, it follows that

$$\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = \mathbf{n}\mathbf{t}^T + \mathbf{t}\mathbf{n}^T.$$

A dyadic product has rank one or less. The sum of two dyadic products has at most rank two. We conclude that

$$\det(\mathbf{P} + \mathbf{P}^T) = 0.$$

This constraint can be expressed in terms of the essential parameters as

$$p_1(p_5p_9 - p_6p_8) + p_4(p_3p_8 - p_2p_9) + p_7(p_2p_6 - p_3p_5) = 0.$$

We can use this equation, for example, to solve for p_9 given p_1, p_2, \dots, p_8 . It is difficult to use this constraint equation directly when one attempts to find \mathbf{P} from image brightness measurements.

There is a simple way around this problem. Note that $(\mathbf{r} \cdot \mathbf{s}) = 0$ because $\mathbf{s} = ((\mathbf{E}_r \times \hat{\mathbf{z}}) \times \mathbf{r})$. Therefore, we have $\mathbf{r}^T \mathbf{I} \mathbf{s} = 0$ (\mathbf{I} is the identity matrix) and

$$c + \mathbf{r}^T (\mathbf{P} + l\mathbf{I}) \mathbf{s} = 0 \quad \text{for arbitrary } l.$$

If we let $\mathbf{P}' = \mathbf{P} + l\mathbf{I}$, we can write

$$c + \mathbf{r}^T \mathbf{P}' \mathbf{s} = 0$$

to conclude that we cannot recover \mathbf{P} uniquely from image brightness measurements alone. To find \mathbf{P} we must impose the constraint $\det(\mathbf{P} + \mathbf{P}^T) = 0$. To avoid dealing directly with the resulting non-linear constraint among the essential parameters we first find any \mathbf{P}' that satisfies the above brightness change constraint equation for all image points being considered. We then determine l such that $\mathbf{P} = \mathbf{P}' - l\mathbf{I}$ satisfies $\det(\mathbf{P} + \mathbf{P}^T) = 0$. Since

$$\det(\mathbf{P} + \mathbf{P}^T) = \det(\mathbf{P}' + \mathbf{P}'^T - 2l\mathbf{I}) = 0,$$

$2l$ must be an eigenvalue of the real symmetric matrix

$$\mathbf{P}'' = \mathbf{P}' + \mathbf{P}'^T.$$

It will become apparent that we ought to choose the middle one of the three real eigenvalues of \mathbf{P}'' for $2l$.

To summarize, the overall plan is to find any matrix \mathbf{P}' that satisfies the image brightness constraint equation,

$$c + \mathbf{r}^T \mathbf{P}' \mathbf{s} = 0,$$

at a suitable number of image points and consequently determine \mathbf{P}'' . We can then solve for the middle eigenvalue of \mathbf{P}'' (which is $2l$) so as to construct the singular matrix $\mathbf{P} = \mathbf{P}' - l\mathbf{I}$. Consequently, we determine \mathbf{n} and \mathbf{t} as well as Ω (and, hence, ω) using the relationship

$$\mathbf{P} = -\Omega + \mathbf{n}\mathbf{t}^T.$$

4.2.3.2 Recovering Essential Parameters

We first consider estimating the essential parameters from the image brightness derivatives at the minimum number of points necessary. Then we consider a least-squares procedure that takes into account information in the whole image region.

We are looking for a matrix \mathbf{P}' that satisfies the brightness change equation,

$$c + \mathbf{r}^T \mathbf{P}' \mathbf{s} = 0,$$

at a chosen number of image points. Since

$$\mathbf{r}^T \mathbf{P}' \mathbf{s} = \text{Trace}\{(\mathbf{r}\mathbf{s}^T)\mathbf{P}'\}$$

or

$$\mathbf{r}^T \mathbf{P}' \mathbf{s} = \text{Flat}(\mathbf{r}\mathbf{s}^T) \cdot \text{Flat}(\mathbf{P}'),$$

where $\text{Flat}(\mathbf{M})$ is the vector obtained from the matrix \mathbf{M} by adjoining its rows, we can write the brightness change equation in the form

$$c + \mathbf{a}^T \mathbf{p}' = 0,$$

where

$$\mathbf{p}' = \text{Flat}(\mathbf{P}') = [p'_1, p'_2, \dots, p'_9]^T,$$

$$\mathbf{a} = \text{Flat}(\mathbf{r}\mathbf{s}^T) = [r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2, r_3 s_3]^T.$$

We can construct the vector \mathbf{a}_i from the derivatives of the brightness at the i^{th} image point such that

$$\mathbf{a}_i^T \mathbf{p}' = -c_i.$$

As discussed above, there are only eight independent degrees of freedom. Therefore, we can arbitrarily fix one of the diagonal elements of the matrix \mathbf{P}' and solve for the other eight using constraint equations derived from eight image points.

Let $\mathbf{p}' = [p'_1, p'_2, \dots, p'_8, 0]^T$ denote the solution obtained by setting the last element equal to zero. If we define

$$\tilde{\mathbf{p}}' = [p'_1, p'_2, \dots, p'_8]^T \quad \text{and} \quad \tilde{\mathbf{a}} = [r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2]^T,$$

then the above constraint equation reduces to

$$\tilde{\mathbf{a}}_i^T \tilde{\mathbf{p}}' = -c_i.$$

Using eight independent points we can solve the linear matrix equation

$$\mathbf{A} \tilde{\mathbf{p}}' = -\mathbf{c},$$

where

$$\mathbf{A} = [\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \dots, \tilde{\mathbf{a}}_8]^T \quad \text{and} \quad \mathbf{c} = [c_1, c_2, \dots, c_8]^T.$$

The solution of the above equation is

$$\tilde{\mathbf{p}}' = -\mathbf{A}^{-1} \mathbf{c}.$$

Image brightness values are corrupted by sensor noise and quantization. These inaccuracies are further accentuated by methods used to estimate the brightness gradient. Thus, it is not advisable to base a method on measurements at just a few points. We instead propose to minimize the error in the brightness constraint equation over the whole region I in the image plane; we choose the vector $\tilde{\mathbf{p}}'$ that minimizes

$$\iint_I (\tilde{\mathbf{a}}^T \tilde{\mathbf{p}}' + c)^2 dx dy.$$

The solution is given by

$$\tilde{\mathbf{p}}' = - \left(\iint_I \tilde{\mathbf{a}} \tilde{\mathbf{a}}^T dx dy \right)^{-1} \left(\iint_I c \tilde{\mathbf{a}} dx dy \right).$$

We can now construct \mathbf{p}' by adjoining a zero to the vector $\tilde{\mathbf{p}}'$. The result immediately gives us the matrix \mathbf{P}' . We then determine the eigenvalues of \mathbf{P}'' so that we can construct \mathbf{P}^* from $\mathbf{P}^* = \mathbf{P}'' - 2l\mathbf{I}$ (l is the middle eigenvalue of \mathbf{P}'') and \mathbf{P} from $\mathbf{P} = \mathbf{P}' - l\mathbf{I}$. At this point we are ready to recover t , ω , and \mathbf{n} .

Note that we do not have to repeat the eigenvalue-eigenvector analysis since \mathbf{P}^* has the same eigenvectors as \mathbf{P}'' and its eigenvalues are merely shifted so as to make the middle one equal to zero. This follows from the fact that if \mathbf{u} and λ form an eigenvector-eigenvalue pair of \mathbf{P}'' , that is, if

$$\mathbf{P}'' \mathbf{u} = \lambda \mathbf{u},$$

then \mathbf{u} and $(\lambda - 2l)$ form an eigenvector-eigenvalue pair of \mathbf{P}^* since

$$\mathbf{P}^* \mathbf{u} = (\mathbf{P}'' - 2l\mathbf{I}) \mathbf{u} = (\lambda - 2l) \mathbf{u}.$$

4.2.3.3 Recovering Motion and Surface Parameters

We first show how to compute the parameters of the translational motion and the plane orientation from the essential parameters. We can then find the rotational parameters using

$$\Omega = \mathbf{n}\mathbf{t}^T - \mathbf{P}.$$

As we saw before, we have

$$\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = \mathbf{t}\mathbf{n}^T + \mathbf{n}\mathbf{t}^T$$

since Ω is skew-symmetric. Let us use the notation $\sigma = \|\mathbf{n}\| \|\mathbf{t}\|$ and $\tau = \hat{\mathbf{n}} \cdot \hat{\mathbf{t}}$, where

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{\|\mathbf{n}\|} \quad \text{and} \quad \hat{\mathbf{t}} = \frac{\mathbf{t}}{\|\mathbf{t}\|}$$

are unit vectors in the directions of the surface normal and the translation vector, respectively. Then we obtain

$$\text{Trace}(\mathbf{P}^*) = \text{Trace}(\mathbf{P}) + \text{Trace}(\mathbf{P}^T) = 2\mathbf{n} \cdot \mathbf{t} = 2\sigma\tau.$$

It turns out that $\{\hat{\mathbf{n}}, \hat{\mathbf{t}}\}$ and $\{\sigma, \tau\}$ can be recovered easily from the eigenvectors and eigenvalues of the matrix \mathbf{P}^* . In the following lemma, we show that the eigenvectors of \mathbf{P}^* are combinations of the desired vectors $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$.

Lemma: Let $\mathbf{P}^* = \mathbf{U}\mathbf{A}\mathbf{U}^T$ be the eigenvalue decomposition of $\mathbf{P}^* = (\mathbf{t}\mathbf{n}^T + \mathbf{n}\mathbf{t}^T)$. If \mathbf{n} is not parallel to \mathbf{t} , then

$$\mathbf{A} = \text{Diag}(\sigma(\tau - 1), 0, \sigma(\tau + 1))$$

and

$$\mathbf{U} = \left[\frac{\hat{\mathbf{t}} - \hat{\mathbf{n}}}{\sqrt{2(1 - \tau)}}, \quad \frac{\hat{\mathbf{t}} \times \hat{\mathbf{n}}}{\sqrt{1 - \tau^2}}, \quad \frac{\hat{\mathbf{t}} + \hat{\mathbf{n}}}{\sqrt{2(1 + \tau)}} \right].$$

Proof: Note that

$$\mathbf{P}^* = \sigma(\hat{\mathbf{t}}\hat{\mathbf{n}}^T + \hat{\mathbf{n}}\hat{\mathbf{t}}^T).$$

Now, $(\hat{\mathbf{t}} \times \hat{\mathbf{n}})$ is the eigenvector with eigenvalue zero since

$$\mathbf{P}^*(\hat{\mathbf{t}} \times \hat{\mathbf{n}}) = \sigma(\hat{\mathbf{t}}\hat{\mathbf{n}}^T + \hat{\mathbf{n}}\hat{\mathbf{t}}^T)(\hat{\mathbf{t}} \times \hat{\mathbf{n}}) = \sigma\hat{\mathbf{t}}[\hat{\mathbf{n}}\hat{\mathbf{t}}\hat{\mathbf{n}}] + \sigma\hat{\mathbf{n}}[\hat{\mathbf{t}}\hat{\mathbf{n}}\hat{\mathbf{t}}] = 0.$$

Since \mathbf{P}^* is real symmetric, it has 3 orthogonal eigenvectors. The other two eigenvectors must, therefore, be in the plane containing $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$. Let $\mathbf{u} = \alpha\hat{\mathbf{t}} + \beta\hat{\mathbf{n}}$ and λ denote an

eigenvector-eigenvalue pair for some α and β that are to be determined. Then we must have

$$\sigma(\hat{\mathbf{t}}\hat{\mathbf{n}}^T + \hat{\mathbf{n}}\hat{\mathbf{t}}^T)(\alpha\hat{\mathbf{t}} + \beta\hat{\mathbf{n}}) = \lambda(\alpha\hat{\mathbf{t}} + \beta\hat{\mathbf{n}}).$$

This becomes

$$\sigma[\alpha(\hat{\mathbf{t}} \cdot \hat{\mathbf{n}}) + \beta(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}})]\hat{\mathbf{t}} + \sigma[\alpha(\hat{\mathbf{t}} \cdot \hat{\mathbf{t}}) + \beta(\hat{\mathbf{t}} \cdot \hat{\mathbf{n}})]\hat{\mathbf{n}} = \lambda\alpha\hat{\mathbf{t}} + \lambda\beta\hat{\mathbf{n}}.$$

Since $(\hat{\mathbf{t}} \cdot \hat{\mathbf{n}}) = \tau$, we can write

$$\begin{pmatrix} \sigma\tau - \lambda & \sigma \\ \sigma & \sigma\tau - \lambda \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

For this pair of homogeneous equations to have a non-trivial solution for α and β the determinant of the 2×2 coefficient matrix must be zero; that is,

$$(\sigma\tau - \lambda)^2 - \sigma^2 = 0 \quad \text{or} \quad \lambda = \sigma(\tau \pm 1).$$

Substituting for λ into the earlier equations we obtain $\alpha = \pm\beta$.

Note that $\sigma(\tau - 1) < 0$ and $\sigma(\tau + 1) > 0$ because $|\tau| < 1$ as it is the cosine of the angle between $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$. Therefore, one eigenvalue is negative and one is positive; this is why we choose to make the middle eigenvalue zero when constructing \mathbf{P}^* from \mathbf{P}^{**} . We find that eigenvectors corresponding to the eigenvalues $\lambda_1 = \sigma(\tau - 1)$ and $\lambda_3 = \sigma(\tau + 1)$ are $\hat{\mathbf{t}} - \hat{\mathbf{n}}$ and $\hat{\mathbf{t}} + \hat{\mathbf{n}}$, respectively. If we normalize these, we obtain the unit vectors

$$\mathbf{u}_1 = \frac{\hat{\mathbf{t}} - \hat{\mathbf{n}}}{\sqrt{2(1 - \tau)}} \quad \text{and} \quad \mathbf{u}_3 = \frac{\hat{\mathbf{t}} + \hat{\mathbf{n}}}{\sqrt{2(1 + \tau)}}. \blacksquare$$

We can determine $\sigma = \|\mathbf{n}\| \|\mathbf{t}\|$ and $\tau = \hat{\mathbf{t}} \cdot \hat{\mathbf{n}}$ from

$$\sigma = \frac{1}{2}(\lambda_3 - \lambda_1) \quad \text{and} \quad \tau = \frac{\lambda_3 + \lambda_1}{\lambda_3 - \lambda_1}.$$

The equations for \mathbf{u}_1 and \mathbf{u}_3 are linear in $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$ and so can be easily solved for these vectors,

$$\hat{\mathbf{n}} = \sqrt{\frac{1}{2}(1 + \tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1 - \tau)}\mathbf{u}_1 \quad \text{and} \quad \hat{\mathbf{t}} = \sqrt{\frac{1}{2}(1 + \tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1 - \tau)}\mathbf{u}_1.$$

The sign of the eigenvectors are arbitrary. If we change the sign of \mathbf{u}_1 we obtain

$$\hat{\mathbf{n}} = \sqrt{\frac{1}{2}(1 + \tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1 - \tau)}\mathbf{u}_1 \quad \text{and} \quad \hat{\mathbf{t}} = \sqrt{\frac{1}{2}(1 + \tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1 - \tau)}\mathbf{u}_1,$$

where $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$ are interchanged. This is the dual solution.

The signs of the two eigenvectors can be chosen independently. This might suggest that we may have a total of four different solutions for $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$. We show next that two of these solutions can be discarded because they correspond to viewing the planar surface "from behind".

We assume that the visible part of the plane is the bounding surface of some solid object. We chose to define the orientation of the surface using the inward pointing normal \mathbf{n} . The equation of the plane is $\mathbf{R} \cdot \mathbf{n} = 1$ or $(\mathbf{r} \cdot \mathbf{n})(\mathbf{R} \cdot \hat{\mathbf{z}}) = 1$ since $\mathbf{R} = (\mathbf{R} \cdot \hat{\mathbf{z}})\mathbf{r}$. $\mathbf{R} \cdot \hat{\mathbf{z}} = Z$ is positive for points in front of the viewer and so $\mathbf{r} \cdot \mathbf{n}$ must be positive for points on the visible portion of the plane. The equation $\mathbf{r} \cdot \mathbf{n} = 0$ corresponds to a line in the image. Points on one side of this line corresponding to $\mathbf{r} \cdot \mathbf{n} > 0$ can be images of points on the plane defined by the inward pointing normal \mathbf{n} . Conversely, points on the other side of the line where $\mathbf{r} \cdot \mathbf{n} < 0$ cannot. They can be thought of as images of points on a parallel but oppositely oriented plane corresponding to the vector $-\mathbf{n}$. We are analyzing brightness gradients for a particular image region. If $\mathbf{r} \cdot \mathbf{n} > 0$ for points in this region, \mathbf{n} is a possible solution for the surface normal. If $\mathbf{r} \cdot \mathbf{n} < 0$ for points in this region, then $-\mathbf{n}$ is a possible solution. If $\mathbf{r} \cdot \mathbf{n} > 0$ for some points and $\mathbf{r} \cdot \mathbf{n} < 0$ for others, we are not dealing with the image of a single planar surface.

Remember that we can recover \mathbf{t} and \mathbf{n} only up to a scale factor. We can, therefore, let \mathbf{t} be a unit vector without loss of generality. Then \mathbf{n} can be found from

$$\mathbf{n} = \|\mathbf{n}\| \hat{\mathbf{n}} = \|\mathbf{n}\| \|\mathbf{t}\| \hat{\mathbf{n}} = \sigma \mathbf{n}$$

using the known value of σ .

So far, we have assumed that \mathbf{n} and \mathbf{t} are not parallel. In the special case that $\hat{\mathbf{t}} \parallel \hat{\mathbf{n}}$ we have

$$\mathbf{P}^* = \sigma(\hat{\mathbf{t}}\hat{\mathbf{n}}^T + \hat{\mathbf{n}}\hat{\mathbf{t}}^T) = 2\sigma\hat{\mathbf{n}}\hat{\mathbf{n}}^T.$$

This dyadic product has rank one; that is, it only has one non-zero eigenvalue. This is easy to show since any vector perpendicular to $\hat{\mathbf{n}}$ is an eigenvector with zero eigenvalue. Also, $\hat{\mathbf{n}}$ is an eigenvector with eigenvalue 2σ . Therefore, when \mathbf{P}^* has two equal eigenvalues (that is, \mathbf{P}^* has two zero eigenvalues) we conclude that $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$ are parallel and equal to the eigenvector corresponding to the remaining eigenvalue.

We then solve for the rotation parameters by substituting the solutions for \mathbf{n} and \mathbf{t} into the equation

$$\mathbf{\Omega} = \mathbf{n}\mathbf{t}^T - \mathbf{P}.$$

It is easy to see that the dual solution gives rise to the same set of essential parameters. We showed that the dual solution is given by

$$\mathbf{n}' = \|\mathbf{n}\| \mathbf{t}, \quad \mathbf{t}' = \frac{\mathbf{n}}{\|\mathbf{n}\|}, \quad \text{and} \quad \boldsymbol{\omega}' = \boldsymbol{\omega} + \mathbf{n} \times \mathbf{t},$$

where we have arbitrarily set $\|\mathbf{t}\| = \|\mathbf{t}'\| = 1$. The closed-form solutions for \mathbf{n} and \mathbf{t} already satisfy the duality relationship given above. To show the same for $\boldsymbol{\omega}$, we use the identity

$$(\mathbf{n} \times \mathbf{t}) \times \mathbf{x} = \mathbf{t}(\mathbf{n} \cdot \mathbf{x}) - \mathbf{n}(\mathbf{t} \cdot \mathbf{x}) = (\mathbf{t}\mathbf{n}^T - \mathbf{n}\mathbf{t}^T)\mathbf{x}$$

which holds for any vector \mathbf{x} . Using this in

$$\boldsymbol{\omega}' \times \mathbf{x} = (\boldsymbol{\omega} + \mathbf{n} \times \mathbf{t}) \times \mathbf{x} = \boldsymbol{\omega} \times \mathbf{x} + (\mathbf{n} \times \mathbf{t}) \times \mathbf{x}$$

we arrive at

$$\boldsymbol{\omega}' \times \mathbf{x} = \boldsymbol{\omega} \times \mathbf{x} + (\mathbf{t}\mathbf{n}^T - \mathbf{n}\mathbf{t}^T)\mathbf{x}$$

or

$$\boldsymbol{\Omega}'\mathbf{x} = (\boldsymbol{\Omega} + \mathbf{t}\mathbf{n}^T - \mathbf{n}\mathbf{t}^T)\mathbf{x},$$

where the isomorphism between $\boldsymbol{\Omega}$ and $\boldsymbol{\omega}$ and between $\boldsymbol{\Omega}'$ and $\boldsymbol{\omega}'$ is used again. This must hold for all vectors \mathbf{x} ; therefore, we must have

$$\boldsymbol{\Omega}' = \boldsymbol{\Omega} + \mathbf{t}\mathbf{n}^T - \mathbf{n}\mathbf{t}^T.$$

We finally obtain

$$-\boldsymbol{\Omega}' + \mathbf{n}'\mathbf{t}' = -\boldsymbol{\Omega} + \mathbf{n}\mathbf{t}^T - \mathbf{t}\mathbf{n}^T + \mathbf{t}\mathbf{n}^T$$

or

$$-\boldsymbol{\Omega}' + \mathbf{n}'\mathbf{t}' = -\boldsymbol{\Omega} + \mathbf{n}\mathbf{t}^T = \mathbf{P}.$$

We conclude that \mathbf{n}' , \mathbf{t}' , and $\boldsymbol{\omega}'$ constitute a second solution since they lead to the same set of essential parameters.

4.3 Some Examples

The examples given in this section represent the results of several simulations of the iterative and the analytical methods for planar surfaces. These examples serve to show the convergence rate of the two iterative schemes, a comparison of the results from the iterative and analytical methods, and the sensitivity of the solution to modeling errors and noisy data.

We have used synthetic data since then the underlying motion is known precisely. Further, in the presence of noise, we can determine the correlation between the (percentage of) noise and the (percentage) degradation from the correct solution. The image plane is a square whose length is two units (twice the focal length) and is discretized to 32 image points in either direction.

To generate perfect synthetic data, we first choose the brightness gradient, E_r , randomly and then compute the temporal derivative, E_t , from the equation

$$E_t = -(\mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t})),$$

where, as shown earlier,

$$\mathbf{s} = (E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r} \quad \text{and} \quad \mathbf{v} = \mathbf{r} \times \mathbf{s}.$$

To generate noisy data, we then add random noise to perfect synthetic data (perfect $E_{\mathbf{r}}$ and $E_{\mathbf{t}}$). This represents errors due to noise in the images and quantization. In practice, first the brightness at image points in two frames are discretized and then the gradients are computed using finite difference methods.

4.3.1 A Comparison of the Two Iterative Schemes

In this example, we compare the two iterative schemes given for planar surfaces using perfect data. This example represents a particular situation when the convergence rate is rather rapid. The true motion parameters are $\boldsymbol{\omega} = [0.005, 0.0075, 0.01]^T$ and $\mathbf{t} = [0.005, -0.005, 0.005]^T$ and the surface normal is $\mathbf{n} = [0.2, 0.4, 1.0]^T$. Figures 4.1 and 4.2 show the results of two simulations when the first algorithm converges to the true and the dual solutions. Figures 4.3 and 4.4 show similar results for the second iterative scheme. The initial conditions were chosen randomly. For this particular example, the first scheme converges to the true solution in about 10 iterations and to the dual solution in about 20 iterations. The convergence rate using the second iterative scheme is about 4 iterations for the true solution and about 8 iterations for the dual solution. In other simulations using random initial conditions, the same convergence rate has been observed for this particular example.

4.3.2 A Case of the Unique Solution for Planar Surfaces

In this example, we present the results of the first iterative scheme in a case when the solution is unique. The motion parameters are $\boldsymbol{\omega} = [0.04, -0.04, 0.08]^T$ and $\mathbf{t} = [0.1, 0.2, 0.2]^T$ and the surface normal is $\mathbf{n} = [0.5, 1, 1]^T$ (this is called the correct solution). Figure 4-5 shows the solution obtained after various number of iterations of the first iterative scheme (this is called the true solution) and the solution obtained using the true solution in the equations for a dual solution (this is called the dual solution). The convergence is obtained when the two (the true and the dual) solutions become equal to the correct solution. Note that the convergence is rather slow for either the true or the dual solution. In fact, the algorithm does not converge to the solution even after 200 iterations; however, the average of the two solutions converges to the correct solution in about 50 iterations. This is a general result that can be exploited to obtain an accurate estimate of the solution when the translation vector is (near) parallel to the surface normal.

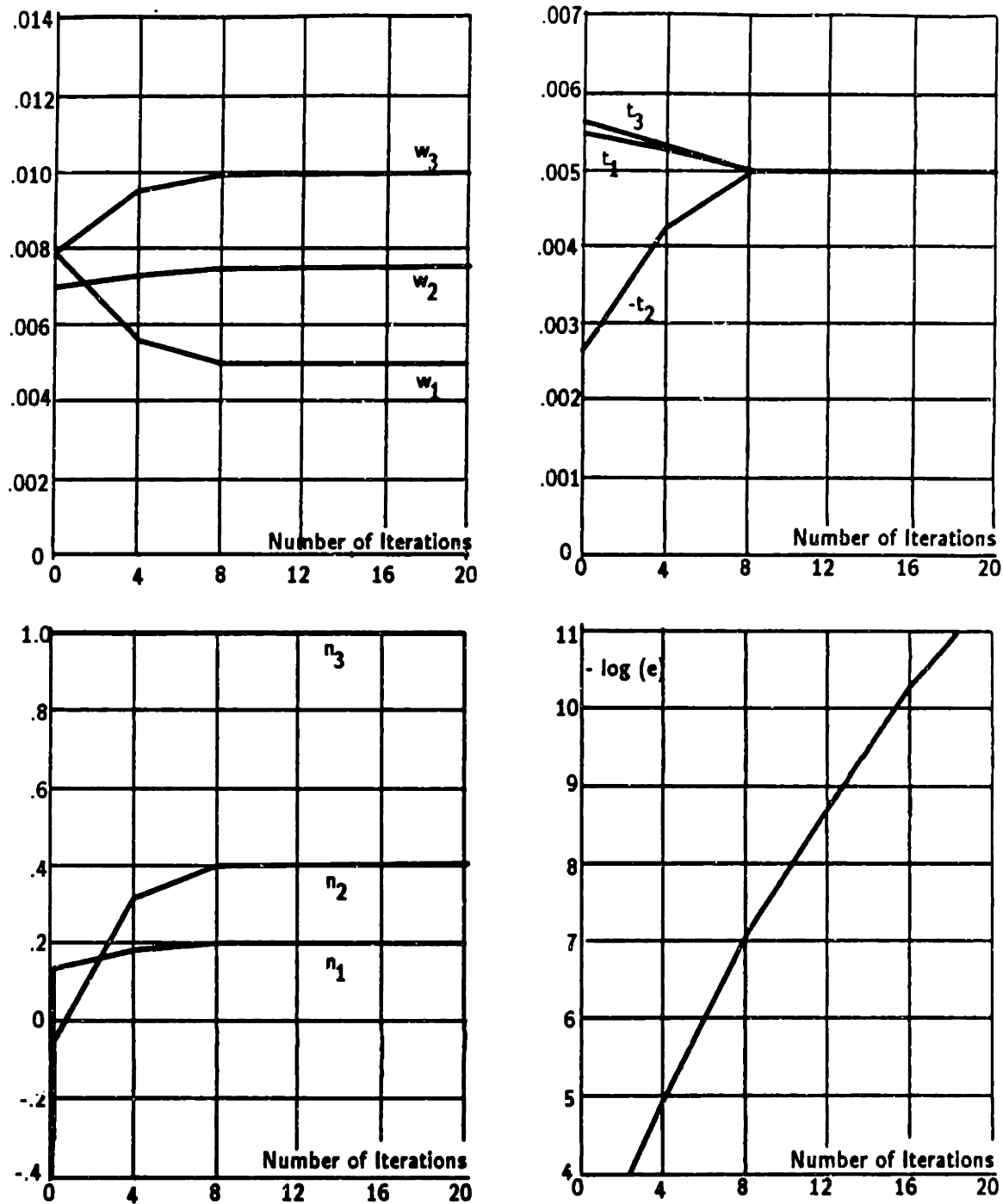


Figure 4-1. A simulation converging to the true solution using the first scheme.

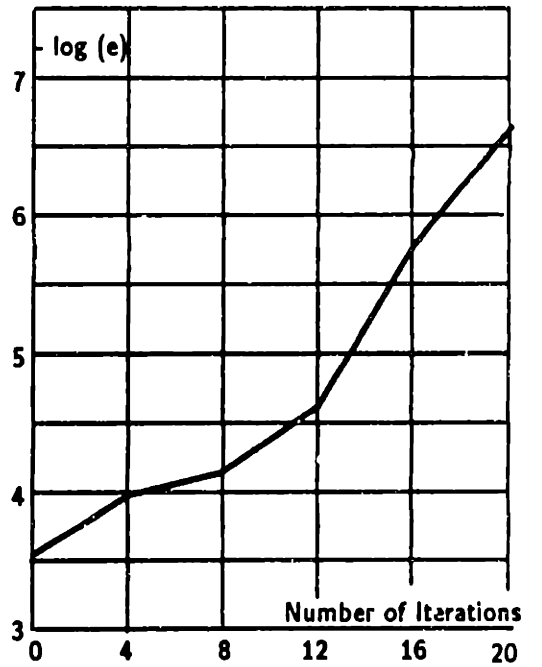
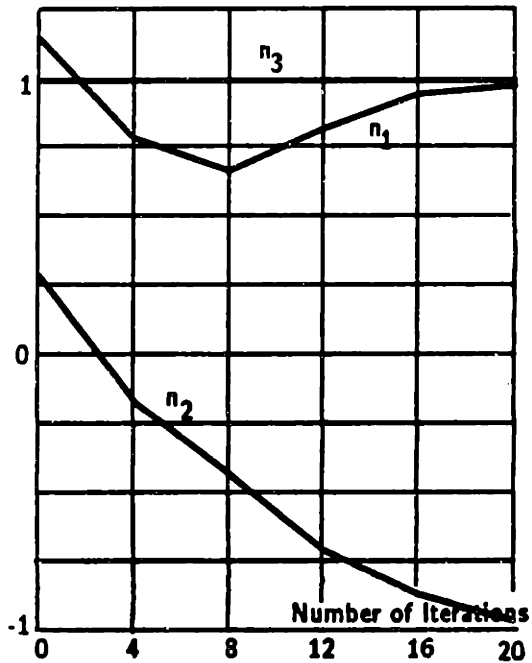
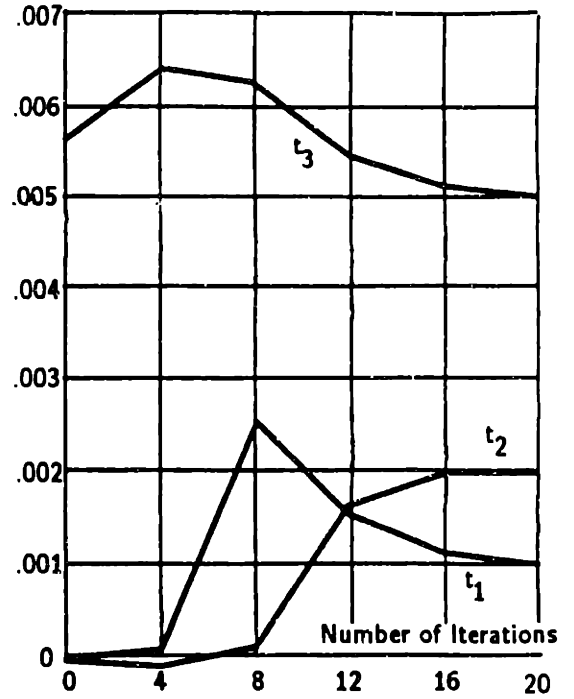
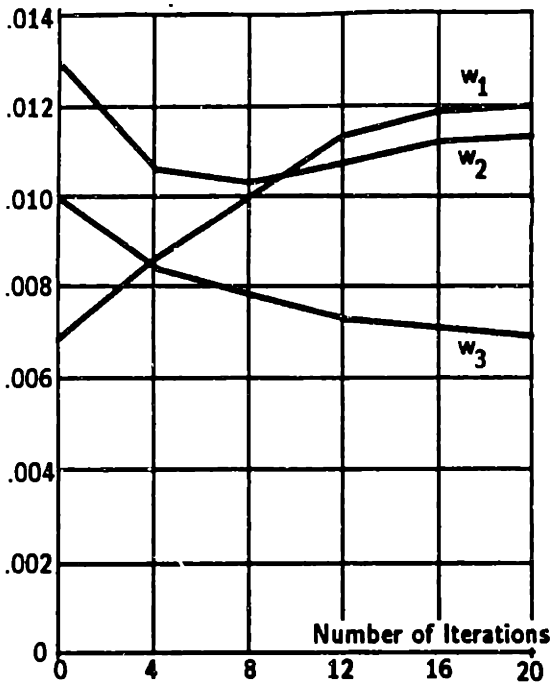


Figure 4-2. A simulation converging to the dual solution using the first scheme.

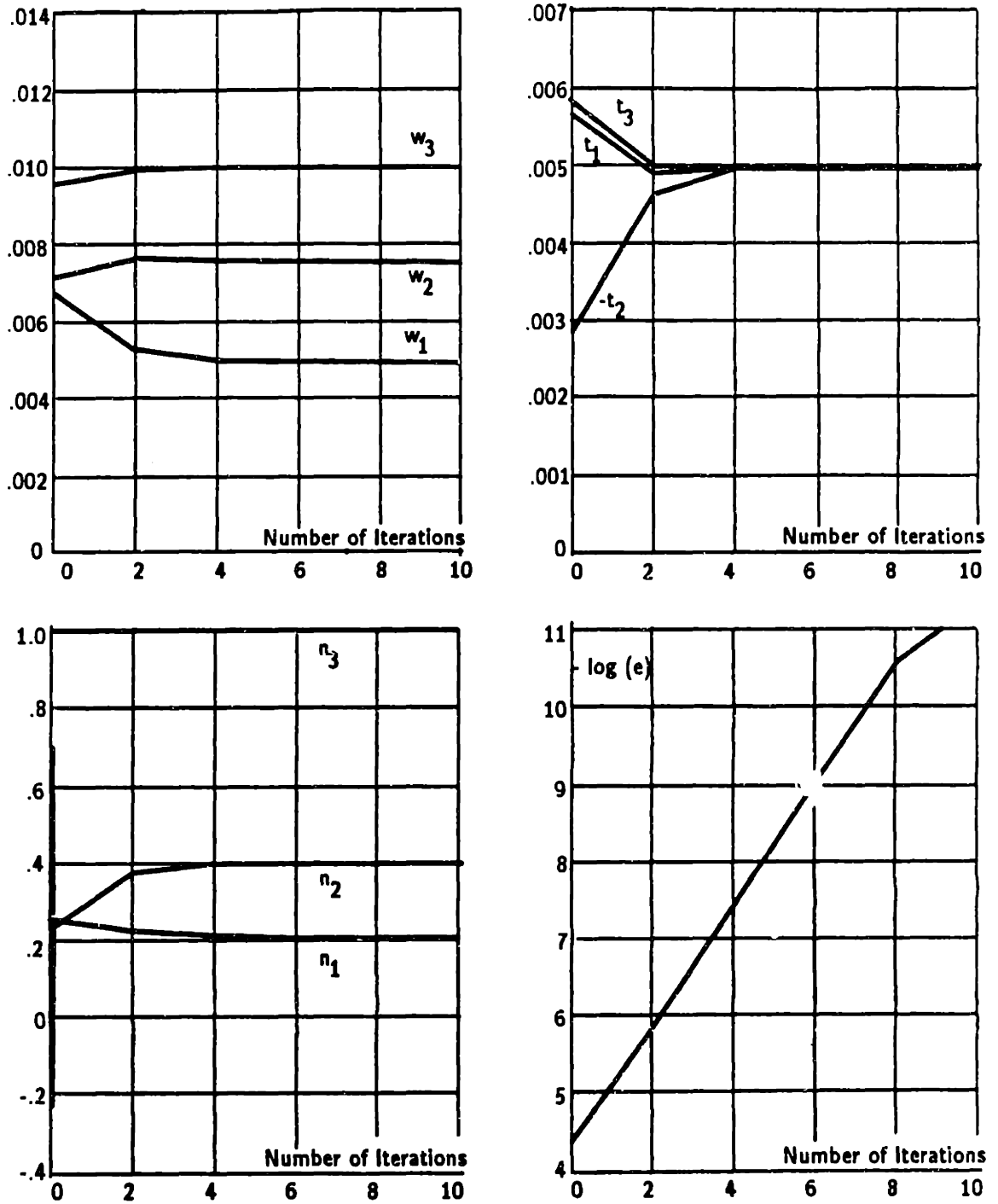


Figure 4-3. A simulation converging to the true solution using the second scheme.

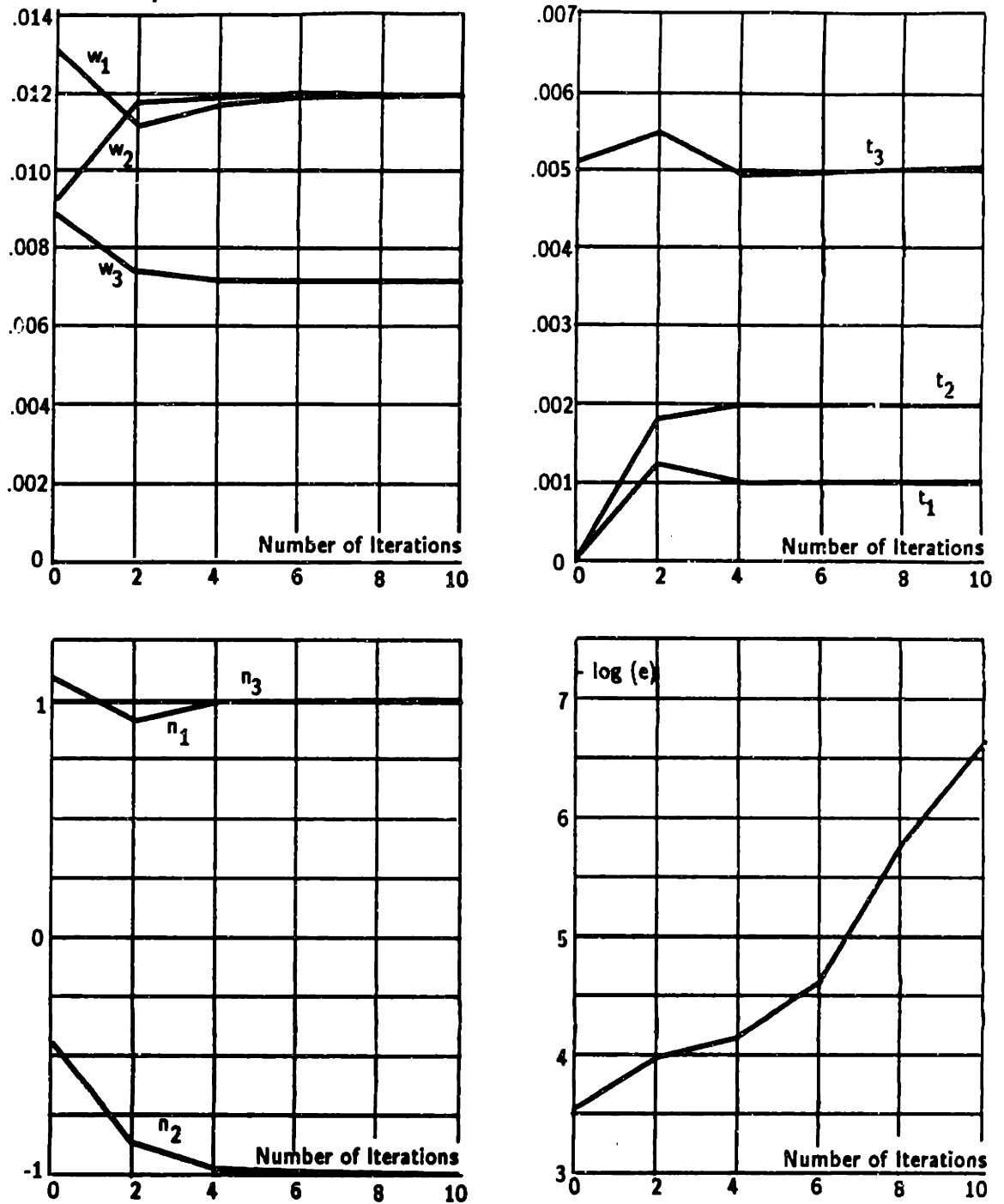


Figure 4-4. A simulation converging to the dual solution using the second scheme.

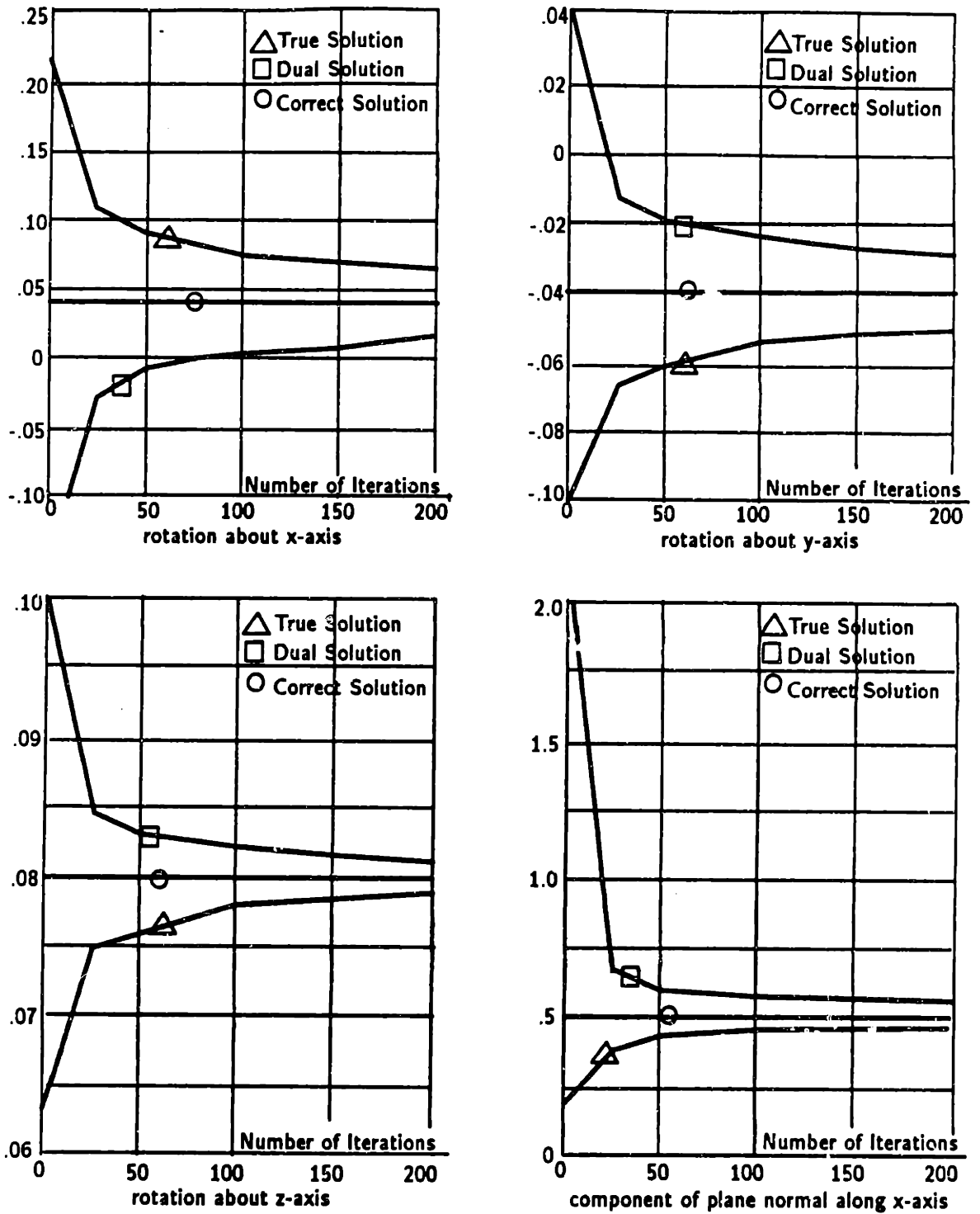


Figure 4-5. Results of a simulation where the solution is unique.

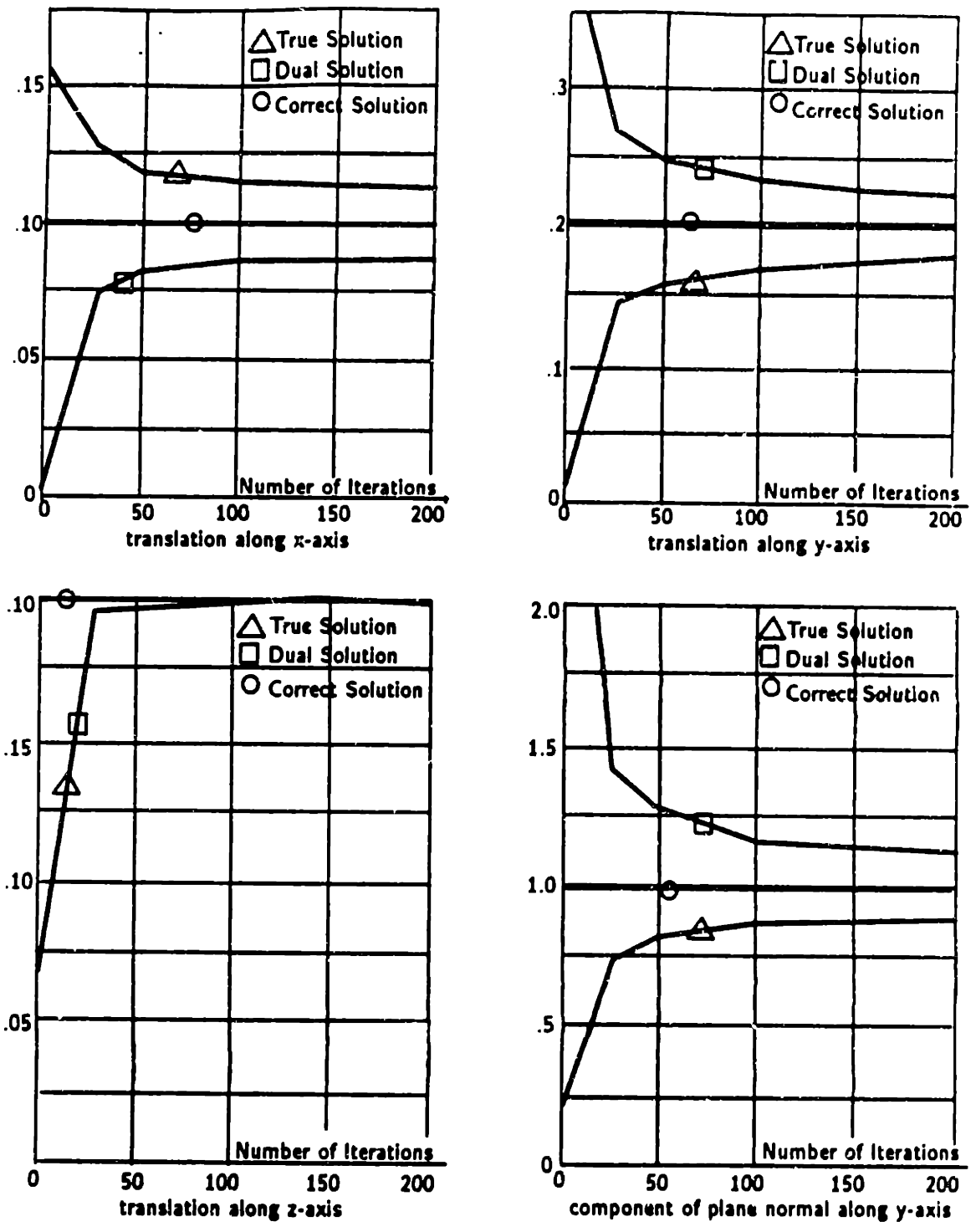


Figure 4-5. Continued.

4.3.3 Sensitivity to Image Noise: Iterative Methods

In the previous example, we demonstrated the slow convergence rate of the iterative scheme when the solution is unique. We will now study the behavior of the solution in the presence of noise in the data. The motion and surface parameters are given as in the previous example. Table 4.1 shows the final results of several simulations after 100 iterations of the algorithm when the noise is increased to a mean of about 30%. The initial condition was chosen randomly. For each noise level, three solutions are given; they are the solution that the algorithm converged to (the true solution), its dual (dual solution), and the average of these two solutions. The true and dual solutions give reasonable estimates of motion and surface parameters; however, the estimate obtained by averaging the two solutions gives a very accurate estimate even when the noise is as much as 30%. Note that the error (the sum of the square of the error in the brightness change equation over the image) for the true solution is slightly less than that for the correct solution. This reflects the fact that the correct solution is no longer the optimal least-squares solution in the presence of noise in the data.

4.3.4 A Comparison of the Analytical and Iterative Methods

In this example, we compare the results obtained for the two methods (iterative and analytical) given for planar surfaces. The true surface is quadratic. The tangent plane of the surface is $\mathbf{n} = [0.02, 0.02, 1]^T$, and the curvature parameters, $\mathbf{d} = [d_{11}, d_{12}, d_{22}]^T$, are given by $\mathbf{d} = [0.5, 0.2, 0.5]^T$; that is, the true surface is an elliptic paraboloid (see next chapter for the equation of a quadratic surface). The camera motion is $\boldsymbol{\omega} = [.05, .05, -.08]^T$ and $\mathbf{t} = [.1, -.1, .05]^T$. A large field of view with a half-angle of 45 degrees has been used so that the second-order terms of the surface are not negligible.

Table 4.2 shows the results of several simulations. It is immediately apparent that the solutions obtained from the iterative method and the analytical method are identical. This has occurred in every case we have tested. Surprisingly, Scott [1986] reports that the iterative method gives a much better estimate than that obtained using the analytical method in his experiments with real images (this may be a consequence of an incorrect implementation as explained in Appendix A). Note that the estimates of the motion parameters are rather robust with respect to noise since the solution changes continuously with the noise level.

4.3.5 Sensitivity to Modeling Error and Image Noise: A Saddle Point

In this example, we test the behavior of the solution of the closed-form method when the surface is a saddle surface. The camera motion is $\boldsymbol{\omega} = [0.05, 0.05, -0.08]^T$ and $\mathbf{t} = [0.1, -0.1, 0.05]^T$. The tangent plane of the surface is $\mathbf{n} = [0.02, 0.02, 1]^T$ and the curvature parameters are varied from $\mathbf{d} = [0.5, 0.75, 0.5]^T$ to $\mathbf{d} = [0.5, 2.5, 0.5]^T$. Fixing the first and third components of \mathbf{d} leaves the mean curvature constant and increasing

the second component increases the Gaussian curvature of the surface; in effect, we are increasing the curvature of the surface at the saddle point. For each choice of the vector d , the noise was varied from a standard deviation of 0% to 57.3%. Again, we have used a larger field of view, a half-angle of 45 degrees, in order to investigate the behavior of the solution when the second order terms of the surface are not negligible.

Tables 4.3a-c show the results as d is varied. Note that the estimates of the components of rotational motion about the optical axis have been determined with good accuracy. This is because a large field of view is used. A simple explanation is that a rotation about the optical axis gives rise to flow lines that form concentric circles with no other consistent interpretation that a rotation about the optical axis. From these results, we can see that the solution deteriorates monotonically as the curvature is increased. This is predictable since as the curvature increases, a planar model is not an accurate representation of the surface anymore.

4.4 Summary

We have addressed the problem of recovering the motion of a viewer relative to a planar surface from a sequence of images that are used to estimate the temporal and spatial derivatives of image brightness (note that this can be done only up to the scale factor ambiguity). We have shown that the problem usually has two solutions. The exception is when the direction of translation of the viewer and the surface normal are parallel in which case the solution is unique. Also, when the direction of translation is parallel to the image plane, the dual solution is considered a degenerate solution since the image of the dual plane reduces to a line corresponding to viewing the planar surface from the side.

The viewer motion can be recovered from six linear equations if the surface of the scene is a frontal plane (or can be approximated as such); that is, the points on the surface are located (approximately) at the same depth from the viewer. When the orientation of the surface is arbitrary, the viewer motion and the orientation of the planar surface can be recovered by solving eight non-linear equations. By exploiting the special structure of these equations, we have proposed two iterative schemes for solving them (actually, we solved nine equations without exploiting the scale-factor ambiguity directly). Once we obtain one of the two possible solutions from the iterative scheme, the dual solution can be determined from analytical expressions. We have also shown that the unknown parameters can be recovered analytically through a two-step procedure. Using the brightness derivatives over the whole or relevant region of the image plane, we first determine a set of intermediate parameters, the so-called essential parameters of planar surfaces, that are the elements of a particular 3×3 matrix. The eigenvalue-eigenvector decomposition of this matrix is then used to determine the motion parameters and the surface orientation.

In Appendix A, we explain why the two (iterative and analytical) methods give the same estimate. This requires, as we showed, incorporating the scale-factor ambiguity as

a constraint among the essential parameters before these parameters are computed from brightness gradients. If this is not enforced, the damage is done. More precisely, the estimates of the essential parameters, and consequently the solution of the motion and surface parameters, will be inaccurate. Furthermore, the solution is generally non-unique.

For the cases we have tested, the rate of convergence of the two iterative schemes seems to depend more on the particular situation that is simulated, namely, on the motion of the camera and the surface being viewed. The choice of the initial condition does not seem to be an important factor; however, the convergence rate is slightly slower when the initial condition is near the boundary of the convergence regions of the two possible solutions (this is the region where some perturbation of the initial condition causes the algorithm to converge to one or the other solution). We have not determined what cases require less number of iterations; however, we know that the slowest rate of convergence corresponds to the special case when the solution is unique. This occurs when the translation vector is parallel to the plane normal.

Through several examples, we have shown the convergence rate of the two iterative methods and the sensitivity of the solution to noisy data and modeling errors. The second iterative scheme usually converges to the solution in half the number of iterations required for the first scheme, but it requires more computation per iteration. On the average, a good estimate of the solution can be obtained in about 40 iterations of the first scheme and in about 20 iterations of the second scheme. When the translation vector is (near) parallel to the surface normal (that is, when the solution is unique), several hundred iterations of either scheme are usually required to obtain a good estimate of the solution, however, the average of the true and dual solutions obtained after about 40-50 iterations gives a very good estimate of the correct solution. In our simulation results using synthetic data, the iterative and the analytical methods have given the same estimate of the solution as expected. This seems to be in contradiction with the results obtained by Scott [1986] using real images. He reports that the iterative scheme gives a better estimate of the solution than the analytical method. This may be because of an incorrect implementation of the analytical method as explained in Appendix A.

Table 4.1: Unique Solution of Planar Surfaces; Results of the First Iterative Scheme

True Solution: $\omega = [0.04, -0.04, 0.08]^T$ $t = [0.1, 0.2, 0.2]^T$ $n = [0.5, 1, 1]^T$

Initial Condition: $n = [0.0590, -0.0232, -0.0548]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
0%	0%	0	0
Rotation Parameters		Translation Parameters	Surface Parameters
T 0.0576	-0.0621 0.0929	0.1228 0.2175 0.1990	0.3978 0.9160 1.000
D 0.0224	-0.0185 0.0670	0.0792 0.1823 0.1990	0.6171 1.0933 1.000
A 0.0400	-0.0403 0.0800	0.1010 0.1999 0.1990	0.5074 1.0046 1.000

Initial Condition: $n = [-0.0586, 0.0240, 0.0335]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
10%	5.7%	.019	.019
Rotation Parameters		Translation Parameters	Surface Parameters
T 0.0270	-0.0529 0.0989	0.1124 0.1866 0.1988	0.4340 1.068 1.000
D 0.0527	-0.0268 0.0599	0.0863 0.2123 0.1988	0.5654 0.9383 1.000
A 0.0399	-0.0399 0.0704	0.0994 0.1995 0.1988	.4997 1.003 1.000

Initial Condition: $n = [0.0229, -0.0475, 0.0157]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
25%	14.2%	.129	.127
Rotation Parameters		Translation Parameters	Surface Parameters
T 0.0015	-0.0324 0.0937	0.0905 0.1591 0.1967	0.5593 1.247 1.000
D 0.0847	-0.0519 0.0698	0.1100 0.2453 0.1967	0.4600 0.8087 1.000
A 0.0423	-0.0421 0.0818	0.1002 0.2022 0.1967	0.5097 1.028 1.000

Initial Condition: $n = [-0.0526, -0.0614, 0.0254]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
50%	28.2%	.510	.496
Rotation Parameters		Translation Parameters	Surface Parameters
T 0.0035	-0.0065 0.0626	0.0630 0.1501 0.1872	0.6642 1.243 1.000
D 0.0791	-0.0679 0.0840	0.1244 0.2328 0.1872	0.3364 0.809 1.000
A 0.0394	0.0372 0.0733	0.0937 0.1915 0.1872	.5003 1.026 1.000

Table 4.2: Iterative Scheme -vs- Closed-Form Method for Planes

True Surface is Quadratic: $\mathbf{n} = [0.02, 0.02, 1.0]^T$ $\mathbf{d} = [.5, .25, .5]^T$
 True Motion Parameters: $\boldsymbol{\omega} = [0.05, 0.05, -0.08]^T$ $\mathbf{t} = [0.1, -0.1, 0.05]^T$

Initial Condition: $\mathbf{n} = [0.0018, 0.0149, -0.0065]^T$

Error

Max. Image Noise	S.D. Noise	True Solution	Iterative Solution	Closed-Form Solution
0%	0%	.095	.034	.034

	Rotation Parameters	Translation Parameters	Surface Parameters
Iterative:	0.0457 0.0530 -0.0793	0.1150 -0.1230 0.0666	0.0573 -0.0261 1.000
Closed-Form:	0.0457 0.0530 -0.0793	0.1150 -0.1230 0.0666	0.0573 -0.0261 1.000

Initial Condition: $\mathbf{n} = [0.0243, -0.0095, 0.0296]^T$

Error

Max. Image Noise	S.D. Noise	True Solution	Iterative Solution	Closed-Form Solution
10%	5.7%	.129	.075	.075

	Rotation Parameters	Translation Parameters	Surface Parameters
Iterative:	0.0469 0.0544 -0.0779	0.1133 -0.1200 0.0662	0.0876 -0.0300 1.000
Closed-Form:	0.0469 0.0544 -0.0779	0.1133 -0.1200 0.0662	0.0876 -0.0301 1.000

Initial Condition: $\mathbf{n} = [0.0255, 0.0034, -0.0346]^T$

Error

Max. Image Noise	S.D. Noise	True Solution	Iterative Solution	Closed-Form Solution
25%	14.1%	.334	.302	.302

	Rotation Parameters	Translation Parameters	Surface Parameters
Iterative:	0.0480 0.0508 -0.0792	0.1136 -0.1152 0.0688	0.0940 -0.0352 1.000
Closed-Form:	0.0480 0.0508 -0.0792	0.1136 -0.1152 0.0688	0.0940 -0.0352 1.000

Initial Condition: $\mathbf{n} = [-0.0402, -0.0306, -0.0572]^T$

Error

Max. Image Noise	S.D. Noise	True Solution	Iterative Solution	Closed-Form Solution
50%	28.4%	.914	.895	.895

	Rotation Parameters	Translation Parameters	Surface Parameters
Iterative:	0.0586 0.0394 -0.0876	0.1209 -0.1152 0.0633	-0.0165 -0.0734 1.000
Closed-Form:	0.0586 0.0394 -0.0876	0.1209 -0.1152 0.0633	-0.0165 -0.0734 1.000

Table 4.3a: Simulation Results of the Analytical Method for Planes;
the true surface is quadratic with a saddle point

$$\begin{aligned} \text{True Solution: } \boldsymbol{\omega} &= [0.05, 0.05, -0.08]^T & \mathbf{t} &= [0.1, -0.1, 0.05]^T \\ \mathbf{n} &= [0.02, 0.02, 1.0]^T & \mathbf{d} &= [.5, .7, .5]^T \end{aligned}$$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
0%	0%	.197	.096
Rotation Parameters		Translation Parameters	Surface Parameters
0.0457	0.0530 -0.0793	0.1150 -0.1230 0.0666	0.0573 -0.0261 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
10%	5.7%	.239	.139
Rotation Parameters		Translation Parameters	Surface Parameters
0.0210	0.0298 -0.0781	0.1471 -0.1574 0.0716	0.0874 -0.0660 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
25%	14.1%	.428	.343
Rotation Parameters		Translation Parameters	Surface Parameters
0.0218	0.0322 -0.0772	0.1423 -0.1517 0.0724	0.0836 -0.0846 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
50%	28.4%	1.08	1.01
Rotation Parameters		Translation Parameters	Surface Parameters
0.0041	0.0344 -0.0696	0.1326 -0.1638 0.0574	0.1412 0.0004 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
100%	57.3%	3.90	3.63
Rotation Parameters		Translation Parameters	Surface Parameters
0.0333	0.0084 -0.0730	0.1285 -0.0959 0.0549	0.0108 -0.1695 1.000

Table 4.3b: Simulation Results of the Analytical Method for Planes;
the true surface is quadratic with a saddle point

$$\begin{aligned} \text{True Solution: } \boldsymbol{\omega} &= [0.05, 0.05, -0.08]^T & \mathbf{t} &= [0.1, -0.1, 0.05]^T \\ \mathbf{n} &= [0.02, 0.02, 1.0]^T & \mathbf{d} &= [.5, 1.5, .5]^T \end{aligned}$$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
0%	0%	.531	.270
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0088	-0.0147 -0.0821	0.2062 -0.1991 0.0863	0.1260 -0.1285 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
10%	5.7%	.582	.310
Rotation Parameters		Translation Parameters	Surface Parameters
-.0113	-0.0185 -0.0803	0.2103 -0.2022 0.0866	0.1288 -0.1123 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
25%	14.1%	.797	.543
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0061	-0.0111 -0.0809	0.2008 -0.1951 0.0887	0.1364 -0.1178 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
50%	28.4%	1.46	1.25
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0199	-0.0253 -0.0779	0.2049 -0.1966 0.0627	0.0059 -0.0709 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
100%	57.3%	4.32	3.94
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0143	-0.0101 -0.0644	0.1662 -0.1599 0.0765	0.1932 -0.0613 1.000

Table 4.3c: Simulation Results of the Analytical Method for Planes;
the true surface is quadratic with a saddle point

$$\begin{aligned} \text{True Solution: } \boldsymbol{\omega} &= [0.05, 0.05, -0.08]^T & \mathbf{t} &= [0.1, -0.1, 0.05]^T \\ \mathbf{n} &= [0.02, 0.02, 1.0]^T & \mathbf{d} &= [.5, 2.5, .5]^T \end{aligned}$$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
0%	0%	1.36	.661
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0569	-0.0675 -0.0843	0.2783 -0.2651 0.1028	0.1509 -0.1656 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
10%	5.7%	1.41	.705
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0603	-0.0700 -0.0835	0.2820 -0.2688 0.0988	0.1500 -0.1548 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
25%	14.1%	1.64	.913
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0577	-0.0696 -0.0843	0.2802 -0.2656 0.1014	0.1420 -0.1738 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
50%	28.4%	2.46	1.75
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0686	-0.0584 -0.0812	0.2520 -0.2691 0.1029	0.1576 -0.1832 1.000
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
100%	57.3%	5.88	4.78
Rotation Parameters		Translation Parameters	Surface Parameters
-0.0452	-0.0469 -0.0709	0.1958 -0.1975 0.0881	0.1264 -0.2647 1.000

Quadratic Surfaces

In this section, we focus our attention on smooth surfaces that are either quadratic or can be approximated in a large region by a quadratic surface. We need a large region only so that its projection onto the image covers a large portion of the image. Consequently, we will have a larger field of view (large FOV) to estimate the motion parameters robustly.

Consider a smooth surface $f(X, Y, Z) = 0$ that is at least twice differentiable in some neighborhood. We assume this region is around the line of sight. Then the Taylor expansion of the surface about some point O in this region, assumed to be the point $(0, 0, Z_0)$ for simplicity, is given by

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial X^2}\right)_0 X^2 + \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 XY + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial Y^2}\right)_0 Y^2 + O(\epsilon),$$

where $O(\epsilon)$ denotes the error term. If $O(\epsilon)$ is negligible, the Taylor expansion up to the second order is a good approximation of the surface and the surface can be labeled as a quadratic one in that neighborhood. The normal of the plane tangent to the surface at O is

$$\mathbf{n} = \frac{1}{Z_0} \left[-\left(\frac{\partial Z}{\partial X}\right)_0, -\left(\frac{\partial Z}{\partial Y}\right)_0, 1 \right]^T$$

and the matrix

$$\mathbf{H} = \begin{pmatrix} \left(\frac{\partial^2 Z}{\partial X^2}\right)_0 & \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 \\ \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 & \left(\frac{\partial^2 Z}{\partial Y^2}\right)_0 \end{pmatrix}$$

is the Hessian of the surface at the origin.

Since we are dealing with images, we need to express the surface in terms of the image coordinates; that is, we write $Z = Z(x, y)$. This can be done using the perspective

5.5 Summary

In this section, we presented an iterative method for recovering the motion parameters and the surface parameters of a quadratic surface (or a surface that can be approximated by a quadratic surface) from a set of 12 non-linear equations. This method is an extension of the first iterative method we presented for planar surfaces. Through a selected example, we demonstrated the convergence rate of the algorithm.

For most iterative methods, the convergence as well as the rate of convergence of the scheme will depend on the initial guess of the solution used to start the iterative process. This is a drawback of most iterative methods that makes them relatively less attractive. Interestingly, the convergence rate of our iterative algorithm, in the cases we have tested, depended more on the type of surface and motion involved than on the initial condition. With noisy data, the algorithm sometimes converges to a local minimum depending on the initial condition. This occurs mostly in two situations; when the solution is nearly ambiguous (this will be addressed in the next section), or when the surface curvature is small. In the first, the noise may perturb the data enough to make it resemble that corresponding to an ambiguous situation. In the second case, the algorithm may converge to a solution that is the dual of the best planar surface solution, the so-called "pseudo-dual" solution. This behavior is not surprising since with noisy data, the surface can be mistaken to be planar. Then, a second solution always exists unless the translational vector and the normal of the tangent plane are parallel. In most other cases, the algorithm converges to the optimum solution (this is different from the true solution in the presence of noise) from 50 to 120 iterations (depending on the type of the surface and motion).

We should emphasize that the behavior described above is shared by many iterative methods given for solving non-linear equations. The robust methods given for recovering motion and structure using local optical flow data or correspondence of image points also require the solution of a set of non-linear equations through an iterative procedure; though, in our case, we need to compute only the first derivatives of brightness function that can be computed easily from time-varying images.

projection equation, $\mathbf{r} = \frac{1}{Z}\mathbf{R}$. It is more convenient to consider the Taylor expansion of $d = 1/Z$ instead since the brightness change constraint equation is in terms of the inverse of the depth function (this is simply a consequence of perspective projection). The Taylor series expansion of d up to the second order terms in terms of the image coordinates is

$$d = d_0 + \left(\frac{\partial d}{\partial x}\right)_0 x + \left(\frac{\partial d}{\partial y}\right)_0 y + \frac{1}{2}\left(\frac{\partial^2 d}{\partial x^2}\right)_0 x^2 + \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 xy + \frac{1}{2}\left(\frac{\partial^2 d}{\partial y^2}\right)_0 y^2.$$

The coefficients of the Taylor series expansion of Z in terms of the object coordinates are related to those of d in terms of the image coordinates. These are

$$d_0 = \frac{1}{Z_0}, \quad \left(\frac{\partial d}{\partial x}\right)_0 = -\frac{1}{Z_0}\left(\frac{\partial Z}{\partial X}\right)_0, \quad \left(\frac{\partial d}{\partial y}\right)_0 = -\frac{1}{Z_0}\left(\frac{\partial Z}{\partial Y}\right)_0,$$

$$\left(\frac{\partial^2 d}{\partial x^2}\right)_0 = -\left(\frac{\partial^2 Z}{\partial X^2}\right)_0, \quad \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 = -\left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0, \quad \text{and} \quad \left(\frac{\partial^2 d}{\partial y^2}\right)_0 = -\left(\frac{\partial^2 Z}{\partial Y^2}\right)_0.$$

The expression for $d = 1/Z$ can be written in a vector or a matrix form. If we let $\mathbf{r} = (x, y, 1)^T$ as before, then we obtain

$$d = \frac{1}{2} \mathbf{r}^T \mathbf{D} \mathbf{r},$$

where

$$\mathbf{D} = \begin{pmatrix} \left(\frac{\partial^2 d}{\partial x^2}\right)_0 & \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 & \left(\frac{\partial d}{\partial x}\right)_0 \\ \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 & \left(\frac{\partial^2 d}{\partial y^2}\right)_0 & \left(\frac{\partial d}{\partial y}\right)_0 \\ \left(\frac{\partial d}{\partial x}\right)_0 & \left(\frac{\partial d}{\partial y}\right)_0 & 2d_0 \end{pmatrix}.$$

Even though the above representation is very compact, it is easier in deriving our results to write d in the form

$$d = \mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d},$$

where

$$\mathbf{q} = \left[\frac{1}{2}x^2, xy, \frac{1}{2}y^2\right]^T, \quad \mathbf{n} = \left[\left(\frac{\partial d}{\partial x}\right)_0, \left(\frac{\partial d}{\partial y}\right)_0, d_0\right]^T, \quad \text{and} \quad \mathbf{d} = \left[\left(\frac{\partial^2 d}{\partial x^2}\right)_0, \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0, \left(\frac{\partial^2 d}{\partial y^2}\right)_0\right]^T.$$

We will assume that the above equation is an accurate representation of the surface in a large portion of the image onto which the surface projects. Again, we emphasize the need for considering a large portion of the image, that is, a large field of view, in order to be able to estimate the motion parameters with good accuracy.

5.1 Recovering Motion and Surface Parameters

If we substitute the expression for $d = 1/Z$ into the brightness change equation, we obtain

$$c + (\mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

There are 12 unknowns, the three components of each of $\boldsymbol{\omega}$, \mathbf{t} , \mathbf{n} , and \mathbf{d} . Since we can recover the translational motion and the surface parameters only up a scale factor, there are only 11 independent parameters to determine. At each point in the image we have one constraint equation and, therefore, 11 non-collinear points are usually sufficient to recover these parameters.

Again, we formulate a least-squares problem so that the solution is robust to noise in local regions, quantization, and the errors in the estimate of brightness derivatives. The problem is to find $\boldsymbol{\omega}$, \mathbf{t} , \mathbf{n} , and \mathbf{d} that minimize the sum of the square of the error in the brightness change equation over the whole region of the image,

$$J = \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d})(\mathbf{s} \cdot \mathbf{t}))^2 dx dy.$$

For an extremum of J we must have

$$\frac{\partial J}{\partial \boldsymbol{\omega}} = 0, \quad \frac{\partial J}{\partial \mathbf{t}} = 0, \quad \frac{\partial J}{\partial \mathbf{n}} = 0, \quad \text{and} \quad \frac{\partial J}{\partial \mathbf{d}} = 0.$$

Performing the above differentiations we obtain

$$\begin{aligned} \iint_I \mathbf{v}(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r} + \mathbf{d} \cdot \mathbf{q})(\mathbf{s} \cdot \mathbf{t})) dx dy &= 0, \\ \iint_I \mathbf{s}(\mathbf{n} \cdot \mathbf{r} + \mathbf{d} \cdot \mathbf{q})(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r} + \mathbf{d} \cdot \mathbf{q})(\mathbf{s} \cdot \mathbf{t})) dx dy &= 0, \\ \iint_I \mathbf{r}(\mathbf{s} \cdot \mathbf{t})(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r} + \mathbf{d} \cdot \mathbf{q})(\mathbf{s} \cdot \mathbf{t})) dx dy &= 0, \\ \iint_I \mathbf{q}(\mathbf{s} \cdot \mathbf{t})(c + \mathbf{v} \cdot \boldsymbol{\omega} + (\mathbf{n} \cdot \mathbf{r} + \mathbf{d} \cdot \mathbf{q})(\mathbf{s} \cdot \mathbf{t})) dx dy &= 0. \end{aligned}$$

These non-linear equations that we refer to as the *motion field equations for quadratic surfaces* can be solved iteratively. Unfortunately, we have not yet found an analytical solution that gives the optimum estimate in the least-squares sense (this can be obtained using the iterative method we describe next). A closed-form estimate may be obtained through a procedure described in the next section; however, the solution is generally non-robust, non-unique, and inferior to the solution obtained from the iterative method.

5.2 An Iterative Solution Procedure

We can exploit the fact that the first two equations are linear in ω , t and the last two equations are linear in \mathbf{n} and \mathbf{d} to formulate an iterative method similar to the one suggested for planar surfaces.

If we denote

$$\mathbf{d}_1 = \iint_I (c\mathbf{v}) \, dx \, dy, \quad \mathbf{d}_2 = \iint_I c(\mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d})\mathbf{s} \, dx \, dy,$$

$$\mathbf{M}_1 = \iint_I (\mathbf{v}\mathbf{v}^T) \, dx \, dy, \quad \mathbf{M}_2 = \iint_I (\mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d})(\mathbf{v}\mathbf{s}^T) \, dx \, dy, \quad \mathbf{M}_4 = \iint_I (\mathbf{r} \cdot \mathbf{n} + \mathbf{q} \cdot \mathbf{d})^2 (\mathbf{s}\mathbf{s}^T) \, dx \, dy,$$

$$\mathbf{e}_1 = \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})(\mathbf{s} \cdot \mathbf{t})\mathbf{r} \, dx \, dy, \quad \mathbf{e}_2 = \iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})(\mathbf{s} \cdot \mathbf{t})\mathbf{q} \, dx \, dy,$$

$$\mathbf{N}_1 = \iint_I (\mathbf{s} \cdot \mathbf{t})^2 (\mathbf{r}\mathbf{r}^T) \, dx \, dy, \quad \mathbf{N}_2 = \iint_I (\mathbf{s} \cdot \mathbf{t})^2 (\mathbf{r}\mathbf{q}^T) \, dx \, dy, \quad \mathbf{N}_4 = \iint_I (\mathbf{s} \cdot \mathbf{t})^2 (\mathbf{q}\mathbf{q}^T) \, dx \, dy,$$

then we can write

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ t \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{N}_1 & \mathbf{N}_2 \\ \mathbf{N}_2^T & \mathbf{N}_4 \end{pmatrix} \begin{pmatrix} \mathbf{n} \\ \mathbf{d} \end{pmatrix} = - \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}.$$

The solution to the above matrix equations can be expressed in terms of the inverse of 3×3 matrices \mathbf{M}_1 and \mathbf{N}_1 in the asymmetrical form

$$t = (\mathbf{M}_4 - \mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{M}_2)^{-1} (\mathbf{M}_2^T \mathbf{M}_1^{-1} \mathbf{d}_1 - \mathbf{d}_2),$$

$$\boldsymbol{\omega} = -\mathbf{M}_1^{-1} (\mathbf{d}_1 + \mathbf{M}_2 t),$$

$$\mathbf{d} = (\mathbf{N}_4 - \mathbf{N}_2^T \mathbf{N}_1^{-1} \mathbf{N}_2)^{-1} (\mathbf{N}_2^T \mathbf{N}_1^{-1} \mathbf{e}_1 - \mathbf{e}_2),$$

$$\mathbf{n} = -\mathbf{N}_1^{-1} (\mathbf{e}_1 + \mathbf{N}_2 \mathbf{d}),$$

or it can be obtained more easily by elimination. The iterative scheme then works as follows:

- (1) We start with an initial guess of the surface parameters, \mathbf{n} and \mathbf{d} .
- (2) We solve for $\boldsymbol{\omega}$ and t in terms of the current estimate of \mathbf{n} and \mathbf{d} , and then for \mathbf{n} and \mathbf{d} using the updated values of $\boldsymbol{\omega}$ and t .
- (3) Finally, we evaluate the improvement in the solution from the last iteration in order to stop if the solution has not changed or to go to step 2 if the algorithm has not converged yet.

Note that we need to solve only 11 equations to obtain the solution because of the scale-factor ambiguity (for example, we can set the length of the translational vector or one of the parameters in \mathbf{n} or \mathbf{t} equal to one without loss of generality). In our simulations, we have found it more convenient to solve all 12 equations and then rescale either the translational vector or n_3 after each update (each iteration).

5.3 Division of Labor

Fortunately, we do not have to perform the integrations for updating the coefficient matrices after each iteration. As was done for planar surfaces, we only have to accumulate some moments of the image brightness derivatives through an initial pass of the image; we do not have to refer back to the image after each iteration. Let us employ tensor notation in order to write

$$\begin{aligned} \{\mathbf{M}_1\}_{i,j} &= \iint_I v_i v_j dx dy, & \{\mathbf{M}_2\}_{i,j} &= \left[\iint_I v_i s_j r_k dx dy \right] n_k + \left[\iint_I v_i s_j q_k dx dy \right] d_k, \\ \{\mathbf{M}_4\}_{i,j} &= \left[\iint_I s_i s_j r_k r_l dx dy \right] n_k n_l + 2 \left[\iint_I s_i s_j r_k q_l dx dy \right] n_k d_l + \left[\iint_I s_i s_j q_k q_l dx dy \right] d_k d_l, \\ \{\mathbf{d}_1\}_i &= \iint_I c v_i dx dy, & \{\mathbf{d}_2\}_i &= \left[\iint_I c s_i r_j dx dy \right] n_j + \left[\iint_I c s_i q_j dx dy \right] d_j, \\ \{\mathbf{N}_1\}_{i,j} &= \left[\iint_I s_k s_l r_i r_j dx dy \right] t_k t_l, & \{\mathbf{N}_2\}_{i,j} &= \left[\iint_I s_k s_l r_i q_j dx dy \right] t_k t_l, \\ & & \{\mathbf{N}_4\}_{i,j} &= \left[\iint_I s_k s_l q_i q_j dx dy \right] t_k t_l, \\ \{\mathbf{e}_1\}_i &= \left[\iint_I c s_j r_i dx dy \right] t_j + \left[\iint_I v_k s_j r_i dx dy \right] w_k t_j, \\ \{\mathbf{e}_2\}_i &= \left[\iint_I c s_j q_i dx dy \right] t_j + \left[\iint_I v_k s_j q_i dx dy \right] w_k t_j. \end{aligned}$$

We only have to store the sums of $c v_i$ (3 numbers), $c r_i s_j$ (9 numbers), $c q_i s_j$ (9 numbers), $v_i v_j$ (6 numbers), $v_i s_j r_k$ (27 numbers), $v_i s_j q_k$ (27 numbers), $s_i s_j r_k r_l$ (36 numbers), $s_i s_j q_k q_l$ (36 numbers), and $s_i s_j r_k q_l$ (54 numbers) for a total of 207 numbers.

The updating of the matrices \mathbf{M}_i 's, $i = 2, \dots, 4$, and \mathbf{N}_i 's, $i = 1, \dots, 4$, and the vectors \mathbf{d}_2 , \mathbf{e}_1 , and \mathbf{e}_2 requires a total of about 420 multiplications per iteration (note that \mathbf{M}_1 and \mathbf{d}_1 are constants). The updating of ω and \mathbf{t} takes about 270 multiplications per iteration. The same number of multiplications are required to compute \mathbf{n} and \mathbf{d} in each iteration. Overall, it takes about 1000 multiplications per iteration to update the motion and surface parameters. We will show the convergence rate of the scheme through an example.

5.4 An Example

In this example, we will demonstrate the speed of convergence of the iterative scheme presented for quadratic surfaces starting from a random initial condition. The image plane is a square whose length is two units and is discretized to 32 image points in either direction. Again, we have used synthetic data so that the underlying motion is known perfectly. Further, in the presence of noise, we can determine the correlation between the (percentage of) noise and the (percentage) degradation from the correct solution. The synthetic data is generated through the procedure described in the previous chapter.

The tangent plane is given by $\mathbf{n} = [0.02, 0.02, 1]^T$. The surface is a saddle point characterized by the curvature parameter vector $\mathbf{d} = [0.5, 2.5, 0.5]^T$. The motion parameters are $\boldsymbol{\omega} = [0.05, 0.05, -0.08]^T$ and $\mathbf{t} = [0.1, -0.1, 0.05]^T$ (as in one of the examples given in the previous chapter). Figure 5.1 shows the results of several simulations for 100 iterations starting from random initial conditions. These results show that the motion parameters are recovered with good accuracy with as much as 57% noise in the data. Further, the algorithm converges to the solution in about 60 iterations. The same convergence rate has been obtained for other initial conditions for this particular example.

5.5 Summary

In this section, we presented an iterative method for recovering the motion parameters and the parameters of a quadratic surface from a set of 11 non-linear equations. This method is an extension of the first iterative method we presented for planar surfaces. Through an example, we demonstrated the convergence rate of the algorithm.

For most iterative methods, the convergence as well as the rate of convergence of the scheme will depend on the initial guess of the solution used to start the iterative process. This is a drawback of most iterative methods that makes them less attractive. Interestingly, the convergence rate of our iterative algorithm (in the cases we have tested) depends more on the type of surface and motion involved than on the initial condition. In general, the algorithm converges to the optimum solution in about 40-120 iterations. With noisy data, the algorithm sometimes converges to a local minimum depending on the initial condition. This occurs mostly in two situations; when the solution is nearly ambiguous (this will be addressed in the next chapter) or when the surface curvature is small. In the first case, the noise may perturb the data enough to make it resemble that corresponding to an ambiguous situation. In the second case, the algorithm may converge to a solution that is the dual of the best planar surface solution, the so-called "pseudo-dual" solution. This behavior is not surprising since, with noisy data, the surface can be mistaken to be planar. Then a second solution always exists unless the translational vector and the normal of the tangent plane are parallel.

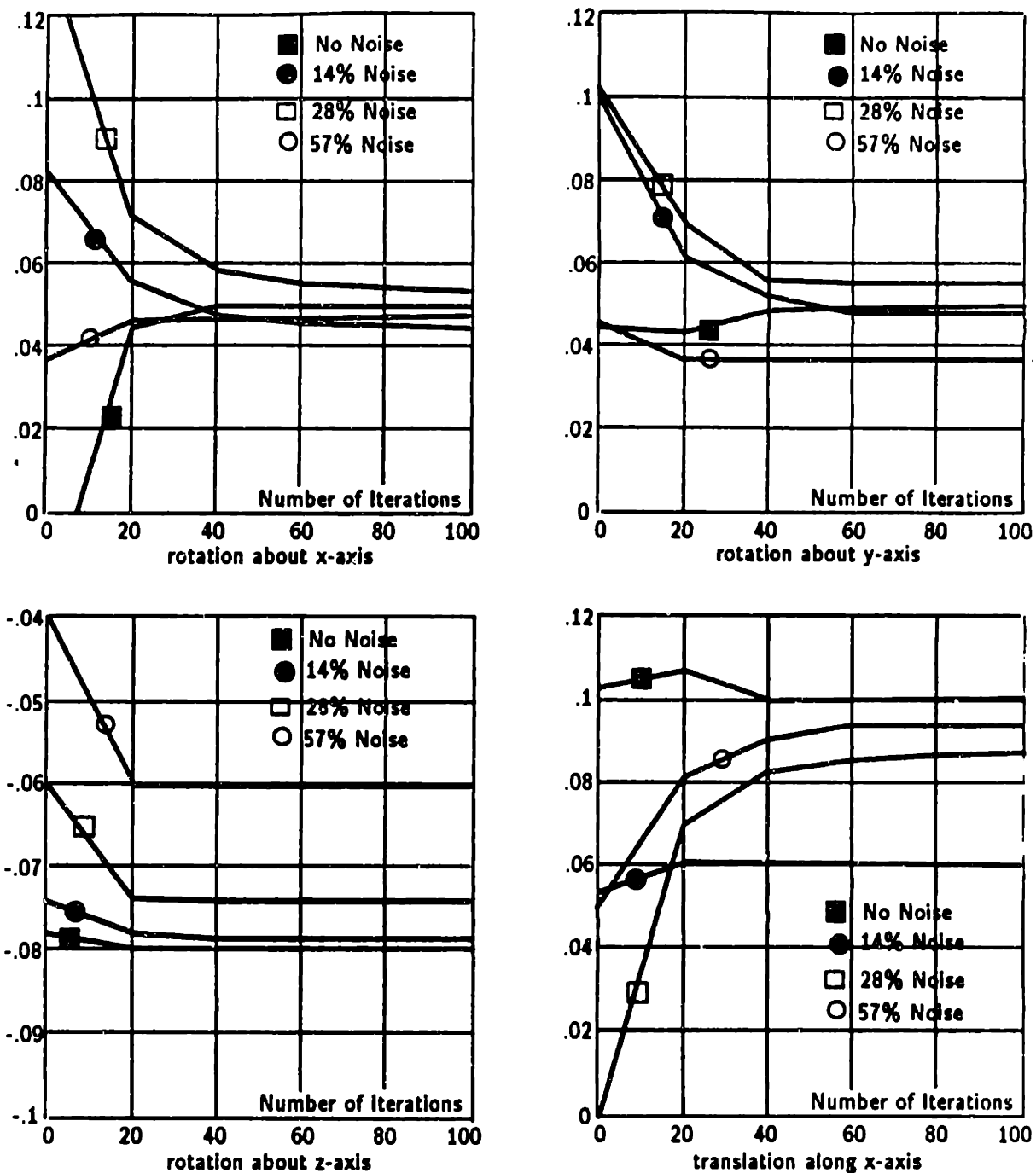


Figure 5-1. Results of simulations with noise-less and noisy data.

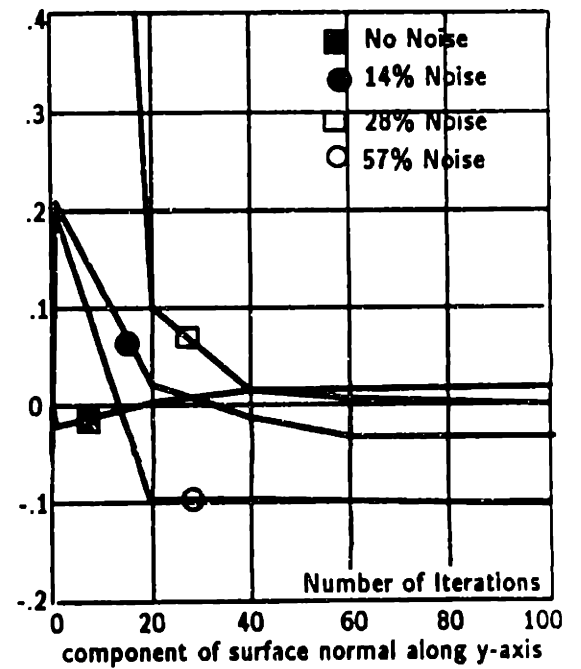
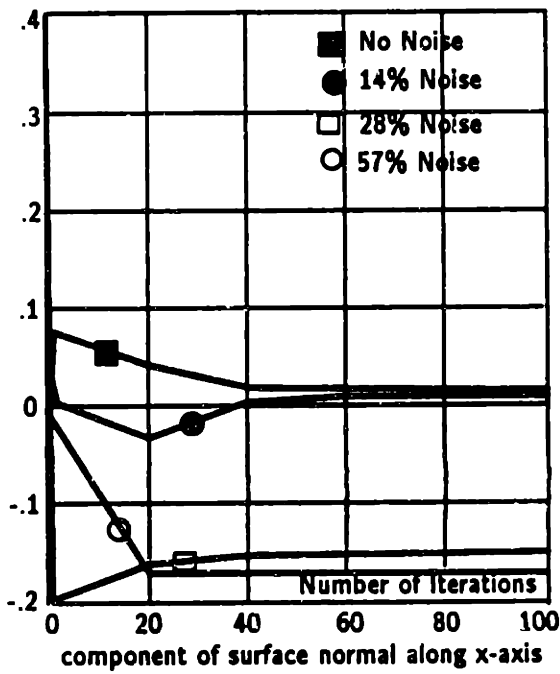
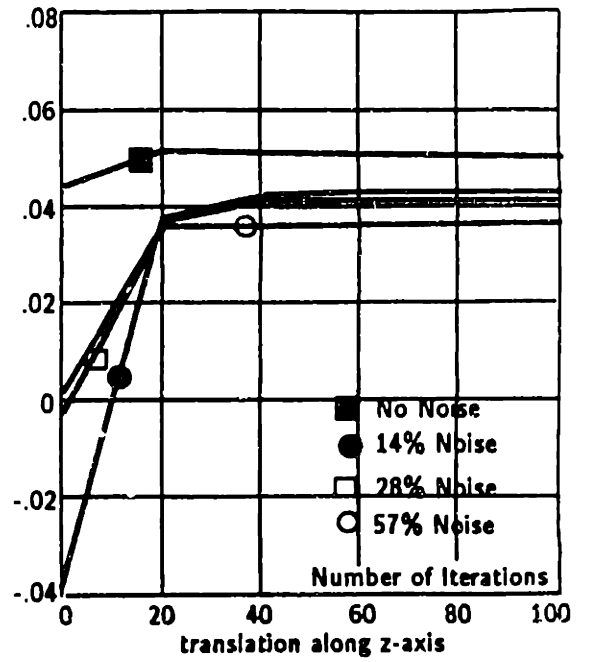
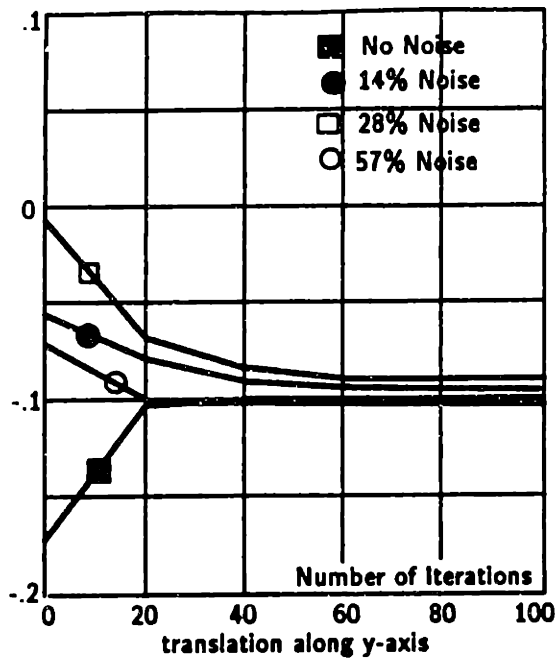


Figure 5-1. Continued.

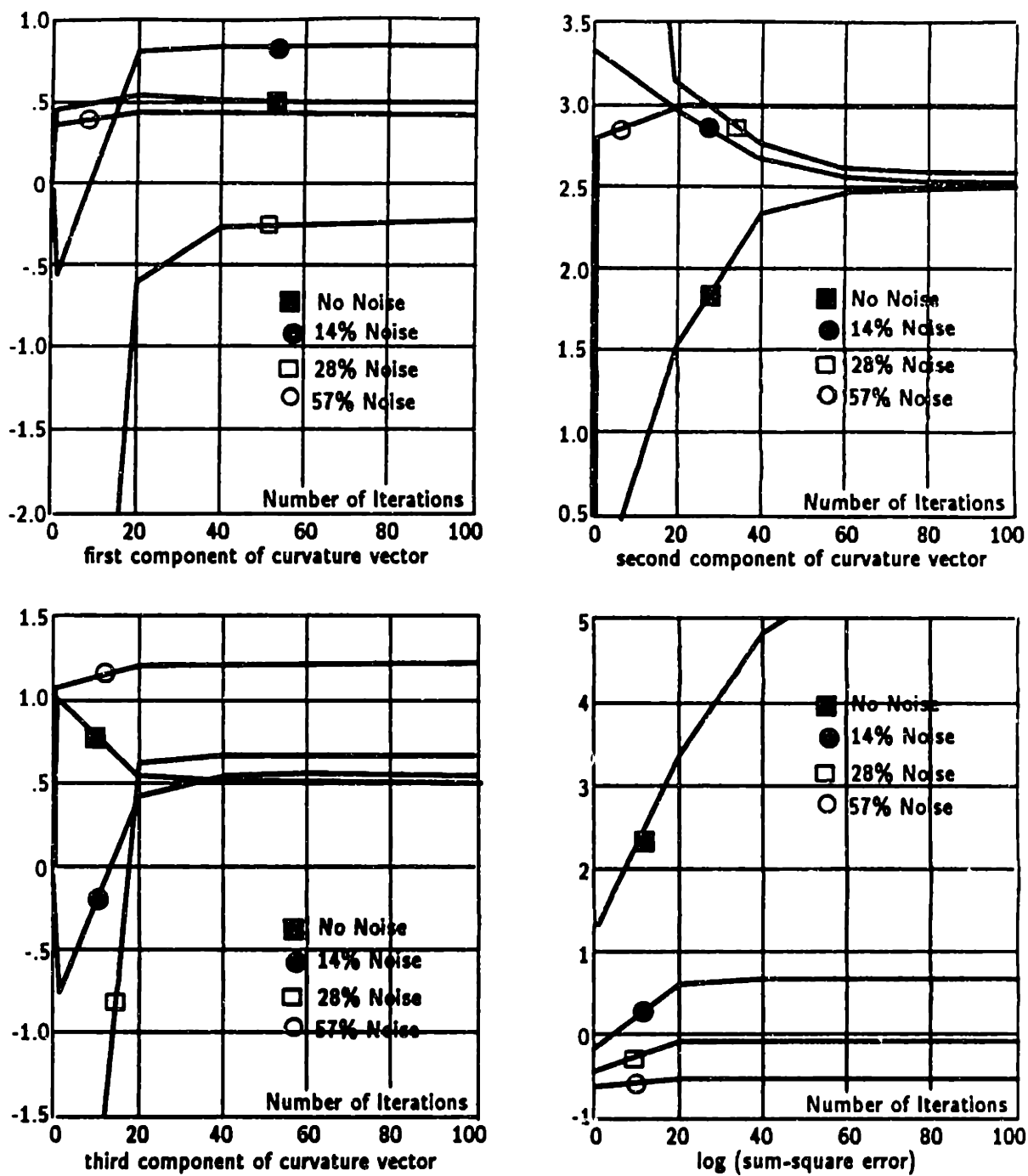


Figure 5-1. Continued.

6. Uniqueness of Motion

An important issue in recovering motion from time-varying imagery is whether or not it can be determined uniquely. In Chapter 4, the special case of planar surfaces was treated and it was shown that these surfaces give rise to a two-fold ambiguity unless the direction of motion coincides with that of the normal of the planar surface (similar results have been shown by several researchers including Hay [1966], Tsai & et al. [1982], Waxman & Ullman [1983], Longuet-Higgins [1984], Maybank [1984], Negahdaripour & Horn [1985a]). For smooth curved surfaces, two types of approaches, local and global, have been pursued. In the local approach, the surface is represented by a Taylor series expansion in a local neighborhood

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial X^2}\right)_0 X^2 + \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 XY + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial Y^2}\right)_0 Y^2 + O(\epsilon).$$

In the global approach, no special model is assumed for the surface of the scene; that is, the surface is assumed to be an arbitrary function $f(X, Y, Z) = 0$.

Using a local second order analysis, Longuet-Higgins & Prazdny [1980] show that three interpretations are possible for the motion parameters and the local structure of the surface of the scene (tangent plane orientation and surface curvature). Using correspondence of nine image points in two frames, Fang & Huang [1984] show that the motion parameters can be determined uniquely unless the points lie on a second-order surface passing through the viewing point. Tsai & Huang [1984] show that the correspondence of seven image points are sufficient unless the points lie on either two planes with one passing through the viewing point or a cone that passes through the viewing point. Waxman et al. [1986] show that only some special cases can give rise to the three-fold ambiguity that was observed by Longuet-Higgins & Prazdny. In addition, their results

indicate that two types of two-fold ambiguities may also occur under other conditions. Negahdaripour & Yuille [1986] show that only one special case can give rise to a three-fold ambiguity and another situation (not similar to the one observed by Waxman et al.) gives rise to a two-fold ambiguity (most of the ambiguities indicated by Waxman et al. are the shortcomings of a local second-order analysis of the motion field). Using a global analysis, Horn [1986b] has recently shown that the class of surfaces that may give rise to an ambiguity in the interpretation of the resulting motion field is restricted to planar surfaces and certain hyperboloids of one sheet that are viewed from a point on their surface. He also shows that the special cases observed by Tsai & Huang become degenerate cases of his results.

We will first extend the results derived by Negahdaripour & Yuille [1986] to present new uniqueness results. Here, we do not restrict ourselves to quadratic surfaces. We consider any smooth surface that can be represented by a Taylor expansion of order n with no restriction on n . We will show that, in most cases, the solution to the problem of recovering 3-D motion from time-varying images is unique up to the scale-factor ambiguity. Exceptions are, as shown earlier, planar surfaces that give rise to a two-fold ambiguity unless the direction of translation and the surface normal coincide and certain quadratic surfaces. These surfaces may give rise to a two-fold or a three-fold ambiguity under very special circumstances. We then present the relationship between our results and those due to Horn [1986b]. This leads us to the derivation of the relationship among the multiple solutions, when they exist.

We show in several examples that the method used to derive the uniqueness results is non-robust and, therefore, cannot be used to recover 3-D motion or structure analytically (this is explained more formally in Appendix A). The alternative is a scheme that solves a set of non-linear equations, obtained from a least-squares formulation, iteratively (for example, the iterative scheme presented in Chapter 5). Iterative methods may converge to a solution that is optimum only locally. At the expense of more computation, it is usually possible to implement an algorithm that eventually converges to the global optimum solution. The difficulty, however, arises in the special cases where two or three global minima exist since each solution is an acceptable one. In general, the algorithm may converge to any of the solutions depending on the initial condition. Knowledge of the number of possible solutions and the relationships among them will allow us to determine all of the possible solutions once any one of them has been obtained. Therefore, we do not have to depend on non-robust analytical methods in order to compute all of the possible solutions in one step. These results are significant not only in motion vision applications but in any other application where the relative position and orientation of two camera stations have to be determined from images of the scene recorded by the two cameras; for example, as in photogrammetry, camera calibration or stereo analysis.

In the derivation of the uniqueness results, a strategy similar to the one used for the case of planar surfaces is employed. We first compute some intermediate parameters,

similar to the essential parameters of planar surfaces, from a least-squares formulation. We then exploit the special structure of the equations for the essential parameters in order to determine the motion and surface parameters analytically.

With noisy data, the essential parameters we need to compute may not be accurate. Furthermore, the method given for the computation of the motion and surface parameters from the essential parameters is generally non-robust (this is addressed in Appendix A where we study the non-robustness of certain closed-form solutions more formally). Therefore, our goal is not to present an analytical solution for recovering motion but to show when the solution can be determined uniquely. The question we address is as follows: Given two consecutive images of a scene with a smooth surface, when can the motion parameters be determined uniquely? Alternatively, we need to know what circumstances can give rise to an ambiguity.

We will take advantage of tensor notation to keep the results as compact as possible. We denote a two-dimensional vector \mathbf{x} by x_i , a 2×2 matrix Ω by Ω_{ij} , and a three-dimensional vector \mathbf{t} by $\{t_i, t_3\}$, where t_i is the component parallel to the image plane (a two-dimensional vector) and t_3 is the component in the direction of the optical axis (a scalar).

6.1 Surface Representation

A surface can be represented in several ways. For example, if the surface is smooth, it can be approximated by a polynomial of order n in terms of the object coordinates

$$Z = Z_0 + \left(\frac{\partial Z}{\partial X}\right)_0 X + \left(\frac{\partial Z}{\partial Y}\right)_0 Y + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial X^2}\right)_0 X^2 + \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_0 XY + \frac{1}{2} \left(\frac{\partial^2 Z}{\partial Y^2}\right)_0 Y^2 + \dots \\ + \frac{1}{n!} \left(\frac{\partial^n Z}{\partial X^n}\right)_0 X^n + \dots + \frac{1}{n!} \left(\frac{\partial^n Z}{\partial Y^n}\right)_0 Y^n.$$

This representation can be viewed as the Taylor series expansion of the surface, up to some order n , in a neighborhood of the fixation point. Alternatively, the surface may be represented in a neighborhood of the image plane by a Taylor expansion in terms of the image coordinates. This is more convenient since we deal with images. Here, we write the expansion of $d = 1/Z$ since the brightness change constraint equation depends on $1/Z$ (as a result of perspective projection). Therefore, the surface is given by

$$d = d_0 + \left(\frac{\partial d}{\partial x}\right)_0 x + \left(\frac{\partial d}{\partial y}\right)_0 y + \frac{1}{2} \left(\frac{\partial^2 d}{\partial x^2}\right)_0 x^2 + \left(\frac{\partial^2 d}{\partial x \partial y}\right)_0 xy + \frac{1}{2} \left(\frac{\partial^2 d}{\partial y^2}\right)_0 y^2 + \dots \\ + \frac{1}{n!} \left(\frac{\partial^n d}{\partial x^n}\right)_0 x^n + \dots + \frac{1}{n!} \left(\frac{\partial^n d}{\partial y^n}\right)_0 y^n.$$

Using tensor notation this can be written

$$d = d_0 + d_i r_i + \frac{1}{2} d_{ij} r_i r_j + \dots + \frac{1}{n!} d_{i_1 \dots i_n} r_{i_1} \dots r_{i_n},$$

where $r_i = (x, y)$ and

$$d_i = \left(\frac{\partial d}{\partial r_i}\right)_0, \quad d_{ij} = \left(\frac{\partial^2 d}{\partial r_i \partial r_j}\right)_0, \quad \dots \quad d_{i_1 \dots i_n} = \left(\frac{\partial^n d}{\partial r_{i_1} \dots \partial r_{i_n}}\right)_0.$$

Since the depth values can be recovered up to a scale factor, we will set $d_0 = 1$ without loss of generality. This implies that the point of fixation is located at the origin of the image plane which is positioned at a unit distance from the viewer. Therefore, we write

$$d = 1 + d_i r_i + \frac{1}{2} d_{ij} r_i r_j + \dots + \frac{1}{n!} d_{i_1 \dots i_n} r_{i_1} \dots r_{i_n}$$

In this derivation, we will assume that the above representation is a good approximation of the surface of the scene. We use the term “quadratic surfaces” whenever a second-order Taylor expansion is used to represent the surface. Similarly, the term “cubic surfaces” will be used for surfaces that are presented by a third-order Taylor expansion. It should be noted that these are only subsets of the class of surfaces commonly labeled as quadratic or cubic. This can be best explained by noting that a general quadric in 3-D space is represented by nine independent parameters; there is a constant term, three linear terms, and six quadratic terms that can be chosen arbitrarily up to a scale factor. Our representation of a quadratic surface, in comparison, allows for only six degrees of freedom (the same reasoning applies to cubic surfaces). More will be said about this later when we present the results on the relationship among multiple solutions.

6.2 Essential Parameters of an n^{th} Order Surface

Using the new notation $\mathbf{s} = \{s_i, s_3\}$, $\mathbf{v} = \{v_i, v_3\}$, $\boldsymbol{\omega} = \{\omega_i, \omega_3\}$, and $\mathbf{t} = \{t_i, t_3\}$, and substituting the equation for $d = 1/Z$ into the brightness change constraint equation,

$$c + \mathbf{v} \cdot \boldsymbol{\omega} + d \mathbf{s} \cdot \mathbf{t} = 0,$$

we arrive at

$$c + (v_i \omega_i + v_3 \omega_3) + \left(1 + d_i r_i + \frac{1}{2} d_{ij} r_i r_j + \dots + \frac{1}{n!} d_{i_1 \dots i_n} r_{i_1} \dots r_{i_n}\right) (s_i t_i + s_3 t_3) = 0,$$

where

$$c = E_t, \quad s_i = -E_{r_i}, \quad s_3 = r_i E_{r_i}, \quad v_i = (\delta_{jk} + r_j r_k) \epsilon_{ik} E_{r_j}, \quad \text{and} \quad v_3 = -\epsilon_{ij} r_i E_{r_j}.$$

E_t and $E_{r_i} = (\partial E / \partial x, \partial E / \partial y)$ denote the temporal and spatial derivative of the image function $E(r_i, t)$. Also, δ_{ij} and ϵ_{ij} denote the Kronecker delta symbol and the anti-symmetric permutation tensor, respectively. These are

$$\delta_{ij} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \epsilon_{ij} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

The brightness change constraint equation can be written as

$$\begin{aligned}
& (\epsilon_{ij}w_i - t_j)E_{r_j} + (\delta_{ij}t_3 - \epsilon_{ij}w_3 - d_i t_j)r_i E_{r_j} + \\
& \frac{1}{2!}((\epsilon_{li}w_l + t_3 d_i)\delta_{jk} + (\epsilon_{lj}w_l + t_3 d_j)\delta_{ik} - d_{ij}t_k)r_i r_j E_{r_k} + \\
& \frac{1}{3!}((d_{ij}\delta_{kl} + d_{ik}\delta_{jl} + d_{jk}\delta_{il})t_3 - d_{ijk}t_l)r_i r_j r_k E_{r_l} + \dots + \\
& \frac{1}{n!}((d_{i_1 \dots i_{n-1}}\delta_{i_n i_{n+1}} + d_{i_1 \dots i_{n-2} i_n}\delta_{i_{n-1} i_{n+1}} + \dots + d_{i_2 \dots i_n}\delta_{i_1 i_{n+1}})t_3 - d_{i_1 \dots i_n}t_{i_{n+1}})r_{i_1} \dots r_{i_n} E_{r_{i_{n+1}}} + \\
& \frac{1}{n!}(d_{i_1 \dots i_n}t_3)r_{i_1} \dots r_{i_{n+1}} E_{r_{i_{n+1}}} = -c.
\end{aligned}$$

This is a non-linear constraint equation in terms of the motion and surface parameters; however, if we define

$$\begin{aligned}
m_i &= \epsilon_{ji}w_j - t_i, \\
m_{ij} &= \delta_{ij}t_3 - \epsilon_{ij}w_3 - d_i t_j, \\
m_{ijk} &= (\epsilon_{li}w_l + t_3 d_i)\delta_{jk} + (\epsilon_{lj}w_l + t_3 d_j)\delta_{ik} - d_{ij}t_k, \\
m_{ijkl} &= (d_{ij}\delta_{kl} + d_{ik}\delta_{jl} + d_{jk}\delta_{il})t_3 - d_{ijk}t_l, \\
&\vdots = \quad \quad \quad \vdots \\
m_{i_1 \dots i_{n+1}} &= (d_{i_1 \dots i_{n-1}}\delta_{i_n i_{n+1}} + d_{i_1 \dots i_{n-2} i_n}\delta_{i_{n-1} i_{n+1}} + \dots + d_{i_2 \dots i_n}\delta_{i_1 i_{n+1}})t_3 - d_{i_1 \dots i_n}t_{i_{n+1}}, \\
\tilde{m}_{i_1 \dots i_n} &= d_{i_1 \dots i_n}t_3,
\end{aligned}$$

as the *essential parameters* of an n^{th} order surface (the same term we used for similar intermediate parameters in the case of planes), then we can express the above equation as a linear constraint equation in terms of the essential parameters,

$$\begin{aligned}
E_{r_i} m_i + r_i E_{r_j} m_{ij} + \frac{1}{2!} r_i r_j E_{r_k} m_{ijk} + \dots + \\
\frac{1}{n!} r_{i_1} \dots r_{i_n} E_{r_{i_{n+1}}} m_{i_1 \dots i_{n+1}} + \frac{1}{n!} r_{i_1} \dots r_{i_{n+1}} E_{r_{i_{n+1}}} \tilde{m}_{i_1 \dots i_n} = -c.
\end{aligned}$$

Using the above equation, we can determine the essential parameters from the image brightness derivatives. Note that d_{ij} , d_{ijk} , ..., and $d_{i_1 \dots i_n}$ are symmetric with respect to every index (i.e., $d_{ij} = d_{ji}$, $d_{ijk} = d_{ikj} = d_{jik} = d_{jki} = d_{kij} = d_{kji}$, etc). The same symmetry is preserved in the definition of m_{ijk} , m_{ijkl} , etc; m_{ijk} is symmetric with respect to i and j , m_{ijkl} is symmetric with respect to i , j and k , etc. Therefore, there are only six distinct elements in m_{ijk} , eight in m_{ijkl} , ..., and $2(n+1)$ in $m_{i_1 \dots i_{n+1}}$. Also, there are only $(n+1)$ distinct elements in $\tilde{m}_{i_1 \dots i_n}$ (the same number as in $d_{i_1 \dots i_n}$). This gives $2+4+6+\dots+2(n+1) = (n+1)(n+2)$ essential parameters in m_i , m_{ij} , m_{ijk} , ..., $m_{i_1 \dots i_{n+1}}$, and $(n+1)$ parameters in $\tilde{m}_{i_1 \dots i_n}$. All together, we have $(n+1)(n+3)$ essential parameters to solve for; we need at least as many data points.

6.3 Computing the Essential Parameters

If we define

$$\mathbf{m} = [m_i, m_{ij}, m_{ijk}, \dots, m_{i_1 \dots i_{n+1}}, \tilde{m}_{i_1 \dots i_n}]^T$$

to be the vector of the $(n+1)(n+3)$ distinct essential parameters and

$$\mathbf{e} = [E_{r_i}, r_i E_{r_j}, \frac{1}{2!} r_i r_j E_{r_k}, \dots, \frac{1}{n!} r_{i_1} \dots r_{i_n} E_{r_{i_{n+1}}}, \frac{1}{n!} r_{i_1} \dots r_{i_{n+1}} E_{r_{i_{n+1}}}]^T$$

to be the vector of moments of image brightness derivatives, then the brightness change equation can be written

$$c + \mathbf{e} \cdot \mathbf{m} = 0.$$

For each point in the relevant image region, we have one constraint equation. We can formulate a least-squares problem to estimate the elements of \mathbf{m} from the information in the whole region of interest. More precisely, we choose the vector \mathbf{m} that minimizes

$$\iint_I (c + \mathbf{e} \cdot \mathbf{m})^2 dx dy.$$

The least-squares solution is given by

$$\mathbf{m} = -\left(\iint_I \mathbf{e} \mathbf{e}^T dx dy\right)^{-1} \left(\iint_I c \mathbf{e} dx dy\right).$$

Mathematically, we can obtain the essential parameters through this formulation. In practice, the procedure does not work in the presence of noise in the images and, therefore, it is not for implementation. To explain this, we need to remember that the actual unknown parameters are the elements of $\{w_i, w_3\}$ (three unknowns), $\{t_i, t_3\}$ (three unknowns), d_i (two unknowns), d_{ij} (three unknowns), d_{ijk} (four unknowns), ..., and $d_{i_1 \dots i_n}$ ($n+1$ unknowns); there is a total of $2 + 3 + \dots + (n+1) + 6 = n(n+3)/2 + 6$ independent degrees of freedom in the $(n+1)(n+3)$ essential parameters. As we showed in Chapter 4, when the surface is planar ($n=1$) we have eight essential parameters with eight degrees of freedom (eight unknowns). However, when $n > 1$, there are always more essential parameters and, therefore, more constraints than the number of independent unknowns; that is, $(n+1)(n+3) > 6 + n(n+3)/2$. When the data are perfect, the essential parameters obtained from the least-squares formulation are equal to their true values and so they satisfy every constraint. With noisy data, the $(n+1)(n+3)$ essential parameters obtained from the least-squares formulation will generally be independent. In other words, they will violate the constraint that they should consist of only $6 + n(n+3)/2$ independent degrees of freedom in the appropriate combinations. This is explained in more detail in appendix A where the robustness issue of closed-form methods is addressed.

6.4 Plan of Action

The essential parameters are defined in terms of some non-linear functions of the motion and surface parameters,

$$\begin{aligned}
 m_i &= \epsilon_{ji} w_j - t_i, \\
 m_{ij} &= \delta_{ij} t_3 - \epsilon_{ij} w_3 - d_i t_j, \\
 m_{ijk} &= (\epsilon_{li} w_l + t_3 d_i) \delta_{jk} + (\epsilon_{lj} w_l + t_3 d_j) \delta_{ik} - t_k d_{ij}, \\
 m_{ijkl} &= (d_{ij} \delta_{kl} + d_{ik} \delta_{jl} + d_{jk} \delta_{il}) t_3 - d_{ijk} t_l, \\
 &\vdots = \vdots \\
 m_{i_1 \dots i_{n+1}} &= (d_{i_1 \dots i_{n-1}} \delta_{i_n i_{n+1}} + d_{i_1 \dots i_{n-2} i_n} \delta_{i_{n-1} i_{n+1}} + \dots + d_{i_2 \dots i_n} \delta_{i_1 i_{n+1}}) t_3 - d_{i_1 \dots i_n} t_{i_{n+1}}, \\
 \tilde{m}_{i_1 \dots i_n} &= d_{i_1 \dots i_n} t_3.
 \end{aligned}$$

We explained that there are only $6 + n(n+3)/2$ degrees of freedom in the above $(n+1)(n+3)$ equations in terms of the essential parameters. Therefore, we may consider another least-squares formulation to find the “optimal” parameters from these non-linear equations; however, this defeats our purpose which is to exploit the special structure of these equations to show uniqueness results.

The first three tensor equations ($2 + 4 + 6 = 12$ scalar equations),

$$\begin{aligned}
 m_i &= \epsilon_{ji} w_j - t_i, \\
 m_{ij} &= \delta_{ij} t_3 - \epsilon_{ij} w_3 - d_i t_j, \\
 m_{ijk} &= (\epsilon_{lj} w_l + t_3 d_j) \delta_{ik} + (\epsilon_{li} w_l + t_3 d_i) \delta_{jk} - t_k d_{ij},
 \end{aligned}$$

are the coefficients of the Taylor series expansion of the motion field up to the second-order terms and can be used to recover motion and the local surface gradient and curvature. This idea was first explored by Longuet-Higgins & Prazdny [1980] (we will refer to their method by “L&P”). To solve these equations, they used an image coordinate transformation that reduced the problem to that of solving a cubic equation. Since they did not attempt to distinguish between the solutions, it was concluded that three solutions are usually possible. Waxman & Ullman [1983] (we will refer to their method by “W&U”) presented a new formulation of L&P in terms of 12 deformation parameters (rate of strain, rotation, etc) associated with a local region of the image (these parameters are linear combinations of the motion field and its first and second derivatives at a single point that was used by L&P). Using the idea of L&P, they solved these equations in a transformed image coordinate system. The angle of transformation, however, was determined from a non-linear equation which was solved iteratively. They illustrated uniqueness results through selected examples. Waxman et al. [1986] derived a closed-form solution to these equations using the image coordinate transformation suggested by L&P. The following conclusions were reached:

- (1) Three solutions are obtained when $d_i = 0$, the surface has a negative Gaussian curvature ($d_{11}d_{22} - d_{12}^2 < 0$), and the mean scaled curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$).
- (2) Two solutions are obtained, one for each possible transformation angle, when the surface has a non-positive Gaussian curvature ($d_{11}d_{22} - d_{12}^2 < 0$), the mean scaled curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and the translation along the line of sight vanishes ($t_3 = 0$). When $d_1/d_2 = t_1/t_2$, the two solutions become equivalent.
- (3) Two solutions are obtained regardless of the sign of the Gaussian curvature when $d_1/d_2 = t_1/t_2$ (this condition was referred to as the *structure-motion coincidence*). The two solutions are characterized by the same transformation angle and degenerate to a unique solution when the translation is along the surface normal or when $t_3 = 0$.
- (4) In all other cases, the solution is unique.

In the case of locally quadratic surfaces, Negahdaripour & Yuille [1986] have shown that the ambiguity in (1) arises *if and only if* there is no translation along the line of sight (the three-fold ambiguity when $t_3 \neq 0$ is a shortcomings of a local second-order analysis). Also, the ambiguity in (2) occurs *if and only if* d_1/d_2 is equal to a spurious solution of the transformation angle; otherwise, the solution is unique. Finally, the two-fold ambiguity in (3) is also a shortcoming of a second-order analysis of the motion field.

We will show that, for curved surfaces, an ambiguity in the interpretation of the motion field may arise only in the case of quadratic surfaces that are viewed by an observer moving parallel to the image plane and it is related to the curvature properties of the surface at the fixation point. In an ambiguous case, the resulting motion field is a second-order flow field; this obviously does not imply that any second-order motion field is ambiguous. Further, no other curved surface can give rise to a potentially ambiguous flow field. The ambiguity in the case of quadratic surfaces that are viewed by an observer moving parallel to the surface ($t_3 = 0$) is either

- (1) three-fold when the surface gradient vanishes ($d_i = 0$), the Gaussian curvature is negative ($d_{11}d_{12} - d_{12}^2 < 0$), and the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$) or
- (2) two-fold when the Gaussian curvature is negative ($d_{11}d_{22} - d_{12}^2 < 0$), the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and the surface normal, the optical axis, and one of the asymptotic lines are in the same plane, that is,

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

When $t_1/t_2 = d_1/d_2$, the two-fold ambiguity is resolved because the two solutions become identical.

Under certain conditions related to the size of the field of view, these ambiguities can be resolved by imposing the constraint that the depth values for an acceptable surface

should be positive (that is, the object is in front of the camera). In other words, a spurious solution may give rise to negative depth values in certain regions of the image. In chapter 7, we will show that this constraint can be exploited to recover 3-D motion from brightness gradient vectors.

6.5 Recovering Motion and Surface Parameters

In general, the motion parameters can be determined uniquely from the 12 equations for m_i , m_{ij} , and m_{ijk} . We will first determine when these equations have multiple solutions. We then show how the equations for m_{ijkl} , ..., $m_{i_1 \dots i_{n+1}}$ and $\tilde{m}_{i_1 \dots i_n}$ are used to select the correct solution.

In order to simplify the derivation, we first eliminate the rotational motion parameters from m_{ijk} using the equations for m_i . The new constraints are given by

$$\begin{aligned} m_i &= \epsilon_{ij} w_j - t_i, \\ m_{ij} &= \delta_{ij} t_3 - \epsilon_{ij} w_3 - d_i t_j, \\ m_{ijk} &= (t_j + t_3 d_j) \delta_{ik} + (t_i + t_3 d_i) \delta_{jk} - t_k d_{ij}, \end{aligned}$$

where

$$(m_{ijk})_{\text{new}} = (m_{ijk})_{\text{old}} - m_i \delta_{jk} - m_j \delta_{ik}.$$

The essential parameters, in component form, are written

$$\begin{array}{ll} m_1 = -w_2 - t_1 & m_2 = w_1 - t_2 \\ m_{11} = t_3 - d_1 t_1 & m_{22} = t_3 - d_2 t_2 \\ m_{21} = w_3 - d_2 t_1 & m_{12} = -w_3 - d_1 t_2 \\ m_{111} = 2(t_1 + d_1 t_3) - t_1 d_{11} & m_{222} = 2(t_2 + d_2 t_3) - t_2 d_{22} \\ m_{121} = t_2 + d_2 t_3 - t_1 d_{12} & m_{212} = t_1 + d_1 t_3 - t_2 d_{12} \\ m_{221} = -t_1 d_{22} & m_{112} = -t_2 d_{11} \end{array}$$

The above equations may be solved directly for the motion and surface parameters in several ways. Each method eventually reduces to solving a cubic equation in terms of t_1/t_2 (or t_2/t_1) similar to the one derived by L&P; however, they did not attempt to address the uniqueness issue. In order to deal with every special case systematically and obtain more insight into the nature of the ambiguities, we present a slightly different procedure. This procedure involves a coordinate transformation in order to simplify the 12 non-linear equations (this idea was first exploited by L&P and later by W&U).

6.5.1 Coordinate Transformation

An image coordinate transformation can be written in the form

$$x'_i = \Omega_{ij} x_j = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} x_j,$$

where the superscript ($'$) is used to denote the coordinates after transformation. The new coordinate system is chosen such that the x'_2 -axis is aligned with the component of the translation in the image plane, t_i ; then we have $t'_1 = 0$ (note that the transformation angle θ is not known a priori since t_i is unknown). Alternatively, it is also possible to align x'_1 -axis with the component of translational vector in the image plane due to the symmetry in the equations for the essential parameters and, hence, obtain $t'_2 = 0$. We consider the first alternative here.

Let m'_i , m'_{ij} , and m'_{ijk} denote the essential parameters in the new coordinate system. Then we can write

$$\begin{aligned} m'_i &= \Omega_{il} m_l = \Omega_{il} (\epsilon_{jl} w_j - t_l), \\ m'_{ij} &= \Omega_{il} \Omega_{jm} m_{lm} = \Omega_{il} \Omega_{jm} (\delta_{lm} t_3 - \epsilon_{lm} w_3 - d_l t_m), \\ m'_{ijk} &= \Omega_{il} \Omega_{jm} \Omega_{kn} m_{lmn} = \Omega_{il} \Omega_{jm} \Omega_{kn} \left((t_m + d_m t_3) \delta_{nl} + (t_l + d_l t_3) \delta_{nm} - t_n d_{lm} \right). \end{aligned}$$

These can be written in the form

$$\begin{aligned} m'_i &= \epsilon_{ji} w'_j - t'_i, \\ m'_{ij} &= \delta_{ij} t'_3 - \epsilon_{ij} w'_3 - d'_i t'_j, \\ m'_{ijk} &= (t'_j + d'_j t'_3) \delta_{ki} + (t'_i + d'_i t'_3) \delta_{kj} - t'_k d'_{ij}, \end{aligned}$$

where

$$w'_i = \Omega_{il} w_l, \quad t'_i = \Omega_{il} t_l, \quad d'_i = \Omega_{il} d_l, \quad \text{and} \quad d'_{ij} = \Omega_{il} \Omega_{jm} d_{lm}.$$

Since the transformation involves a rotation about the optical axis (z -axis), it is clear that $t_3 = t'_3$ and $w_3 = w'_3$. Writing the equations in the new coordinates frame and setting $t'_1 = 0$, we obtain

$$\begin{array}{ll} m'_1 = -w'_2 & m'_2 = w'_1 - t'_2 \\ m'_{11} = t'_3 & m'_{22} = t'_3 - d'_2 t'_2 \\ m'_{21} = w'_3 & m'_{12} = -w'_3 - d'_1 t'_2 \\ m'_{111} = 2d'_1 t'_3 & m'_{222} = 2(t'_2 + d'_2 t'_3) - t'_2 d'_{22} \\ m'_{121} = t'_2 + d'_2 t'_3 & m'_{212} = d'_1 t'_3 - t'_2 d'_{12} \\ m'_{221} = 0 & m'_{112} = -t'_2 d'_{11} \end{array}$$

We can use the constraint equation $m'_{221} = 0$ to determine the transformation angle θ . If we substitute for the transformation matrix Ω_{ij} into the equation

$$m'_{221} = \Omega_{2l} \Omega_{2m} \Omega_{1n} m_{lmn} = 0,$$

we obtain

$$-m_{112} \tan^3 \theta + (m_{111} - 2m_{122}) \tan^2 \theta - (m_{222} - 2m_{121}) \tan \theta + m_{221} = 0.$$

This is similar to the cubic equation L&H arrived at in their derivation. In our case, it can be used to determine the transformation angle up to a possible three-fold ambiguity. Of course, only one solution yields the correct transformation. In order to obtain more insight into the nature of the spurious solutions, we need to write this equation in terms of the motion and surface parameters; that is,

$$t_2 d_{11} \tan^3 \theta + (2t_2 d_{12} - t_1 d_{11}) \tan^2 \theta + (t_2 d_{22} - 2t_1 d_{12}) \tan \theta - t_1 d_{22} = 0,$$

which can be factored into the following form

$$(t_2 \tan \theta - t_1)(d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22}) = 0.$$

The solutions for the transformation angle are

$$\tan \theta = \frac{t_1}{t_2}, \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

The first solution gives the correct transformation angle; that is, the transformation that aligns the x'_2 -axis with t_i . The spurious solutions exist *if and only if* the other two roots are real. In Appendix B, we show that the transformation angle is non-unique for surfaces with a negative Gaussian curvature; in this case, there is a three-fold ambiguity corresponding to the true solution and two spurious solutions. In the case of surfaces with zero Gaussian curvature where a principal curvature vanishes, the ambiguity reduces to two-fold since the two spurious solutions become identical; because of this, we will, in the remainder of this chapter, treat surfaces with zero Gaussian curvature as a degenerate case of surfaces with a negative Gaussian curvature. Finally, the transformation angle is unique for surfaces with a positive Gaussian curvature since the two spurious solutions become complex.

The spurious solutions of the cubic equation are related to the asymptotic directions of the surface. The two spurious solutions for surfaces with a negative Gaussian curvature correspond to the two possible rotations that align the x'_2 -axis with the two asymptotic directions. For surfaces with a zero Gaussian curvature, the only spurious solution results in a rotation that aligns the x'_2 -axis with the only asymptotic direction of the surface. Finally, surfaces with a positive Gaussian curvature have no asymptotes. This is consistent with the fact that the cubic equation for the transformation angle has only one real root corresponding to the correct rotation that aligns the x'_2 -axis with t_i . To summarize, the cubic equation in terms of the sought-after transformation angle can have as many as three solutions. The number of spurious solutions depends on the type of surface viewed. More precisely, it depends on the number of the asymptotic directions of the surface at the fixation point.

The two asymptotic directions of the surface, when these exist, and the direction of the component of the translational motion parallel to the image plane constitute what

we will refer to as the *characteristic lines of motion-structure ambiguity*. We will see that these have important implications on the relationship among multiple solutions.

The coefficients of the cubic equation in terms of the transformation angle vanish *if and only if* either $t_i = 0$ or $d_{ij} = 0$. In both cases, the curvature parameters d_{ij} drop out of the equations and cannot be computed from the 12 equations. The method given for planar surfaces, however, can be used to recover the motion parameters, $\{t_i, t_3\}$ and $\{w_i, w_3\}$, and the parameters of the normal vector of the tangent plane, d_i . Of course, there is a two-fold ambiguity in the solution unless $\{t_i, t_3\} = t_3\{d_i, 1\}$; that is, the translation is along the normal to the tangent plane of the surface at the origin.

We consider two separate cases depending on whether the coefficients of the cubic equation do or do not vanish. If the coefficients of the cubic equation,

$$-m_{112} \tan^3 \theta + (m_{111} - 2m_{122}) \tan^2 \theta - (m_{222} - 2m_{121}) \tan \theta + m_{221} = 0,$$

vanish, then either $d_{ij} = 0$ or $t_i = 0$. Conversely, when the coefficients of the cubic do not vanish, we conclude that $t_i \neq 0$ and $d_{ij} \neq 0$.

6.5.2 The Case of $t_i \neq 0$ and $d_{ij} \neq 0$

As explained earlier, the cubic equation

$$-m_{112} \tan^3 \theta + (m_{111} - 2m_{122}) \tan^2 \theta - (m_{222} - 2m_{121}) \tan \theta + m_{221} = 0,$$

has three roots unless the surface has a positive Gaussian curvature (two are the same if the Gaussian curvature is zero). For each solution of θ , we compute the essential parameters in the rotated coordinate frame.

We can determine the solution for t'_3 from $t'_3 = m'_{11}$. When $t'_3 = 0$, the uniqueness results can be derived more easily. Therefore, depending on the value of t'_3 , two distinct cases occur.

6.5.2.1 The Case of $t_3 = 0$

Suppose $t_3 = 0$ and, therefore, $t'_3 = 0$. This implies that $m'_{11} = 0$ for the correct transformation (as we show later, this may not be the case for an incorrect transformation since then $m'_{11} \neq t'_3$) and the 12 equations reduce to

$$\begin{array}{ll} m'_1 = -w'_2 & m'_2 = w'_1 - t'_2 \\ m'_{11} = 0 & m'_{22} = -d'_2 t'_2 \\ m'_{21} = w'_3 & m'_{12} = -w'_3 - d'_1 t'_2 \\ m'_{111} = 0 & m'_{222} = (2 - d'_{22}) t'_2 \\ m'_{121} = t'_2 & m'_{212} = -t'_2 d'_{12} \\ m'_{221} = 0 & m'_{112} = -t'_2 d'_{11} \end{array}$$

These can be solved easily for the motion and surface parameters. The solution is

$$\begin{array}{lll}
w'_1 = m'_2 + m'_{121} & w'_2 = -m'_1 & w'_3 = m'_{21} \\
t'_1 = 0 & t'_2 = m'_{121} & t'_3 = m'_{11} = 0 \\
d'_0 = 1 & d'_1 = -\frac{m'_{21} + m'_{12}}{m'_{121}} & d'_2 = -\frac{m'_{22}}{m'_{121}} \\
d'_{11} = -\frac{m'_{112}}{m'_{121}} & d'_{12} = -\frac{m'_{212}}{m'_{121}} & d'_{22} = -\frac{m'_{222} - 2m'_{121}}{m'_{121}}
\end{array}$$

6.5.2.2 Ambiguities of $t_3 = 0$

In general, the constraint equations $m'_{11} = 0$ and $m'_{111} = 0$ are not satisfied for an incorrect transformation. To show this, let us first consider

$$m'_{11} = \Omega_{1i} \Omega_{1j} m_{ij} = 0.$$

Substituting for Ω_{ij} and m_{ij} into the above equation we arrive at

$$(t_3 - d_2 t_2) \tan^2 \theta + (d_1 t_2 + d_2 t_1) \tan \theta + (t_3 - d_1 t_1) = 0.$$

When $t_3 = 0$, this factors into

$$(t_2 \tan \theta - t_1)(d_2 \tan \theta - d_1) = 0$$

with the following solutions

$$\tan \theta = \frac{t_1}{t_2} \quad \text{and} \quad \tan \theta = \frac{d_1}{d_2}.$$

The constraint equation $m'_{11} = 0$ is satisfied for the correct transformation angle; however, it is also satisfied in two other cases, when $d_i = 0$ (the constraint $m'_{11} = 0$ reduces to $0 = 0$) and when d_1/d_2 is equal to one of the two spurious solutions for the transformation angle (if they exist); that is, when

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

This implies that, at the fixation point, the surface normal, the optical axis and one of the asymptotic lines are in the same plane. Note that we can discard one of the two spurious solutions of the transformation angle and, therefore, only two solutions may be possible. The ambiguity is resolved when $t_1/t_2 = d_1/d_2$ since the two possible solutions become identical. For every other case, only the correct transformation angle satisfies $m'_{11} = 0$.

Now consider the constraint equation $m'_{111} = 0$. This can be written

$$m'_{111} = \Omega_{1i}\Omega_{1j}\Omega_{1k}m_{ijk} = 0.$$

After substituting for Ω_{ij} and m_{ijk} it reduces to a cubic equation

$$m_{222} \tan^3 \theta - (m_{221} + 2m_{122}) \tan^2 \theta + (m_{112} + 2m_{211}) \tan \theta - m_{111} = 0.$$

In terms of the motion and surface parameters, this simplifies to

$$(t_2 \tan \theta - t_1) \left((2 - d_{22}) \tan^2 \theta + 2d_{12} \tan \theta + (2 - d_{11}) \right) = 0.$$

As expected, the correct transformation angle satisfies this equation. The question is: Does either of the two spurious solutions, if they exist, also satisfy this constraint equation? In general, this seldom occurs. It may happen *if and only if* either of the two spurious solutions that are the solution of

$$d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22} = 0$$

also satisfy the quadratic equation

$$(2 - d_{22}) \tan^2 \theta + 2d_{12} \tan \theta + (2 - d_{11}) = 0.$$

It is easy to show that these two equations reduce to the same equation when

$$d_{11} + d_{22} = 2.$$

To summarize, when $t_3 = 0$,

- (1) a three-fold ambiguity arises when the surface gradient is zero ($d_i = 0$), the Gaussian curvature is negative ($d_{11}d_{22} - d_{12}^2 < 0$), and the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$) and
- (2) a two-fold ambiguity arises when the Gaussian curvature is negative ($d_{11}d_{22} - d_{12}^2 < 0$), the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and the optical axis, the surface normal, and one of the asymptotic lines are in the same plane, that is,

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

In the latter case, the ambiguity is resolved when $t_1/t_2 = d_1/d_2$ since the two solutions become identical.

6.5.2.3 The Case of $t_3 \neq 0$

The 12 constraint equations in the new coordinate system are

$$\begin{aligned}
 m'_1 &= -w'_2 & m'_2 &= w'_1 - t'_2 \\
 m'_{11} &= t'_3 & m'_{22} &= t'_3 - d'_2 t'_2 \\
 m'_{21} &= w'_3 & m'_{12} &= -w'_3 - d'_1 t'_2 \\
 m'_{111} &= 2d'_1 t'_3 & m'_{222} &= 2(t'_2 + d'_2 t'_3) - t'_2 d'_{22} \\
 m'_{121} &= t'_2 + d'_2 t'_3 & m'_{212} &= d'_1 t'_3 - t'_2 d'_{12} \\
 m'_{221} &= 0 & m'_{112} &= -t'_2 d'_{11}
 \end{aligned}$$

We can solve for w'_2 and w'_3 directly

$$w'_2 = -m'_1, \quad \text{and} \quad w'_3 = m'_{21}.$$

Eliminating d'_2 from the equations for m'_{121} and m'_{22} yields

$$t'^2_2 + m'_{121} t'_2 + m'_{11}(m'_{11} - m'_{22}) = 0.$$

The two solutions for t'_2 are given by

$$t'_2 = \frac{1}{2}(m'_{121} \pm \sqrt{m'^2_{121} - 4m'_{11}(m'_{11} - m'_{22})}).$$

The remaining parameters can be determined easily in terms of t'_2 . If we denote the two solutions of t'_2 by τ_- and τ_+ , then we have

$$\begin{aligned}
 w'_1 &= m'_2 + t'_2 & w'_2 &= -m'_1 & w'_3 &= m'_{21} \\
 t'_1 &= 0 & t'_2 &= \tau_- \text{ and } \tau_+ & t'_3 &= m'_{11} \\
 d'_0 &= 1 & d'_1 &= \frac{m'_{111}}{2m'_{11}} & d'_2 &= \frac{m'_{121} - t'_2}{m'_{11}} \\
 d'_{11} &= -\frac{m'_{112}}{t'_2} & d'_{12} &= -\frac{2m'_{122} - m'_{111}}{2t'_2} & d'_{22} &= -\frac{m'_{222} - 2m'_{121}}{t'_2}
 \end{aligned}$$

6.5.2.4 Ambiguities of $t_3 \neq 0$

One may think that a three-fold ambiguity in the solution of the transformation angle and a two-fold ambiguity in the solution of t'_2 may bring up to a six-fold ambiguity in the solution. We now show that the ambiguity can be, at most, three-fold. More precisely, the solution for t'_2 is unique when there are three possible solutions for the transformation angle. Also, the correct transformation angle can be identified when there is a two-fold ambiguity in the solution of t'_2 .

Of the two solutions for t'_2 , we have to eliminate the spurious one. Substituting for m'_{11} , m'_{22} , and m'_{121} into the equation for t'_2 in terms of the motion and surface parameters, the two solutions are written

$$t'_2 = t'_2 \quad \text{and} \quad t'_2 = d'_2 t'_3.$$

We assumed that $t_i = t'_i \neq 0$; that is, t_1 and t_2 cannot be zero simultaneously (the special case of $t_i = t'_i = 0$ will be addressed later). Therefore, the spurious solution, $d'_2 t'_3$, can be distinguished immediately if $d'_2 = 0$ or $d'_i = d_i = 0$. When both solutions are nonzero, the acceptable solution is the one that satisfies the extra constraint equation we have not used yet (there is an extra constraint equation since we have 12 essential parameters and only 11 unknowns). This is

$$m'_{12} + m'_{21} = -d'_1 t'_2.$$

In general, only one solution of t'_2 (provided that the two are different), the correct one, satisfies this constraint. The ambiguity cannot be resolved if $d'_1 = 0$ since the above equation becomes independent of t'_2 , that is,

$$m'_{12} + m'_{21} = 0.$$

In this case, we have

$$d'_1 = \Omega_{1i} d_i = 0 \quad \text{and} \quad t'_1 = \Omega_{1i} t_i = 0$$

which reduce to

$$d_1 \cos \theta - d_2 \sin \theta = 0 \quad \text{and} \quad t_1 \cos \theta - t_2 \sin \theta = 0.$$

Therefore, the two-fold ambiguity in the solution of t'_2 is unresolved *if and only if* $\tan \theta = d_1/d_2 = t_1/t_2$. As mentioned earlier, this ambiguity is at most a two-fold one and occurs for the correct transformation (we show later that the ambiguity is resolved when we consider the higher-order terms of the motion field).

In general, we may obtain three solutions for the transformation angle, each giving rise to a possible solution for the motion and surface parameters. In most cases, the multiple ambiguity in the transformation angle also can be resolved by the extra constraint equation

$$m'_{12} + m'_{21} = -d'_1 t'_2.$$

If we substitute

$$m'_{12} = \Omega_{1i} \Omega_{2j} m_{ij}, \quad m'_{21} = \Omega_{2i} \Omega_{1j} m_{ij}, \quad d'_1 = \Omega_{1i} d_i, \quad \text{and} \quad t'_2 = \Omega_{2i} t_i$$

into the above equation and simplify the result, we arrive at

$$(t_2 \tan \theta - t_1)(d_1 \tan \theta + d_2) = 0.$$

The correct transformation angle satisfies this constraint equation; however, it is also satisfied trivially ($0=0$) when $d_i=0$. In this case, we must have $m'_{111}=0$ which occurs when the mean curvature is equal to one, as shown earlier. Therefore, when $d_i=0$ and $\frac{1}{2}(d_{11}+d_{22})=1$ simultaneously, every solution of the transformation angle satisfies the extra constraint equation and, therefore, there is a three-fold ambiguity in the solution (note that the solution of t'_2 is unique in this case). To summarize, when $t_3 \neq 0$,

- (1) there is a three-fold ambiguity, one solution for each possible transformation angle, if the surface gradient is zero ($d_i=0$), the Gaussian curvature is negative ($d_{11}d_{22}-d_{12}^2 < 0$), and the mean curvature is unity ($\frac{1}{2}(d_{11}+d_{22})=1$), and
- (2) there is a two-fold ambiguity, two possible solutions for t'_2 that cannot be distinguished, when $d_1/d_2=t_1/t_2$.

6.5.2.5 Information in the Higher-Order Terms of the Motion Field

The relative motion between an observer and a scene with a smooth surface gives rise to a smoothly-varying motion field (at least within the boundaries of depth discontinuity). So far, we have shown that, in most cases, the solution for $\{t_i, t_3\}$, $\{w_i, w_3\}$, d_i , and d_{ij} can be determined uniquely from the twelve equations in

$$\begin{aligned} m_i &= \epsilon_{ji} w_j - t_i, \\ m_{ij} &= \delta_{ij} t_3 - \epsilon_{ij} w_3 - d_i t_j, \\ m_{ijk} &= (\epsilon_{ik} w_l + t_3 d_i) \delta_{jk} + (\epsilon_{lj} w_l + t_3 d_j) \delta_{ik} - t_k d_{ij}. \end{aligned}$$

These are the coefficients of the Taylor series expansion of the resulting motion field up to the second-order terms. Some special cases can give rise to an ambiguity, up to three-fold, in the interpretation of the underlying 3-D motion. This, however, does not imply that the solution is non-unique since we have considered only up to the second-order terms of the motion field. We will show that some interpretations are inconsistent with the true motion field when we consider the higher-order terms. Therefore, the constraints imposed through the higher-order terms of the flow field can be used to distinguish among different solutions. Again, it should be noted that, in practice, even the second-order terms cannot be computed robustly (Le Guilloux [1986]); therefore, the uniqueness analysis should not be considered as a method for computing the surfaces and motion parameters in closed form.

The coefficients of the cubic terms of the motion field are given by

$$m_{ijkl} = (d_{ij} \delta_{kl} + d_{ik} \delta_{jl} + d_{jk} \delta_{il}) t_3 - d_{ijk} t_l.$$

These, in component form, are

$$m_{1111} = 3t_3 d_{11} - t_1 d_{111} \qquad m_{2222} = 3t_3 d_{22} - t_2 d_{222}$$

$$\begin{aligned}
m_{2111} &= 2t_3d_{12} - t_1d_{211} & m_{1222} &= 2t_3d_{12} - t_2d_{122} \\
m_{2211} &= t_3d_{22} - t_1d_{122} & m_{1122} &= t_3d_{11} - t_2d_{211} \\
m_{2221} &= -t_1d_{222} & m_{1112} &= -t_2d_{111}
\end{aligned}$$

Performing the same coordinate transformation as before and setting $t'_1 = 0$ we obtain

$$\begin{aligned}
m'_{1111} &= 3t'_3d'_{11} & m'_{2222} &= 3t'_3d'_{22} - t'_2d'_{222} \\
m'_{2111} &= 2t'_3d'_{12} & m'_{1222} &= 2t'_3d'_{12} - t'_2d'_{122} \\
m'_{2211} &= t'_3d'_{22} & m'_{1122} &= t'_3d'_{11} - t'_2d'_{211} \\
m'_{2221} &= 0 & m'_{1112} &= -t'_2d'_{111}
\end{aligned}$$

where $m'_{ijkl} = \Omega_{im}\Omega_{jn}\Omega_{kp}\Omega_{lq}m_{mnpq}$. We can solve for d'_{ijk} in terms of t'_3 , d'_{ij} , and t'_2 using the equations for m'_{ijk2} ,

$$\begin{aligned}
d'_{111} &= -\frac{m'_{1112}}{t'_2} & d'_{112} &= \frac{(t'_3d'_{11} - m'_{1122})}{t'_2} \\
d'_{122} &= \frac{(2t'_3d'_{12} - m'_{1222})}{t'_2} & d'_{222} &= \frac{(3t'_3d'_{22} - m'_{2222})}{t'_2}
\end{aligned}$$

To resolve the ambiguity in the solution, if it exists, we consider two cases as before.

When $t_3 \neq 0$, two types of ambiguities arise; they are either a two-fold ambiguity in the solution of t'_2 or a three-fold ambiguity in the solution of the transformation angle.

Each of the two possible solutions for t'_2 (when $t_1/t_2 = d_1/d_2$) gives rise to a possible solution for d'_{ij} . These, as shown earlier, are

$$d'_{11} = -\frac{m'_{112}}{t'_2}, \quad d'_{12} = -\frac{2m'_{122} - m'_{111}}{2t'_2}, \quad \text{and} \quad d'_{22} = -\frac{m'_{222} - 2m'_{121}}{t'_2}.$$

We can distinguish the correct solution by observing that only one of the two solutions (the one that corresponds to the true solution for t'_2) can be consistent with that obtained from the equations in m'_{ij11} ,

$$d'_{11} = \frac{m'_{1111}}{3m'_{11}}, \quad d'_{12} = \frac{m'_{2111}}{2m'_{11}}, \quad \text{and} \quad d'_{22} = \frac{m'_{2211}}{m'_{11}}.$$

Let us now address the transformation angle ambiguity that occurs when $d_i = 0$. First, consider the special case $d'_{ijk} = d_{ijk} = 0$. The expression for m_{ij} (when $d_i = 0$) depends only on t_3 and w_3 that are unaffected by the coordinate transformation. Therefore, t_3 and w_3 are the same for the three solutions (the true solution and the two spurious ones). Now, only one of the three solutions can satisfy the constraint

$$m_{ij11} = (1 + \delta_{i1} + \delta_{j1})t_3d_{ij}$$

unless d_{ij} is the same for the three solutions. Examining the expressions for m_{ijk} reveals that this can only happen if t_i is the same for the three solutions. Finally, examining the equations for m_i shows that w_i has to be the same for every possible solution. We conclude that the three solutions have to be identical.

Now suppose $d'_{ijk} \neq 0$. The constraint equations

$$m'_{ijk1} = (\delta_{i1} + \delta_{j1} + \delta_{k1}) t'_3 d'_{ij}$$

after using

$$d'_{ij} = \Omega_{ik} \Omega_{jl} d_{kl},$$

$$m'_{ijkl} = \Omega_{im} \Omega_{jn} \Omega_{kp} \Omega_{lq} m_{mnpq},$$

$$t'_3 = m'_{11} = m_{11} \cos^2 \theta - (m_{12} + m_{21}) \sin \theta \cos \theta + m_{22} \sin^2 \theta,$$

reduce to

$$\begin{aligned} (t_2 \tan \theta - t_1) (d_{222} \tan^3 \theta - 3d_{122} \tan^2 \theta + 3d_{211} \tan \theta - d_{111}) = \\ 3(t_2 \tan \theta - t_1) (d_2 \tan \theta - d_1) (d_{22} \tan^2 \theta - 2d_{12} \tan \theta + d_{11}), \end{aligned}$$

$$\begin{aligned} (t_2 \tan \theta - t_1) (d_{122} \tan^3 \theta + (d_{222} - 2d_{211}) \tan^2 \theta + (d_{111} - 2d_{122}) \tan \theta + d_{211}) = \\ 2(t_2 \tan \theta - t_1) (d_2 \tan \theta - d_1) (d_{12} \tan^2 \theta + (d_{22} - d_{11}) \tan \theta + d_{12}), \end{aligned}$$

$$\begin{aligned} (t_2 \tan \theta - t_1) (d_{211} \tan^3 \theta - (d_{111} - 2d_{122}) \tan^2 \theta + (d_{222} - 2d_{211}) \tan \theta - d_{122}) = \\ (t_2 \tan \theta - t_1) (d_2 \tan \theta - d_1) (d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22}), \end{aligned}$$

$$(t_2 \tan \theta - t_1) (d_{111} \tan^3 \theta + 3d_{112} \tan^2 \theta + 3d_{221} \tan \theta + d_{222}) = 0.$$

The correct transformation angle, $\tan \theta = t_1/t_2$, is a common solution of these equations. When $d_i = 0$, we must have

$$\begin{aligned} d_{222} \tan^3 \theta - 3d_{122} \tan^2 \theta + 3d_{211} \tan \theta - d_{111} &= 0, \\ d_{122} \tan^3 \theta + (d_{222} - 2d_{211}) \tan^2 \theta + (d_{111} - 2d_{122}) \tan \theta + d_{211} &= 0, \\ d_{211} \tan^3 \theta - (d_{111} - 2d_{122}) \tan^2 \theta + (d_{222} - 2d_{211}) \tan \theta - d_{122} &= 0, \\ d_{111} \tan^3 \theta + 3d_{112} \tan^2 \theta + 3d_{221} \tan \theta + d_{222} &= 0. \end{aligned}$$

Writing these in the form

$$\begin{pmatrix} -1 & 3 \tan \theta & -3 \tan^2 \theta & \tan^3 \theta \\ \tan \theta & 1 - 2 \tan^2 \theta & \tan^3 \theta - 2 \tan \theta & \tan^2 \theta \\ -\tan^2 \theta & \tan^3 \theta - 2 \tan \theta & -1 + 2 \tan^2 \theta & \tan \theta \\ \tan^3 \theta & 3 \tan^2 \theta & 3 \tan \theta & 1 \end{pmatrix} \begin{pmatrix} d_{111} \\ d_{211} \\ d_{221} \\ d_{222} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

we can show that the determinant of the 4×4 matrix is $(1 + \tan^2 \theta)^4 > 0$. This implies that the above equations are satisfied *if and only if* $d_{ijk} = 0$. Since we assumed that $d_{ijk} \neq 0$, we conclude that we can resolve the 3-fold ambiguity in the transformation angle.

When $t_3 = 0$, the only ambiguity is that due to the transformation angle. This occurs when either $d_i = 0$ or when d_1/d_2 is equal to a spurious solution of the cubic equation solved for the transformation angle. In either case, the constraint equations

$$m'_{ijk1} = (\delta_{i1} + \delta_{j1} + \delta_{k1})t'_3 d'_{ij}$$

reduce to

$$\begin{aligned} (t_2 \tan t - t_1)(d_{222} \tan^3 \theta - 3d_{122} \tan^2 \theta + 3d_{211} \tan \theta - d_{111}) &= 0, \\ (t_2 \tan t - t_1)(d_{122} \tan^3 \theta - (d_{222} - 2d_{211}) \tan^2 \theta + (d_{111} - 2d_{122}) \tan \theta - d_{211}) &= 0, \\ (t_2 \tan t - t_1)(d_{211} \tan^3 \theta - (d_{111} - 2d_{122}) \tan^2 \theta + (d_{222} - 2d_{211}) \tan \theta - d_{122}) &= 0, \\ (t_2 \tan t - t_1)(d_{111} \tan^3 \theta - 3d_{112} \tan^2 \theta + 3d_{221} \tan \theta - d_{222}) &= 0. \end{aligned}$$

As explained earlier, only $\tan \theta = t_1/t_2$ satisfies all of the above equations simultaneously unless $d_{ijk} = 0$. When $d_{ijk} = 0$ and $t_3 = 0$, we obtain $m_{ijkl} = 0$; that is, the cubic terms of the flow field vanish and cannot be used to resolve the three-fold ambiguity. In this case, we can repeat the same procedure given above with the fourth-order terms of the flow field, m_{ijklm} . It is easy to see that the ambiguity remains unresolved *if and only if*

$$d_{ijk} = 0, \quad d_{ijkl} = 0, \quad \dots \quad d_{i_1 i_2 \dots i_n} = 0.$$

We conclude that neither the two-fold ambiguity nor the three-fold ambiguity can be resolved when the surface is quadratic.

6.5.3 The Cases of $d_{ij} = 0$ or $t_i = 0$

In the previous sections, it was assumed that $d_{ij} \neq 0$ and $t_i \neq 0$. This is necessary since, otherwise, the coefficients of the cubic equation for the transformation angle,

$$m_{112} \tan^3 \theta - (m_{111} - 2m_{122}) \tan^2 \theta + (m_{222} - 2m_{211}) \tan \theta - m_{221} = 0,$$

vanish. When either of the two conditions occur, the 12 equations cannot be solved as before. Instead, we can use the method for planar surfaces to determine the motion and the tangent plane parameters. The essential parameters of the planar surface up to a shift factor l are given by

$$\mathbf{P}' = \mathbf{P} - l\mathbf{I} = \begin{pmatrix} -m_{11} & -m_{12} & m_{122} + m_1 \\ -m_{21} & -m_{22} & m_{211} + m_2 \\ -m_1 & -m_2 & 0 \end{pmatrix},$$

where $2l$ is the middle eigenvalue of $(\mathbf{P}' + \mathbf{P}'')$. The eigenvalue-eigenvector decomposition of \mathbf{P} is used to determine the sought-after parameters. Of course, there is a two-fold ambiguity in the solution that we have to contend with unless $\{t_i, t_3\} = t_3\{d_i, 1\}$; that is, if the translation is along the normal to the tangent plane. Also, the dual solution when $t_3 = 0$ may be considered a degenerate one since the image of the dual plane reduces to a line. As before, the equations for the higher-order terms of the motion field are used to resolve the two-fold ambiguity. We distinguish 4 cases:

(1) $\{t_i, t_3\} = 0$: when the motion is purely rotational, the solution is unique; however, we cannot recover the surface structure. To determine the surface parameters, we should have either $t_3 \neq 0$ or $t_i \neq 0$.

(2) $t_3 \neq 0$, $t_i = 0$, and $d_i = 0$: In this case, the solution is unique since the translation vector is parallel to the surface normal. We can determine d_{ij} from the equation

$$d_{ij} = \frac{m_{ijkk}}{(1 + \delta_{ik} + \delta_{jk})t_3}.$$

Similarly, the other surface parameters may be obtained uniquely from the higher-order terms in the flow field (for example, d_{ijk} is determined from m_{ijkl}).

(3) $t_i \neq 0$ and $d_i = 0$: In this case, the dual solution is $\tilde{t}_i = 0$ and $\tilde{d}_i \neq 0$ (also $\tilde{t}_3 \neq 0$). For the true solution, we must have $d_{ij} = 0$ (note that either $t_i = 0$ or $d_{ij} = 0$). For the dual solution, \tilde{d}_{ij} is obtained from

$$\tilde{d}_{ij} = \frac{m_{ijkk}}{(1 + \delta_{ik} + \delta_{jk})\tilde{t}_3}.$$

If $m_{ijkl} \neq 0$, that is, $\tilde{d}_{ij} \neq 0$, the two cases can be distinguished. If $m_{ijkl} = 0$, then $\tilde{d}_{ij} = 0$. But $m_{ijkl} = 0$ and $d_{ij} = 0$ implies that $d_{ijk} = 0$. Extending this argument, we find that we can either distinguish between the two solutions (when, for some n , $d_{i_1 i_2 \dots i_n}$ is shown to be zero and $\tilde{d}_{i_1 i_2 \dots i_n}$ is non-zero) or we must have

$$d_{ij} = \tilde{d}_{ij} = 0, \quad d_{ijk} = \tilde{d}_{ijk} = 0, \quad \dots \quad d_{i_1 i_2 \dots i_n} = \tilde{d}_{i_1 i_2 \dots i_n} = 0.$$

In the former case, the solution is unique while, in the latter case, the surface has to be planar and we cannot resolve the two-fold ambiguity.

(4) $t_3 \neq 0$, $t_i \neq 0$, and $d_i \neq 0$: The dual solution is of the form $\tilde{t}_3 \neq 0$, $\tilde{t}_i \neq 0$, and $\tilde{d}_i \neq 0$. In this case, we have $d_{ij} = \tilde{d}_{ij} = 0$ and

$$d_{ijk} = -\frac{m_{ijk1}}{t_1} \text{ or } d_{ijk} = -\frac{m_{ijk2}}{t_2} \quad \text{and} \quad \tilde{d}_{ijk} = -\frac{m_{ijk1}}{t_1} \text{ or } \tilde{d}_{ijk} = -\frac{m_{ijk2}}{t_2}.$$

Suppose we use m_{ijk1} to determine d_{ijk} and \tilde{d}_{ijk} . Then the two solutions can be distinguished using the equation for m_{ijk2} unless either $d_{ijk} = 0$ (which implies $\tilde{d}_{ijk} = 0$) or $t_1/t_2 = \tilde{t}_1/\tilde{t}_2$ (note that $\tilde{t}_1/\tilde{t}_2 = d_1/d_2$ from the equation for a dual plane). We can show that the solution is unique in either case using an argument similar to the one given in the previous case (constraints from the higher-order terms resolve the ambiguity) unless the surface is planar.

There are no other cases to be considered. We conclude that the solution is always unique when either $t_i = 0$ or $d_{ij} = 0$ provided that the surface is not planar.

6.6 Ambiguities in the Motion of Smooth Surfaces

We have shown that the ambiguity in motion vision occurs in very rare situations. The ambiguity is restricted to quadratic surfaces with non-positive Gaussian curvature and can be either three-fold (for surfaces with negative Gaussian curvature) or two-fold (for surfaces with negative or zero Gaussian curvature). Planar surfaces can be viewed as a degenerate case of surfaces with zero Gaussian curvature where both principal curvatures vanish.

In the case of planar surfaces, the true solution and the dual solution follow the relationship

$$\mathbf{n}' = k\mathbf{t}, \quad \mathbf{t}' = 1/k\mathbf{n}, \quad \text{and} \quad \boldsymbol{\omega}' = \boldsymbol{\omega} + \mathbf{n} \times \mathbf{t},$$

where \mathbf{n} is the normal of the planar surface, $\boldsymbol{\omega}$ and \mathbf{t} are the rotational and translational components of the observer motion, and k is some non-zero arbitrary constant that represents the scale-factor ambiguity inherent in the motion vision problem. Clearly, the solution is unique if the vectors \mathbf{n} and \mathbf{t} are parallel; that is, the observer translates in a direction perpendicular to the planar surface. Also, when the translation is parallel to the image plane ($t_3 = 0$), the dual solution may be considered a degenerate one since the image of the dual plane reduces to a single line corresponding to viewing the planar surface "from the side".

In the case quadratic surfaces, we have shown that the ambiguity arises in two special cases:

- (1) A three-fold ambiguity arises when the surface gradient is zero ($d_i = 0$), the Gaussian curvature is negative ($d_{11}d_{22} - d_{12}^2 < 0$), and the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$).
- (2) A two-fold ambiguity arises when the Gaussian curvature is negative ($d_{11}d_{22} - d_{12}^2 < 0$), the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and the optical axis, the surface normal, and one of the asymptotic lines are in the same plane, that is,

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

Furthermore, the observer must move parallel to the image plane; that is, $t_3 = 0$.

6.6.1 Surfaces That Give Rise to an Ambiguity

We have shown that the ambiguity in motion vision is restricted to quadratic surfaces with non-positive Gaussian curvature. This ambiguity can be either three-fold (for surfaces with negative Gaussian curvature) or two-fold (for surfaces with zero and negative Gaussian curvature). Planar surfaces can be viewed as a degenerate case of surfaces with zero Gaussian curvature where both principal curvatures vanish.

Consider, once more, the surface given by

$$d = 1 + d_1x + d_2y + d_{11}x^2 + d_{12}xy + d_{22}y^2.$$

When the Gaussian curvature is negative, the ambiguity is either three-fold if $d_i = 0$ or two-fold if

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

In either case, the mean curvature is unity ($\frac{1}{2}(d_{11} + d_{22}) = 1$). It is easy to see that the surface is a hyperboloid of one sheet. We simply have to note that the conic section

$$1 + d_1x + d_2y + d_{11}x^2 + d_{12}xy + d_{22}y^2 = 0$$

is the image of the boundaries of the surface. When $d_{11}d_{22} - d_{12}^2 < 0$, the conic section is a hyperbola. This implies that the surface can be a hyperboloid of one or two sheets. The origin of the image plane which is a point on the surface ($d = 1$ when $x = y = 0$) lies within a region that is the image of a hyperboloid of one sheet. Therefore, the surface is a hyperboloid of one sheet.

For surfaces with zero Gaussian curvature, the three-fold ambiguity degenerates to a two-fold one since the two-spurious solution become identical. Consider, for example, the two-fold ambiguity in the case where $d_i = 0$. Then the equation of the surface is given by

$$d = 1 + d_{11}x^2 + d_{12}xy + d_{22}y^2,$$

with the constraint that

$$d_{11}d_{22} - d_{12}^2 = 0 \quad \text{and} \quad \frac{1}{2}(d_{11} + d_{22}) = 1.$$

We can show that $2 > d_{11} > 0$ and $2 > d_{22} > 0$. To see this, note that if $d_{11} < 0$, then we must have $d_{22} < 0$ (because their product is d_{12}^2). But both cannot be negative since this violates the constraint that the mean curvature is unity. Similarly, neither can be larger than two since then the other has to be negative. Then the equation of the surface can be written

$$d = 1 + \frac{1}{2}(\sqrt{d_{11}}x + \text{sgn}(d_{12})\sqrt{d_{22}}y)^2 \geq 1.$$

As expected, the surface intersects the image plane along the line

$$\sqrt{d_{11}}x + \text{sgn}(d_{12})\sqrt{d_{22}}y = 0$$

which is the only asymptotic direction of the surface. If we multiply both sides of the equation for d by Z^2 and simplify we obtain

$$\left(Z - \frac{1}{2}\right)^2 + \frac{1}{2}(\sqrt{d_{11}}X + \text{sgn}(d_{12})\sqrt{d_{22}}Y)^2 = \frac{1}{4}.$$

Therefore, the surface of ambiguity degenerates to a circular cylinder.

It should be noted that we restricted our earlier analysis to only a subset of the class of quadratic surfaces. This special subset includes any surface whose inverse depth function is given by a second-order polynomial in the image coordinates. Using a global analysis, Horn [1986b] has also shown that, for curved surfaces, an ambiguity may arise in the case of certain hyperboloids of one sheet that are viewed from a point on the surface. For these surfaces, the inverse depth function is given by the ratio of a second-order polynomial to a first-order polynomial in the image coordinates. We will show that this is a generalization of the results presented earlier. Using the generalized representation of quadratic surfaces, we then derive the relationship among multiple solutions. We need to do this since we cannot hope to recover motion robustly, in the case of smooth surfaces, from an analytical method like the one presented in the previous sections. An iterative method, on the other hand, may converge to any one of the possible solutions (when multiple solutions exist) depending on the initial condition. When we know the number of possible solutions in a particular case, several runs of the algorithm may be necessary before we obtain every possible solution; however, this is expensive computationally. Further, in some cases, a solution may be hard to find (this is important if it is the true solution) if it has a small radius of convergence.

Therefore, it is not only important to know what circumstances can give rise to an ambiguity and the number of possible solutions, but it is equally important to know the relationship between the true and the spurious solutions.

6.6.2 Critical Surface Pairs

We have determined the circumstances that give rise to an ambiguity in the interpretation of the motion field. Horn [1986b], instead, addressed the following question: What surfaces could lead to the same motion field given two different motions? He calls two surfaces that yield the same motion field for two given motions a *critical surface pair*. He shows that members of such a pair are either planes or some quadratic surfaces, namely, certain hyperboloids of one sheet, that are viewed from a point on their surface.

The equation of a general quadric is given by

$$R_0 + \mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{M} \mathbf{R} = 0,$$

where $\mathbf{R} = [X, Y, Z]^T$ is a point on the surface, \mathbf{n} is the normal of the tangent plane of the surface at the origin, and \mathbf{M} , a 3×3 symmetric matrix, includes the coefficients of the second-order terms.

If the viewing point (origin of the coordinate system) is located on the surface of the quadric, we must have $R_0 = 0$. Setting R_0 equal to zero we obtain

$$\mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{M} \mathbf{R} = 0.$$

Since we deal with images, it is more convenient to write this equation in terms of image coordinates. Using the perspective projection equation, $\mathbf{R} = Z \mathbf{r}$ (as before, $\mathbf{r} = [x, y, 1]^T$ is the image of the point \mathbf{R}), and dividing by Z we arrive at

$$\mathbf{n}^T \mathbf{r} + \frac{1}{2} Z \mathbf{r}^T \mathbf{M} \mathbf{r} = 0.$$

Finally, solving for Z we obtain

$$Z = -\frac{2\mathbf{n}^T \mathbf{r}}{\mathbf{r}^T \mathbf{M} \mathbf{r}}.$$

Therefore, the depth function can be written as the ratio of a first-order polynomial to a second-order polynomial in terms of the image coordinates. In terms of the inverse depth function, this is written as

$$d = -\frac{\mathbf{r}^T \mathbf{M} \mathbf{r}}{2\mathbf{n}^T \mathbf{r}}.$$

Note that there are only eight degrees of freedom in the above representation of the surface since the coefficients can be scaled by an arbitrary scale factor (eight independent parameters in \mathbf{n} and \mathbf{M}). This is a different scale-factor ambiguity from that found in the motion vision problem.

Horn's important results regarding critical surfaces can be summarized in the following theorem that we refer to as Horn's Theorem of critical surfaces.

Horn's Theorem of Critical Surfaces: Let $\{\mathbf{t}, \boldsymbol{\omega}\}$ and $\{\tilde{\mathbf{t}}, \tilde{\boldsymbol{\omega}}\}$ denote the motion vectors corresponding to a critical surface pair S and \tilde{S} , and let $\delta\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}$. Then the two surfaces in a critical surface pair, S and \tilde{S} , respectively, are given by

$$\mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{M} \mathbf{R} = 0 \quad \text{and} \quad \mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \tilde{\mathbf{M}} \mathbf{R} = 0,$$

where $\mathbf{n} = \tilde{\mathbf{t}} \times \mathbf{t}$ and

$$\mathbf{M} = \tilde{\mathbf{t}} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \tilde{\mathbf{t}}^T - 2(\tilde{\mathbf{t}} \cdot \delta\boldsymbol{\omega}) \mathbf{I} \quad \text{and} \quad \tilde{\mathbf{M}} = \mathbf{t} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \mathbf{t}^T - (\mathbf{t} \cdot \delta\boldsymbol{\omega}) \mathbf{I}.$$

Note that the equations of the two surfaces depend on the difference (and not on the absolute values) of the two rotational vectors. Also, the equation of the first surface depends on the translational vector corresponding to the second surface, and vice versa.

If we let $\sigma = \|\tilde{\mathbf{t}}\| \|\delta\boldsymbol{\omega}\|$ and $\tau = \frac{1}{\sigma}(\tilde{\mathbf{t}} \cdot \delta\boldsymbol{\omega})$, then the eigenvalues of \mathbf{M} , in ascending order, are given by

$$\lambda_1 = -\sigma(1 + \tau), \quad \lambda_2 = -2\sigma\tau, \quad \text{and} \quad \lambda_3 = \sigma(1 - \tau).$$

The corresponding normalized eigenvectors are

$$\mathbf{u}_1 = \frac{(\hat{\mathbf{t}} - \delta\hat{\boldsymbol{\omega}})}{\sqrt{2(1 - \tau)}}, \quad \mathbf{u}_2 = \frac{(\hat{\mathbf{t}} \times \delta\hat{\boldsymbol{\omega}})}{\sqrt{1 - \tau^2}}, \quad \text{and} \quad \mathbf{u}_3 = \frac{(\hat{\mathbf{t}} + \delta\hat{\boldsymbol{\omega}})}{\sqrt{2(1 + \tau)}},$$

where $\hat{\mathbf{t}}$ and $\delta\hat{\boldsymbol{\omega}}$ are unit vectors in the directions of $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$, respectively. Since $\sigma > 0$ and $-1 \leq \tau \leq 1$, it is clear that $\lambda_1 \leq 0$ and $\lambda_3 \geq 0$. Further, we have $\lambda_1 + \lambda_3 = \lambda_2$ (from symmetry, we can derive similar results for the second surface in terms of the eigenvalue-eigenvector decomposition of $\tilde{\mathbf{M}}$ instead). This eigenvalue-eigenvector decomposition is reminiscent of the one given for the 3×3 matrix of the essential parameters of planar surfaces. Since the eigenvalues of the matrix \mathbf{M} (and, similarly, $\tilde{\mathbf{M}}$) are constrained, we conclude that not every quadric can be a critical surface.

6.7 Relationship Among Multiple Solutions

Suppose we have determined a motion-and-surface pair, $\{\mathbf{t}, \boldsymbol{\omega}\}$ and S , that is consistent with the data. The data can be the velocity of image points (that is, the motion field or, more precisely, optical flow), the correspondences of some dominant feature points, or simply the brightness gradients computed from an image sequence. Further, suppose we know that the solution is not unique (it either satisfies the constraints of, or due to noise is near, an ambiguous case). From Horn's theorem, the multiple solutions should satisfy the equations given for a critical surface pair. If we denote a spurious solution by the motion-and-surface pair, $\{\tilde{\mathbf{t}}, \tilde{\boldsymbol{\omega}}\}$ and \tilde{S} , then the equations for S and \tilde{S} should, respectively, satisfy

$$\mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{M} \mathbf{R} = 0 \quad \text{and} \quad \mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \tilde{\mathbf{M}} \mathbf{R} = 0,$$

where $\mathbf{n} = \tilde{\mathbf{t}} \times \mathbf{t}$,

$$\mathbf{M} = \tilde{\mathbf{t}} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \tilde{\mathbf{t}}^T - 2(\tilde{\mathbf{t}} \cdot \delta\boldsymbol{\omega})\mathbf{I}, \quad \text{and} \quad \tilde{\mathbf{M}} = \mathbf{t} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \mathbf{t}^T - (\mathbf{t} \cdot \delta\boldsymbol{\omega})\mathbf{I}.$$

Again, $\delta\boldsymbol{\omega}$ denotes the vector difference of the two rotation vectors $\tilde{\boldsymbol{\omega}}$ and $\boldsymbol{\omega}$ ($\delta\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}$). We will use the equations for \mathbf{M} and \mathbf{n} to determine $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$. Consequently, we can easily solve for $\tilde{\boldsymbol{\omega}}$ and determine $\tilde{\mathbf{M}}$ that gives us the second surface \tilde{S} .

Both $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$ can be determined from the eigenvalue-eigenvector decomposition of \mathbf{M} . As before, let $\sigma = \|\tilde{\mathbf{t}}\| \|\delta\boldsymbol{\omega}\|$ and $\tau = \frac{1}{\sigma}(\tilde{\mathbf{t}} \cdot \delta\boldsymbol{\omega})$. Then the eigenvalues of \mathbf{M} , in ascending order, are given by

$$\lambda_1 = -\sigma(1 + \tau), \quad \lambda_2 = -2\sigma\tau, \quad \text{and} \quad \lambda_3 = \sigma(1 - \tau).$$

The corresponding normalized eigenvectors are

$$\mathbf{u}_1 = \frac{(\hat{\mathbf{t}} - \delta\hat{\boldsymbol{\omega}})}{\sqrt{2(1-\tau)}}, \quad \mathbf{u}_2 = \frac{(\hat{\mathbf{t}} \times \delta\hat{\boldsymbol{\omega}})}{\sqrt{1-\tau^2}}, \quad \text{and} \quad \mathbf{u}_3 = \frac{(\hat{\mathbf{t}} + \delta\hat{\boldsymbol{\omega}})}{\sqrt{2(1+\tau)}},$$

where $\hat{\mathbf{t}}$ and $\delta\hat{\boldsymbol{\omega}}$, as before, are unit vectors in the directions of $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$, respectively. We can compute σ and τ from

$$\sigma = \lambda_3 - \frac{1}{2}\lambda_2 \quad \text{and} \quad \tau = \frac{-\lambda_2}{2\lambda_3 - \lambda_2}.$$

It is easy to show that (see a similar derivation given in Lemma 1 for planar surfaces)

$$\hat{\mathbf{t}} = \sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1 \quad \text{and} \quad \delta\hat{\boldsymbol{\omega}} = \sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1.$$

The expression for \mathbf{M} is symmetric with respect to $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$. Another solution is obtained from

$$\hat{\mathbf{t}} = \sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1 \quad \text{and} \quad \delta\hat{\boldsymbol{\omega}} = \sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1.$$

Finally, we obtain two more solutions by reversing the sign of the above two solutions,

$$\hat{\mathbf{t}} = -\sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1 \quad \text{and} \quad \delta\hat{\boldsymbol{\omega}} = -\sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1,$$

as well as

$$\hat{\mathbf{t}} = -\sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 - \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1 \quad \text{and} \quad \delta\hat{\boldsymbol{\omega}} = -\sqrt{\frac{1}{2}(1+\tau)}\mathbf{u}_3 + \sqrt{\frac{1}{2}(1-\tau)}\mathbf{u}_1.$$

We conclude that $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$ can have up to four solutions. These solutions, however, have to be consistent with the constraint equation $\mathbf{n} = \tilde{\mathbf{t}} \times \mathbf{t}$. We can immediately resolve the sign ambiguity and, therefore, only two solutions are possible. Now, the question is: what circumstances give rise to two spurious solutions or, equivalently, when is the ambiguity a three-fold one?

The three-fold ambiguity presents itself *if and only if*

$$\mathbf{n} = \|\tilde{\mathbf{t}}\| (\hat{\mathbf{t}} \times \mathbf{t}) \quad \text{and} \quad \mathbf{n} = \|\delta\boldsymbol{\omega}\| (\delta\hat{\boldsymbol{\omega}} \times \mathbf{t}),$$

that is, *if and only if* \mathbf{t} , $\tilde{\mathbf{t}}$, and $\delta\boldsymbol{\omega}$ are in the same plane. Otherwise, only one of the two solutions can satisfy this constraint and, therefore, the ambiguity is two-fold. Note that if neither solution satisfies this constraint, we conclude that the solution is unique (clearly, in the presence of noise, we may have to specify some threshold since it is unlikely that

the constraint holds exactly). For each acceptable solution, we can obtain $\|\tilde{\mathbf{t}}\|$ from the above constraint equation and, consequently, determine $\|\delta\omega\|$ from

$$\|\delta\omega\| = \frac{\|\delta\omega\| \|\tilde{\mathbf{t}}\|}{\|\tilde{\mathbf{t}}\|} = \frac{\sigma}{\|\tilde{\mathbf{t}}\|}.$$

We can then construct the spurious surface(s) using the equation

$$\mathbf{n}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \tilde{\mathbf{M}} \mathbf{R} = 0,$$

where

$$\tilde{\mathbf{M}} = \mathbf{t} \delta\omega^T + \delta\omega \mathbf{t}^T - (\mathbf{t} \cdot \delta\omega) \mathbf{I}.$$

We will next show that these results are consistent with those we presented earlier. We then derive simple expressions for the spurious solutions for the two special cases that give rise to either a three-fold or a two-fold ambiguity. Before we proceed, we emphasize that the earlier results included quadratic surfaces whose inverse depth function can be expressed as a second-order function of the image coordinates,

$$d = d_0 + d_1 x + d_2 y + \frac{1}{2} d_{11} x^2 + d_{12} xy + \frac{1}{2} d_{22} y^2,$$

where we arbitrarily set $d_0 = 1$ because of the scale-factor ambiguity. This can be written in vector form

$$d = \frac{1}{2} \mathbf{r}^T \mathbf{D} \mathbf{r},$$

where

$$\mathbf{D} = \begin{pmatrix} d_{11} & d_{12} & d_1 \\ d_{12} & d_{22} & d_2 \\ d_1 & d_2 & 2 \end{pmatrix}.$$

In terms of object coordinates, the surface can be written as (by multiplying both sides of the equation for d by Z^2)

$$\hat{\mathbf{z}}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{D} \mathbf{R} = 0, \quad \text{where } \hat{\mathbf{z}} = (0, 0, -1)^T.$$

Comparing this with the general equation of quadrics that pass through the viewing point we immediately see that we must have $\mathbf{n} = \hat{\mathbf{z}}$. This is the restriction we have to deal with when we present the surface in a polynomial form; that is, \mathbf{n} cannot be any arbitrary vector. It is also important to note that the previous representation of the surface is a single-values function of the image coordinates,

$$d = d_0 + d_1 x + d_2 y + \frac{1}{2} d_{11} x^2 + d_{12} xy + \frac{1}{2} d_{22} y^2,$$

and that the surface does not pass through the origin even though $\mathbf{R} = \mathbf{0}$ satisfies the equation

$$\hat{\mathbf{z}}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{D} \mathbf{R} = 0.$$

By multiplying both sides of the expression for d by Z^2 we have artificially made the origin ($\mathbf{R} = \mathbf{0}$) a point on the surface. To be precise, we should define our surface as the quadric given by

$$\hat{\mathbf{z}}^T \mathbf{R} + \frac{1}{2} \mathbf{R}^T \mathbf{D} \mathbf{R} = 0, \quad \mathbf{R} \neq \mathbf{0}.$$

This is not an important issue since we simply want to exploit the similarity in the two representations to derive the expressions for the dual solutions.

Again, we let $\{\mathbf{t}, \boldsymbol{\omega}\}$ and S denote the true solution, $\{\tilde{\mathbf{t}}, \tilde{\boldsymbol{\omega}}\}$ and \tilde{S} the spurious one, and we let $\delta\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} - \boldsymbol{\omega}$. Using Horn's theorem, it follows that, for any two possible solutions, we must have $\tilde{\mathbf{t}} \times \mathbf{t} = k\hat{\mathbf{z}}$ (k represents the scale-factor ambiguity). Since $\hat{\mathbf{z}}$ is a unit vector in the negative z direction, we immediately conclude that both \mathbf{t} and $\tilde{\mathbf{t}}$ should be parallel to the image plane; that is, $t_3 = \tilde{t}_3 = 0$. This means that when the inverse depth function is a polynomial in the image coordinates, an ambiguity may occur *only and only if* the observer moves parallel to the image plane; this is consistent with the earlier results we presented. We next show that the other restrictions on the surface should also hold.

6.7.1 Case One: Three-Fold Ambiguity of Hyperboloids of One Sheet

We have shown that a three-fold ambiguity may arise if the surface is a hyperboloid of one sheet. The "surface of ambiguity" is characterized by the following conditions:

- (1) the Gaussian curvature is negative ($d_{12}^2 - d_{11}d_{22} > 0$),
- (2) the mean curvature is equal to one ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and
- (3) the surface gradient vanishes ($d_i = 0$).

Using Horn's theorem, any two solutions should satisfy the equations for the critical surface pair. Therefore, S and \tilde{S} , respectively, are given by

$$\hat{\mathbf{z}}^T \mathbf{R} + \mathbf{R}^T \mathbf{D} \mathbf{R} = 0 \quad \text{and} \quad \hat{\mathbf{z}}^T \mathbf{R} + \mathbf{R}^T \tilde{\mathbf{D}} \mathbf{R} = 0,$$

where

$$\mathbf{D} = \frac{1}{k} (\tilde{\mathbf{t}} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \tilde{\mathbf{t}}^T - (\tilde{\mathbf{t}} \cdot \delta\boldsymbol{\omega}) \mathbf{I}) \quad \text{and} \quad \tilde{\mathbf{D}} = \frac{1}{k} (\mathbf{t} \delta\boldsymbol{\omega}^T + \delta\boldsymbol{\omega} \mathbf{t}^T - (\mathbf{t} \cdot \delta\boldsymbol{\omega}) \mathbf{I}).$$

Note that we have to include the constant k due to the scale-factor ambiguity. Once we obtain $\tilde{\mathbf{t}}$ and $\delta\boldsymbol{\omega}$ up to a scale factor, k will be determined from

$$k = \|\tilde{\mathbf{t}} \times \mathbf{t}\|.$$

We can write

$$\mathbf{D} = \frac{1}{k} \left(\tilde{\mathbf{t}} \delta \omega^T + \delta \omega \tilde{\mathbf{t}}^T - (\tilde{\mathbf{t}} \cdot \delta \omega) \mathbf{I} \right),$$

in the component form

$$\begin{pmatrix} d_{11} & d_{12} & d_1 \\ d_{12} & d_{22} & d_2 \\ d_1 & d_2 & 2 \end{pmatrix} = \frac{1}{k} \begin{pmatrix} -2\tilde{t}_2 \delta \omega_2 & \tilde{t}_1 \delta \omega_2 + \tilde{t}_2 \delta \omega_1 & \tilde{t}_1 \delta \omega_3 \\ \tilde{t}_1 \delta \omega_2 + \tilde{t}_2 \delta \omega_1 & -2\tilde{t}_1 \delta \omega_1 & \tilde{t}_2 \delta \omega_3 \\ \tilde{t}_1 \delta \omega_3 & \tilde{t}_2 \delta \omega_3 & -2(\tilde{t}_1 \delta \omega_1 + \tilde{t}_2 \delta \omega_2) \end{pmatrix}.$$

Since $d_1 = d_2 = 0$, we conclude that $\delta \omega_3 = 0$. This is consistent with our earlier results since the three-fold ambiguity is due to the existence of three possible solutions for the transformation angle. Since the rotation to the new image coordinates is about the optical axis, then any motion parameter along the optical axis, for example, ω_3 , is unaffected by the transformation. Equating the two sides of the matrix equation term by term we can write

$$\begin{aligned} -\frac{2}{k} \tilde{t}_2 \delta \omega_2 &= d_{11}, \\ -\frac{1}{k} (\tilde{t}_1 \delta \omega_2 + \tilde{t}_2 \delta \omega_1) &= d_{12}, \\ -\frac{2}{k} \tilde{t}_1 \delta \omega_1 &= d_{22}, \\ -\frac{2}{k} (\tilde{t}_1 \delta \omega_1 + \tilde{t}_2 \delta \omega_2) &= 2. \end{aligned}$$

Adding the first two equations and setting the resulting equation equal to the fourth equation we arrive at

$$d_{11} + d_{22} = 2 \quad \text{or} \quad \frac{1}{2}(d_{11} + d_{22}) = 1.$$

Multiplying the first two equations and subtracting the result from the square of the third equation we obtain

$$d_{12}^2 - d_{11} d_{22} = \frac{1}{k^2} (\tilde{t}_1 \delta \omega_2 - \tilde{t}_2 \delta \omega_1)^2 \geq 0.$$

Nothing is new since these conditions are exactly the same as the ones we derived earlier. Therefore, all of the conditions in the case of a three-fold ambiguity are satisfied by the equation of a critical surface pair. We will see that we actually obtain two solutions for the second surface in a critical pair.

We can solve for $\delta \omega_1$ and $\delta \omega_2$ from the first two equations given earlier,

$$\delta \omega_1 = -\frac{k}{2} \frac{d_{22}}{\tilde{t}_1} \quad \text{and} \quad \delta \omega_2 = -\frac{k}{2} \frac{d_{11}}{\tilde{t}_2}.$$

Using these in the third equation and simplifying the result we obtain

$$d_{11}\left(\frac{\tilde{t}_1}{\tilde{t}_2}\right)^2 + 2d_{12}\left(\frac{\tilde{t}_1}{\tilde{t}_2}\right) + d_{22} = 0.$$

This can be solved for \tilde{t}_1/\tilde{t}_2 ,

$$\frac{\tilde{t}_1}{\tilde{t}_2} = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

These are the spurious solutions of the transformation angle in our earlier derivation.

Let us denote $\alpha = \tilde{t}_1/\tilde{t}_2$. In order to solve for \tilde{t}_1 and \tilde{t}_2 , we need another equation in terms of these parameters. This is obtained using the equation of a second surface. Exploiting the symmetry in the equations we can write

$$\begin{aligned} -\frac{2}{k}t_2\delta\omega_2 &= \tilde{d}_{11}, \\ -\frac{1}{k}(t_1\delta\omega_2 + t_2\delta\omega_1) &= \tilde{d}_{12}, \\ -\frac{2}{k}t_1\delta\omega_1 &= \tilde{d}_{22}, \\ -\frac{2}{k}(t_1\delta\omega_1 + t_2\delta\omega_2) &= 2. \end{aligned}$$

Substituting the expressions given earlier for $\delta\omega_1$ and $\delta\omega_2$ in the last equation, we obtain the extra sought after equation,

$$\frac{t_1}{\tilde{t}_1 d_{22}} + \frac{t_2}{\tilde{t}_2 d_{11}} = 2.$$

Substituting $\tilde{t}_1 = \alpha\tilde{t}_2$, in the above equation and solving for \tilde{t}_2 we obtain

$$\tilde{t}_2 = \frac{1}{2}\left(\frac{1}{\alpha}t_1 d_{22} + t_2 d_{11}\right).$$

Finally, using this in the equations for the remaining parameters, we arrive at the equations for a spurious solution in terms of the true solution,

$$\begin{array}{lll} \tilde{t}_1 = \alpha\tilde{t}_2 & \tilde{t}_2 = \frac{1}{2}\left(\frac{1}{\alpha}t_1 d_{22} + t_2 d_{11}\right) & \tilde{t}_3 = 0 \\ \delta\omega_1 = -\frac{k}{2\alpha}\frac{d_{22}}{t_2} & \delta\omega_2 = -\frac{k}{2}\frac{d_{11}}{t_2} & \delta\omega_3 = 0 \\ \tilde{d}_0 = 1 & \tilde{d}_1 = 0 & \tilde{d}_2 = 0 \\ \tilde{d}_{11} = \frac{t_2}{t_2}d_{11} & \tilde{d}_{12} = -\frac{1}{2t_2}(t_1 d_{11} + \frac{1}{\alpha}t_2 d_{22}) & \tilde{d}_{22} = \frac{t_1}{\alpha t_2}d_{22} \end{array}$$

where

$$\alpha = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{and} \quad k = (\alpha t_2 - t_1)\tilde{t}_2.$$

For each solution of α , we obtain one spurious solution from these equations. We can alternatively determine both spurious solutions using only one of the solutions for α . This requires that we derive another set of relationships for the dual solutions. We can do this by exploiting the symmetry in $\tilde{\mathbf{t}}$ and $\delta\omega$ in the expression for \mathbf{D} and, therefore, repeating the solution procedure given above in terms of $\delta\omega_1$ and $\delta\omega_2$ (instead of \tilde{t}_1 and \tilde{t}_2). Skipping the tedious procedure, the final results are given by

$$\begin{aligned} \tilde{t}_1 &= \frac{1}{2}(t_1 + \frac{1}{\alpha}t_2)d_{22} & \tilde{t}_2 &= \frac{\alpha}{2}(t_1 + \frac{1}{\alpha}t_2)d_{11} & \tilde{t}_3 &= 0 \\ \delta\omega_1 &= \alpha\delta\omega_2 & \delta\omega_2 &= -\frac{k}{\alpha t_1 + t_2} & \delta\omega_3 &= 0 \\ \tilde{d}_0 &= 1 & \tilde{d}_1 &= 0 & \tilde{d}_2 &= 0 \\ \tilde{d}_{11} &= \frac{2t_2}{\alpha t_1 + t_2} & \tilde{d}_{12} &= -\frac{t_1 + \alpha t_2}{\alpha t_1 + t_2} & \tilde{d}_{22} &= \frac{2\alpha t_1}{\alpha t_1 + t_2} \end{aligned}$$

where k is given by

$$k = -\frac{1}{2}(t_1 + \frac{1}{\alpha}t_2)(\alpha t_1 d_{11} - t_2 d_{22}).$$

To summarize, we can determine the two dual solutions either (1) by substituting the two values obtained for α into one of the two sets of relationships given above or (2) by substituting one of the two solutions of α into both sets of equations.

These equations shed more light into the nature of the spurious solutions. Consider the equations given for $\delta\omega$ and $\tilde{\mathbf{t}}$. Assuming that the true motion is purely translational, we see that each spurious solution is characterized by an observer rotating about an axis parallel to one of the two asymptotic directions and translating parallel to the other asymptotic direction of the true surface. For the corresponding spurious surface, the two asymptotic directions are the direction of translation and one of the asymptotic lines of the true solution (the one that is the axis of rotation of the viewer observing the surface). In general, the difference of the rotational vectors of any two solutions is aligned with the direction of translation of the third surface. Since the ambiguity in the solution and the relationship between multiple solutions are directly related to the direction of translation of the viewer and the asymptotic lines of the true surface, we will call the three directions the *characteristic lines of motion-structure ambiguity*.

6.7.2 Case Two: Two-Fold Ambiguity of Hyperboloids of One Sheet

Certain Hyperboloids of one sheet can give rise to a two-fold ambiguity. The surface of ambiguity is characterized by the following conditions:

- (1) the Gaussian curvature is negative ($d_{12}^2 - d_{11}d_{22} > 0$),
- (2) the mean curvature is equal to one ($\frac{1}{2}(d_{11} + d_{22}) = 1$), and
- (3) the optical axis, the surface normal, and one of the asymptotic lines of the surface are in the same plane, that is,

$$\text{either } \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

Again, we start with

$$\begin{pmatrix} d_{11} & d_{12} & d_1 \\ d_{12} & d_{22} & d_2 \\ d_1 & d_2 & 2 \end{pmatrix} = \frac{1}{k} \begin{pmatrix} -2\tilde{t}_2\delta\omega_2 & \tilde{t}_1\delta\omega_2 + \tilde{t}_2\delta\omega_1 & \tilde{t}_1\delta\omega_3 \\ \tilde{t}_1\delta\omega_2 + \tilde{t}_2\delta\omega_1 & -2\tilde{t}_1\delta\omega_1 & \tilde{t}_2\delta\omega_3 \\ \tilde{t}_1\delta\omega_3 & \tilde{t}_2\delta\omega_3 & -2(\tilde{t}_1\delta\omega_1 + \tilde{t}_2\delta\omega_2) \end{pmatrix}.$$

In this case, we have

$$d_1 = \frac{1}{k}\tilde{t}_1\delta\omega_3 \quad \text{and} \quad d_2 = \frac{1}{k}\tilde{t}_2\delta\omega_3.$$

Writing the ratio of the two equations, we obtain

$$\frac{\tilde{t}_1}{\tilde{t}_2} = \frac{d_1}{d_2}.$$

Since the remaining equations are exactly as in the previous case, we can use those results directly here. We showed that we must have

$$\frac{\tilde{t}_1}{\tilde{t}_2} = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

This actually has to hold for either the plus or the minus sign, not for both; then we are guaranteed to satisfy the quadratic equation that was solved for \tilde{t}_1/\tilde{t}_2 , that is, the equation

$$d_{11}\left(\frac{\tilde{t}_1}{\tilde{t}_2}\right)^2 + 2d_{12}\left(\frac{\tilde{t}_1}{\tilde{t}_2}\right) + d_{22} = 0.$$

We conclude that, in order to have a two-fold ambiguity, we must have

$$\text{either } \frac{\tilde{t}_1}{\tilde{t}_2} = \frac{d_1}{d_2} = \frac{-d_{12} + \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}} \quad \text{or} \quad \frac{\tilde{t}_1}{\tilde{t}_2} = \frac{d_1}{d_2} = \frac{-d_{12} - \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

In the event that $t_1/t_2 = d_1/d_2$, then $\tilde{t}_1/\tilde{t}_2 = t_1/t_2$ and the two solutions become identical as expected. Again, nothing new is found. We can now solve for $\delta\omega_3$ from

$$\delta\omega_3 = \frac{kd_2}{\tilde{t}_2}.$$

Using the symmetry in the equations for the two surfaces we can write

$$\tilde{d}_1 = \frac{1}{k}t_1\delta\omega_3 \quad \text{and} \quad \tilde{d}_2 = \frac{1}{k}t_2\delta\omega_3.$$

Substituting for $\delta\omega_3$ we arrive at the expressions for \tilde{d}_1 and \tilde{d}_2 ,

$$\tilde{d}_1 = \frac{t_1}{t_2}d_2 \quad \text{and} \quad \tilde{d}_2 = \frac{t_2}{t_2}d_2.$$

The remaining parameters are determined from the equations given in the previous case. All together, we obtain

$$\begin{aligned} \tilde{t}_1 &= \alpha\tilde{t}_2 & \tilde{t}_2 &= \frac{1}{2}\left(\frac{1}{\alpha}t_1d_{22} + t_2d_{11}\right) & \tilde{t}_3 &= 0 \\ \delta\omega_1 &= -\frac{k}{2\alpha}\frac{d_{22}}{t_2} & \delta\omega_2 &= -\frac{k}{2}\frac{d_{11}}{t_2} & \delta\omega_3 &= k\frac{d_2}{t_2} \\ \tilde{d}_0 &= 1 & \tilde{d}_1 &= \frac{t_1}{t_2}d_2 & \tilde{d}_2 &= \frac{t_2}{t_2}d_2 \\ \tilde{d}_{11} &= \frac{t_2}{t_2}d_{11} & \tilde{d}_{12} &= -\frac{1}{2t_2}(t_1d_{11} + \frac{1}{\alpha}t_2d_{22}) & \tilde{d}_{22} &= \frac{t_1}{\alpha t_2}d_{22} \end{aligned}$$

where $\alpha = d_1/d_2$ and $k = (\alpha t_2 - t_1)\tilde{t}_2$. We can alternatively use the second set of equations given in the previous section,

$$\begin{aligned} \tilde{t}_1 &= \frac{1}{2}(t_1 + \frac{1}{\alpha}t_2)d_{22} & \tilde{t}_2 &= \frac{\alpha}{2}(t_1 + \frac{1}{\alpha}t_2)d_{11} & \tilde{t}_3 &= 0 \\ \delta\omega_1 &= \alpha\delta\omega_2 & \delta\omega_2 &= -\frac{k}{\alpha t_1 + t_2} & \delta\omega_3 &= k\frac{d_2}{t_2} \\ \tilde{d}_0 &= 1 & \tilde{d}_1 &= \frac{t_1}{t_2}d_2 & \tilde{d}_2 &= \frac{t_2}{t_2}d_2 \\ \tilde{d}_{11} &= \frac{2t_2}{\alpha t_1 + t_2} & \tilde{d}_{12} &= -\frac{t_1 + \alpha t_2}{\alpha t_1 + t_2} & \tilde{d}_{22} &= \frac{2\alpha t_1}{\alpha t_1 + t_2} \end{aligned}$$

where k is given by

$$k = -\frac{1}{2}(t_1 + \frac{1}{\alpha}t_2)(\alpha t_1 d_{11} - t_2 d_{22}).$$

Interestingly, to obtain the correct solution from the above equations, we have to use the solution of α that is different from d_1/d_2 . One explanation is that the two sets of relationships constitute a dual set. Similarly, the two solutions of α form a dual set (of course, only one is acceptable). If the correct solution is obtained by using $\alpha = d_1/d_2$ in the first set of equations, then we must use the dual solution of α in the dual set to obtain the correct spurious solution.

6.8 Imposing Depth Positivity Constraint to Resolve Ambiguity

In addressing the issue of ambiguities in motion vision, we have shown that certain rare situations can give rise to up to three interpretations. What we will show now is that, in many cases, an incorrect interpretation may be ruled out due to the fact that it violates physical constraints. The physical constraint that we will impose is that the depth value for a point that is projected onto the image should be positive. This means that a point can be seen *if and only if* it is in front of the camera. A negative depth value corresponds to a point on a surface that either is behind the camera or translates in the opposite direction. In either case, the point cannot belong to the same physical surface that is projected into the region of the image with positive depth values.

Consider the surface,

$$d = 1 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2.$$

If the solution is acceptable, we should have $d \geq 0$. Since $d = 1/Z$, then $d = 0$ implies that $Z \rightarrow \infty$. Therefore, the conic section

$$1 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2 = 0$$

is the image of points at infinity. It is also the boundary between regions with positive and negative depth values. Negative depth values are physically impossible. In other words, the image of the surface cannot include the negative depth regions. Therefore, these regions are the image(s) of the background or other objects in the scene. In order to distinguish the correct solution from a spurious one, we want the image to include the conic sections of the possible solutions.

It is more instructive to study an ambiguous situation. We consider the case that gives rise to a three-fold ambiguity since it is easier to derive the necessary equations. In this case, we have

$$d_i = 0, \quad \frac{1}{2}(d_{11} + d_{22}) = 1, \quad \text{and} \quad d_{12}^2 - d_{11}d_{22} > 0.$$

Suppose the image plane is a circle of radius l corresponding to a field of view with half-angle $\phi = \tan^{-1}(l)$ degrees (the focal length is assumed to be unity). In order to have (some parts of) the boundaries of the true surface within the image, we want to have $l > l_c$, where l_c is the shortest distance from the origin of the image plane to a point on the conic section. Ideally, we want the field of view very large so that we can have as much of the conic section in the image as possible.

When $d_{12} \neq 0$, it can be shown (see Appendix D) that $l_c = \sqrt{x_c^2 + y_c^2}$, where

$$(x_c = \pm \sqrt{\frac{-2}{d_{22}k^2 + 2d_{12}k + d_{11}}}, \quad y_c = kx_c)$$

and

$$k = \left(\frac{d_{22} - d_{11}}{2d_{12}} \right) \pm \sqrt{\left(\frac{d_{22} - d_{11}}{2d_{12}} \right)^2 + 1}.$$

Only one of the two signs gives a value of k that makes the solution for x_c real-valued (the proof follows easily from the geometry of the problem). When $d_{12} = 0$, we instead have either

$$\left(x_c = 0, y_c = \pm \sqrt{-\frac{2}{d_{22}}} \right) \quad \text{if } d_{22} < 0$$

or

$$\left(y_c = 0, x_c = \pm \sqrt{-\frac{2}{d_{11}}} \right) \quad \text{if } d_{11} < 0.$$

Now consider the two spurious surfaces that are obtained by substituting the two values of α ,

$$\alpha = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}},$$

into the equations

$$\tilde{d}_{11} = \frac{t_2}{\tilde{t}_2} d_{11}, \quad \tilde{d}_{12} = -\frac{1}{2\tilde{t}_2} (t_1 d_{11} + \frac{1}{\alpha} t_2 d_{22}), \quad \text{and} \quad \tilde{d}_{22} = \frac{t_1}{\alpha \tilde{t}_2} d_{22},$$

where

$$\tilde{t}_2 = \frac{1}{2} \left(\frac{1}{\alpha} t_1 d_{22} + t_2 d_{11} \right).$$

For each spurious surface, the point on its conic section that is closest to the origin is given by equations similar to the ones given for the true solution. When $\tilde{d}_{12} \neq 0$, we have

$$\left(\tilde{x}_c = \pm \sqrt{\frac{-2}{\tilde{d}_{22}\tilde{k}^2 + 2\tilde{d}_{12}\tilde{k} + \tilde{d}_{11}}}, \tilde{y}_c = \tilde{k}\tilde{x}_c \right)$$

and

$$\tilde{k} = \left(\frac{\tilde{d}_{22} - \tilde{d}_{11}}{2\tilde{d}_{12}} \right) \pm \sqrt{\left(\frac{\tilde{d}_{22} - \tilde{d}_{11}}{2\tilde{d}_{12}} \right)^2 + 1}.$$

When $\tilde{d}_{12} = 0$, we instead obtain either

$$\left(\tilde{x}_c = 0, \tilde{y}_c = \pm \sqrt{-\frac{2}{\tilde{d}_{22}}} \right) \quad \text{if } \tilde{d}_{22} < 0$$

or

$$\left(\tilde{y}_c = 0, \tilde{x}_c = \pm \sqrt{-\frac{2}{\tilde{d}_{11}}} \right) \quad \text{if } \tilde{d}_{11} < 0.$$

The distance from the origin to this point is $l_c = \sqrt{\tilde{x}_c^2 + \tilde{y}_c^2}$. If $l > \tilde{l}_c$, then the image will include some parts of the conic section of the spurious solution. Assuming that we have somehow determined the boundaries of the true surface from the image data, only the conic section for the correct solution can be consistent with the depth discontinuity boundaries. If $l < \tilde{l}_c$, the depth values obtained using the equation of a spurious solution are positive everywhere in the image and so the image will not include any part of the corresponding conic section. Again, this will be inconsistent with the fact that the image should include some part of the conic section of the true surface. It is only when $l \leq \min(l_c, \tilde{l}_{c_1}, \tilde{l}_{c_2})$ that we cannot distinguish among different solutions. In this case, the depth values are positive everywhere in the image for each solution.

Example: Consider a viewer that has a translational velocity $\mathbf{t} = [2, 2, 0]^T$ with respect to surface S defined by

$$d = 1 + \frac{1}{2}x^2 + \frac{5}{3}xy + \frac{1}{2}y^2.$$

Using the equations given earlier, the spurious solutions are

(1) an observer moving with velocity $\tilde{\mathbf{t}}_1 = [-2, \frac{2}{3}, 0]^T$ and $\tilde{\omega}_1 = [-\frac{4}{3}, 4, 0]^T$ with respect to surface \tilde{S}_1 given by

$$d = 1 + \frac{3}{2}x^2 - xy - \frac{1}{2}y^2$$

and (2) an observer moving with velocity $\tilde{\mathbf{t}}_2 = [\frac{2}{3}, -2, 0]^T$ and $\tilde{\omega}_2 = [-4, \frac{4}{3}, 0]^T$ with respect to surface \tilde{S}_2 given by

$$d = 1 - \frac{1}{2}x^2 - xy + \frac{3}{2}y^2.$$

The conic sections that are the images of points at infinity for the three surfaces are shown in Figure 6-1. The asymptotic lines and the direction of translation form the characteristic lines of motion-surface ambiguity. For each solution, the regions of negative depth values are shown by hatched lines.

As explained earlier, the difference of the rotational vectors for any two solutions is aligned with the direction of translation of the third surface. For example, we have

$$\tilde{\omega}_1 - \tilde{\omega}_2 = \frac{4}{3}\mathbf{t}.$$

A needle-map diagram of the resulting motion given by

$$\mathbf{r}_t = \begin{pmatrix} -2 - x^2 - \frac{10}{3}xy - y^2 \\ -2 - x^2 - \frac{10}{3}xy - y^2 \\ 0 \end{pmatrix}$$

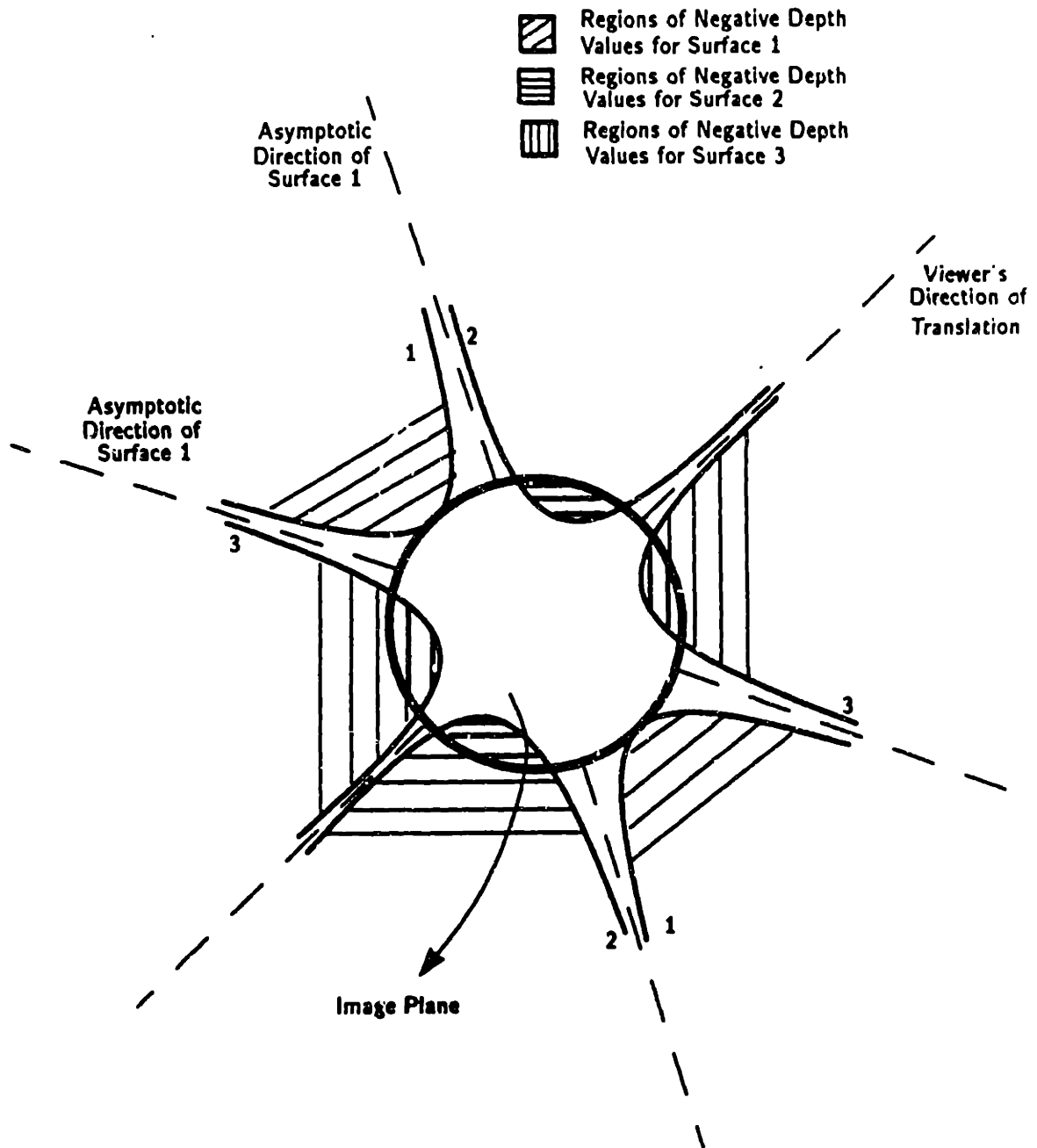


Figure 6-1. The direction of translation and the two asymptotic lines of the surface form the characteristic lines of motion-structure ambiguity. The ambiguity in the interpretation of a motion field can be resolved using a large field of view.

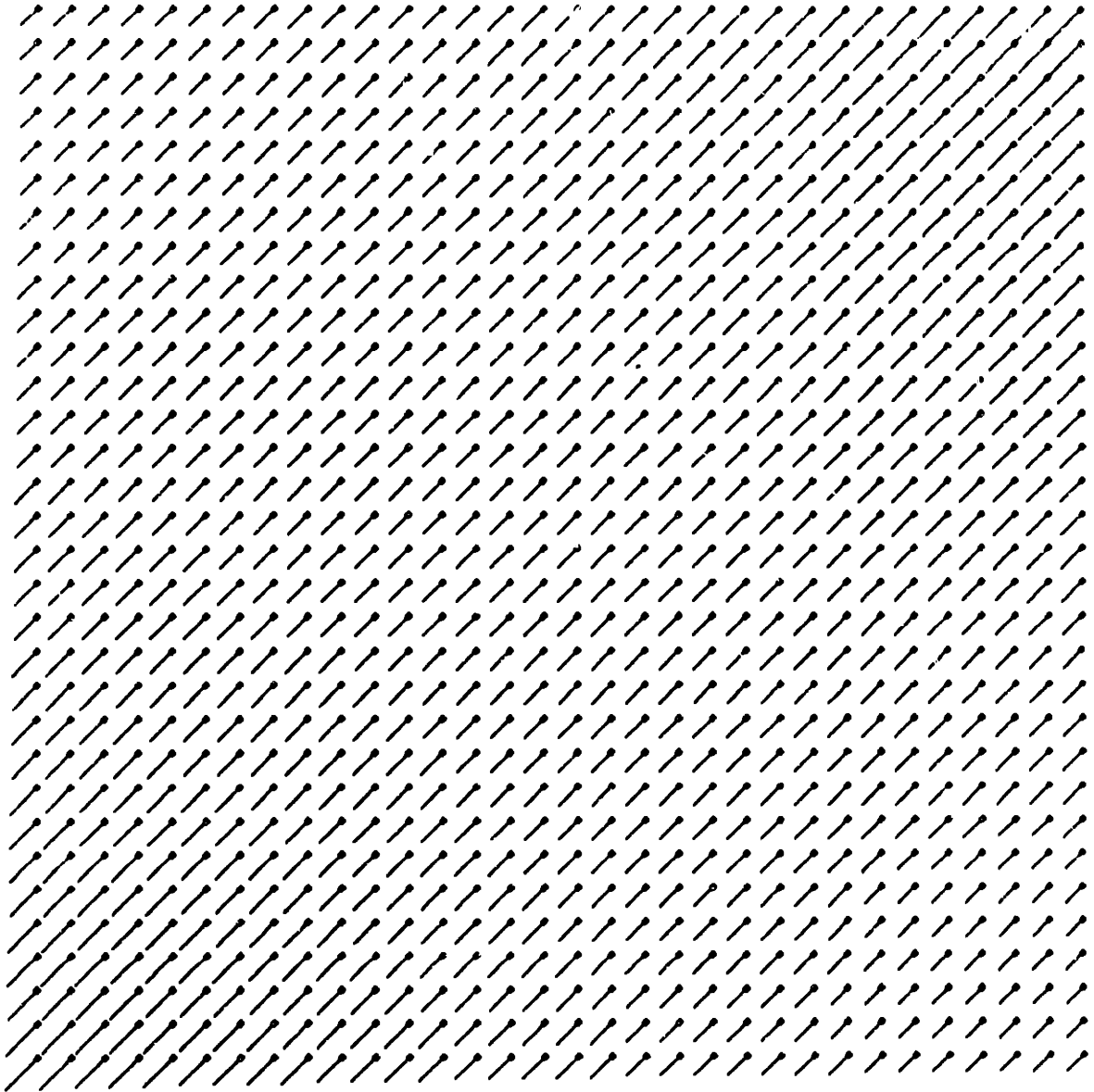


Figure 6-2. A motion field with three rigid body motion interpretations.

is shown in Figure 6-2 (a $\tan^{-1} 0.5 \approx 27$ degrees half-angle field of view is assumed). As expected, the velocity vectors are all parallel emanating from the FOE at infinity and the slope of the vectors is that of the translational vector. The two spurious solutions involve a viewer rotating about an axis parallel to one of the asymptotic directions and translating parallel to the other asymptotic direction of the true surface. This is quite counter-intuitive since the motion field suggests that the solution involves only the translation of the observer.

Now, suppose the image is circular (as shown in Figure 6-1) with a radius of $\sqrt{3}$ (this is the distance from the origin to the closest point on the conic section of the true surface); that is, the half-angle field of view is 60 degrees. This guarantees that we capture as much of the positive region for the true surface as possible without including its conic section; thus, there are no boundaries of depth discontinuity. Then the depth values for \tilde{S}_1 and \tilde{S}_2 are negative in some regions near the image boundaries. This, in theory, precludes both spurious solutions from being a possible solution. If the field of view is more than 60 degrees, then the image will include some parts of the conic section for the true surface which can also be used to distinguish the true solution from the spurious solutions. When the half-angle field of view is less than 50 degrees, the depth values are positive over the whole image for every solution. In this case, the ambiguity cannot be resolved unless other information is available.

6.9 Some Experimental Results

In this section, our goals are

- (1) to present examples of the ambiguities in motion vision (As explained earlier, these ambiguities occur in the case of planar surfaces and certain types of quadratic surfaces. In this section, we consider only quadratic surfaces.)
- (2) to study, in more detail, the behavior of the iterative method given in Chapter 5, and
- (3) to show the difficulties that arise, in the presence of noise in the data, when we use the procedure given for deriving the uniqueness results in order to obtain a solution in closed form (we call this the analytical method).

We have used synthetic images since then the underlying motion is known precisely. Further, in the presence of noise, we can determine the correlation between the (percentage of) noise and the (percentage) degradation from the correct solution. The perfect and noisy data were generated as described in Chapters 4 and 5.

6.9.1 An Example of a Three-Fold Ambiguity

The motion parameters are given by $\omega = [0, 0, 0]^T$ and $t = [1, 2, 0]^T$. The normal of the tangent plane is $n = [0, 0, 1]^T$ and the curvature parameters are given by $d = [1, 10, 1]^T$.

The equation of the surface is given by

$$d = 1 + 0.5 x^2 + 10 xy + 0.5 y^2.$$

Figure 6-3a shows the conic section

$$1 + 0.5 x^2 + 10 xy + 0.5 y^2 = 0$$

which is the locus of points on the surface at infinity. This conic section, which is a hyperbola, separates regions of positive and negative (hatched region) depth values. The two spurious solutions are

$$\tilde{\omega}_1 = [-10.97, 0.5501, 0]^T, \quad \tilde{t}_1 = [0.4499, -8.975, 0]^T,$$

$$\tilde{n}_1 = [0, 0, 1]^T, \quad \tilde{d}_1 = [-0.2228, -2.167, 2.223]^T,$$

and

$$\tilde{\omega}_2 = [-1.025, 20.45, 0]^T, \quad \tilde{t}_2 = [-19.45, 0.9750, 0]^T,$$

$$\tilde{n}_2 = [0, 0, 1]^T, \quad \tilde{d}_2 = [2.051, -0.4614, -0.0514]^T.$$

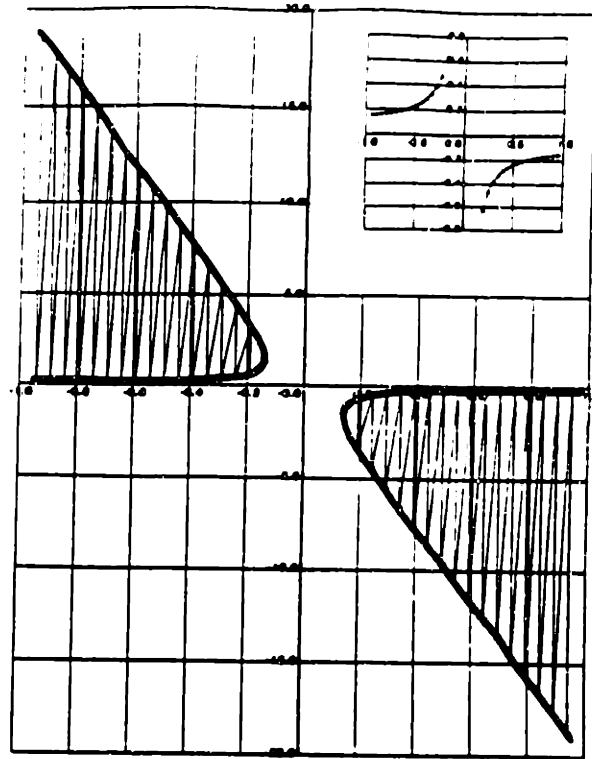
The spurious surfaces are given by

$$d = 1 - .11 x^2 - 2.17 xy + 1.11 y^2 \quad \text{and} \quad d = 1 + 1.03 x^2 - .46 xy - .03 y^2.$$

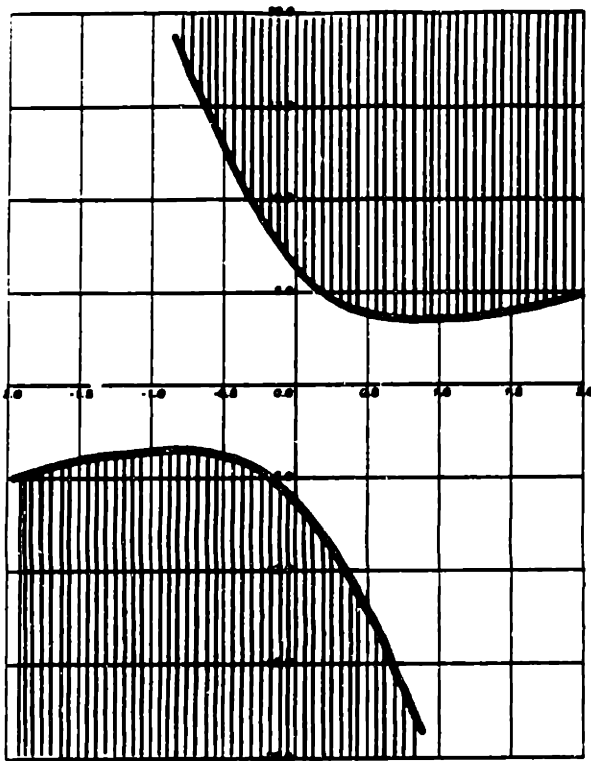
Figures 6-3b and 6-3c show the regions of positive and negative (hatched region) depth values for the two spurious solutions and Figure 6-3d shows the same for the three surfaces. The apices of the conic section for the true surface are located at $\pm(1/3, -1/3)$. If the field of view is equal to or less than $\tan^{-1}(1/3) \approx 18$ degrees, then the image will consist of only one object whose surface is given by the equation of the true surface (provided that no part of the object is occluded); however, we cannot distinguish between the three solutions since the depth values are positive over the image for the three solutions. If the field of view is larger than 18 degrees, we can distinguish the correct solution as follows. Assuming that we have determined the boundaries of the true surface in the image, these should be consistent with the fact that the object does not extend beyond a 18 degrees field of view (in the second and fourth quadrants). In this case, the image will include other objects (or background) that are (is) imaged into the regions with negative depth values. If the true surface is one of the two given by the spurious solutions, then the same object has to extend beyond the 18 degrees field of view; therefore, we should not observe any boundaries of depth discontinuity.

The resulting motion field, shown in Figure 6-4 (for a 27 degrees half-angle field of view), is given by

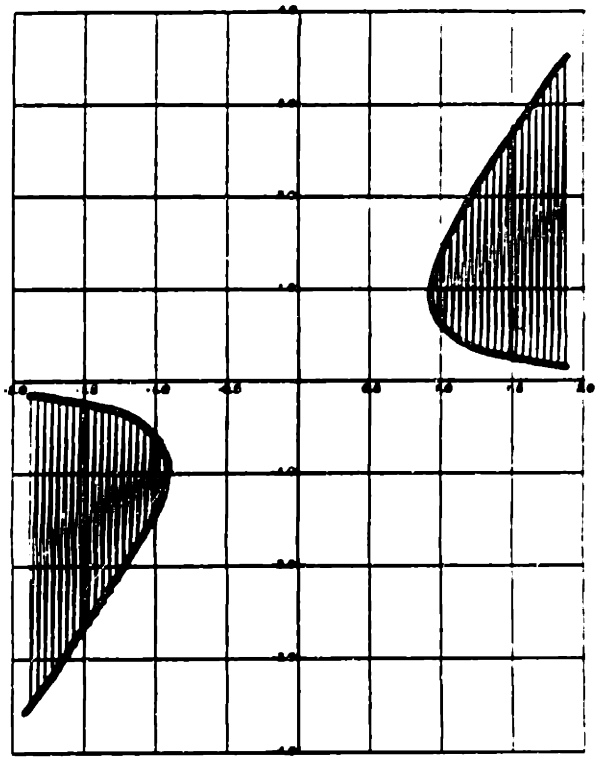
$$\mathbf{r}_t = \begin{pmatrix} -1 - 0.5 x^2 - 10 xy - 0.5 y^2 \\ -2 - x^2 - 20 xy - y^2 \\ 0 \end{pmatrix}.$$



(a)



(b)



(c)

Figure 6-3. The images of the boundaries of three hyperboloids of one sheet responsible for a three-fold ambiguity. The ambiguity in a motion field is usually restricted to small region of the image. For a sufficiently large field of view, the ambiguity can be resolved by matching the depth discontinuity boundaries with the conic section of the true surface.

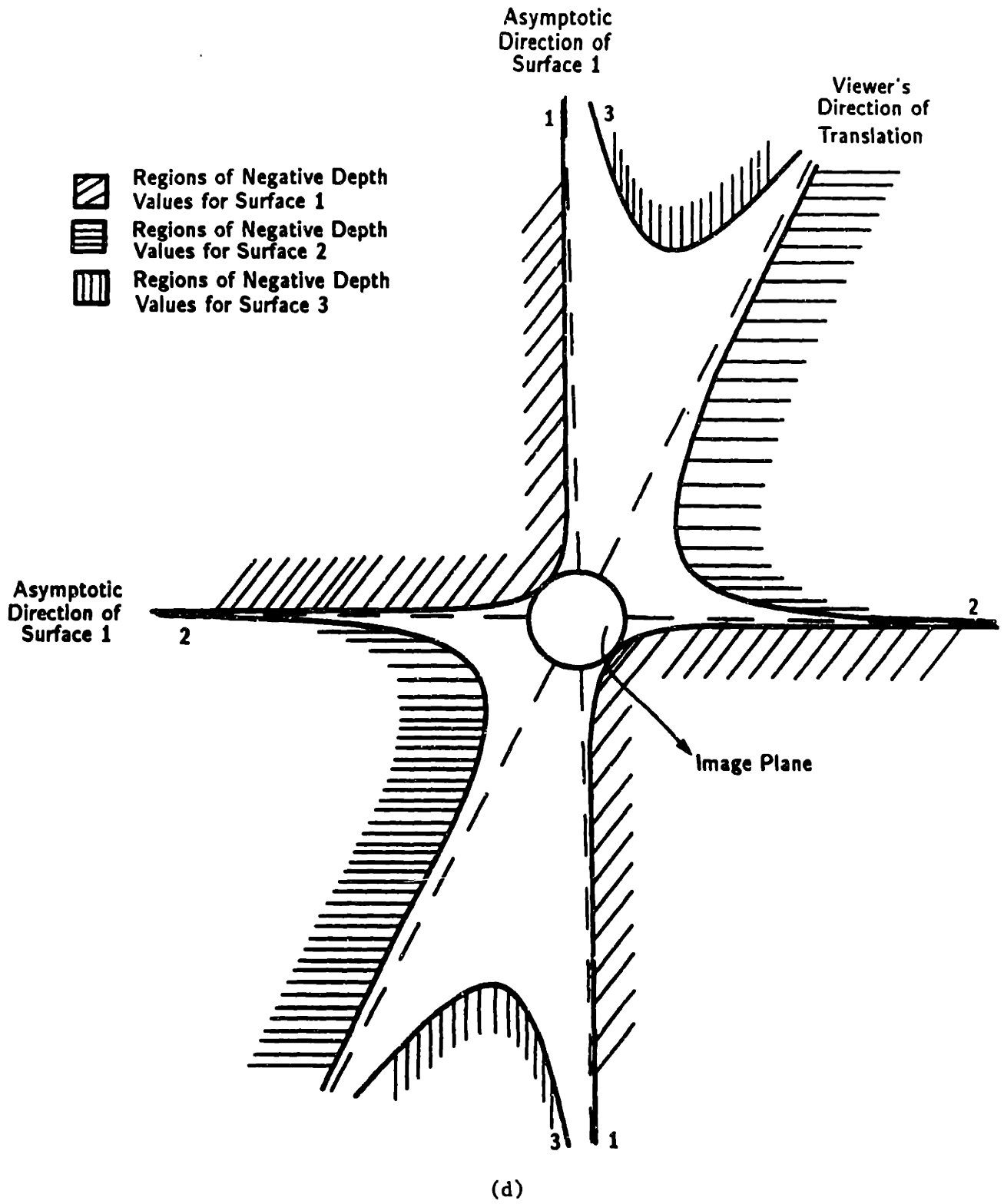


Figure 6-3. Continued.

As shown by the above equation, the three-fold ambiguity is characterized by a second-order motion field. Since the motion is a translation parallel to the image plane, the flow vectors are parallel. Note that the velocity vectors emanate from the focus of expansion at infinity in the region where depth values are positive (the region including the center, the bottom-left and the top-right of the image). In the bottom-right and the top-left of the image plane, these vectors point in the opposite direction. This implies that either the motion is in the opposite direction (to the true one) or the depth values are negative. In either case, this region cannot be the image of part of the same object that is imaged into region with positive depth values.

6.9.1.1 Simulation Results

Table 6.1a gives the results of several simulations of the iterative method after 100 iterations using random initial conditions. In every simulation, the algorithm either converged to a solution near the true solution or a solution near the first spurious solution (these are not included here), but it never converged to the second spurious solution; remember that after a solution is obtained, the other two can be determined using the expressions given in the previous sections. Note that, with noisy data, the error for the optimal solution is less than that for the true solution; this is expected since it is unlikely that the true solution is the optimum one when data are noisy. These results show that a good estimate of the solution can be determined even with as much as 28% noise in the data.

Table 6.1b shows the solutions obtained using the analytical method (that is, using the essential parameters). These solutions were obtained assuming that it was known a priori that $t_3 = 0$. This was necessary since, in the case $t_3 = 0$, the constraint $m'_{11} = 0$ has to be satisfied for a possible solution (m'_{11} is one of the two coefficients of the linear terms of the x -component of the motion field after the transformation described in the previous sections). In the presence of noise, the solution is not expected to satisfy this constraint exactly. In practice, we may need to set a threshold that has to be adjusted to account for the amount of noise in the data. Also, when there is a three-fold ambiguity, the constraint $m'_{111} = 0$ has to be satisfied. This is one of the coefficients of the quadratic terms of the x -component of the motion field after transformation. In our simulations, both constraints were violated with less than 6% noise in the data. For example, with an average noise of 5.7%, $m'_{111} = 0.65$ for the true solution, $m'_{111} = 0.88$ for the first spurious solution, and $m'_{11} = -.09$ for the second spurious solution. With 14.2% noise, $m'_{11} = -.11$ and $m'_{111} = 1.04$ for the true solution, $m'_{11} = -.11$ and $m'_{111} = .73$ for the first spurious solution, and $m'_{11} = -.18$ and $m'_{111} = -.75$ for the second spurious solution.

Alternatively, we may determine two sets of solutions one assuming that $t_3 = 0$ and one for $t_3 \neq 0$ (remember that each case requires a different solution procedure). This may be nine solutions in the most general case; two solutions assuming that $t_3 \neq 0$ and one solution assuming that $t_3 = 0$ for each of the three possible transformation angles. Then, we somehow have to determine the correct solution which, in general, may not be

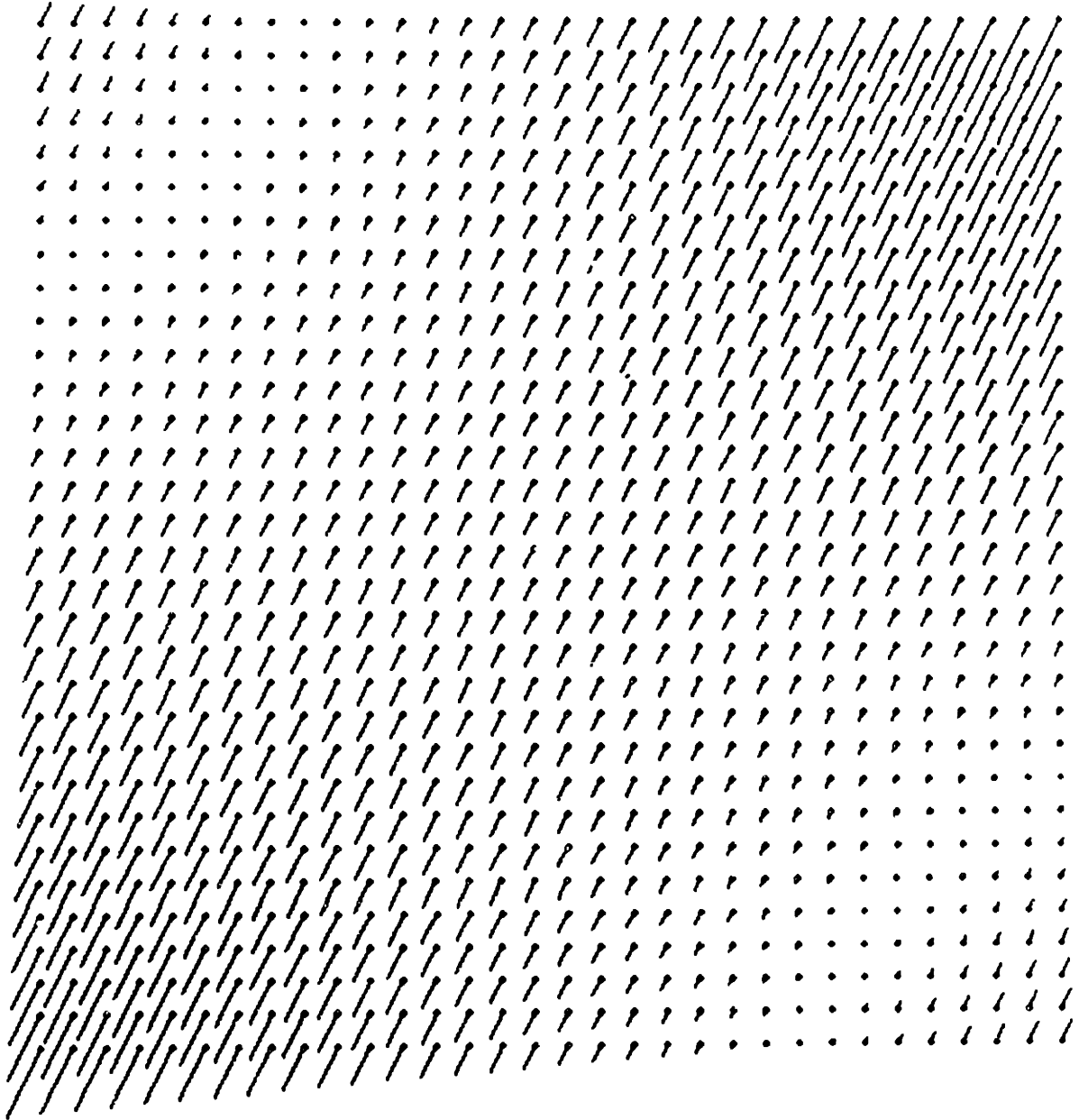


Figure 6-4. A motion field with three rigid body interpretations in a local region. The discontinuity in the motion field is consistent with the depth discontinuity boundaries of the true surface.

an easy task. For example, it is not possible to distinguish between the true solution and the spurious solutions based on the value of the error. This is easy to see if we consider the results obtained for different noise levels. In every case, the solution near the true solution has a larger error than either of the two spurious solutions.

A comparison of the results of the iterative and analytical methods shows that the we obtain a better estimate from the iterative method than from the analytical method when data are noisy.

6.9.2 An Example of a Two-Fold Ambiguity

The possibility of a two-fold ambiguity in the solution of the quadratic surface is demonstrated through the following example. The motion parameters are given by $\omega = [0, 0, 0]^T$ and $\mathbf{t} = [1, 2, 0]^T$. The normal of the tangent plane is $\mathbf{n} = [-.0501, 1, 1]^T$ and the curvature parameters are given by $\mathbf{d} = [1, 10, 1]^T$.

For the spurious solution, we obtain

$$\tilde{\omega} = [-10.97, .5501, -1.100]^T, \quad \tilde{\mathbf{t}} = [.4499, -8.975, 0]^T,$$

$$\tilde{\mathbf{n}} = [-.1114, -.2228, 1]^T, \quad \tilde{\mathbf{d}} = [-.2228, -2.167, 2.223]^T.$$

The equation of the true and spurious surfaces, respectively, are

$$d = 1 - .05 x + y + .5 x^2 + 10 xy + 0.5 y^2$$

and

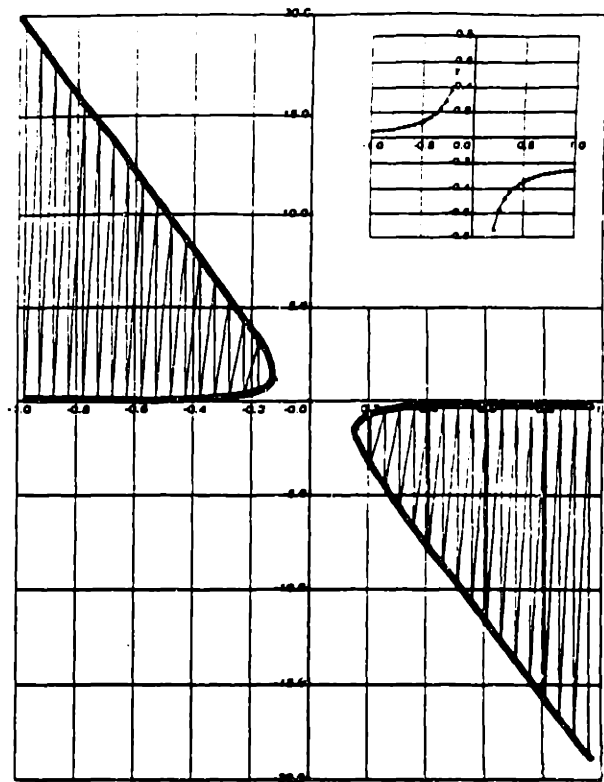
$$d = 1 - .11 x - .22 y - .11 x^2 - 2.17 xy + 1.11 y^2.$$

The corresponding conic sections are shown in Figures 6-5a and 6-5b (the hatched regions show areas where depth is negative). Again, it is possible to distinguish the correct solution if the field of view is sufficiently large.

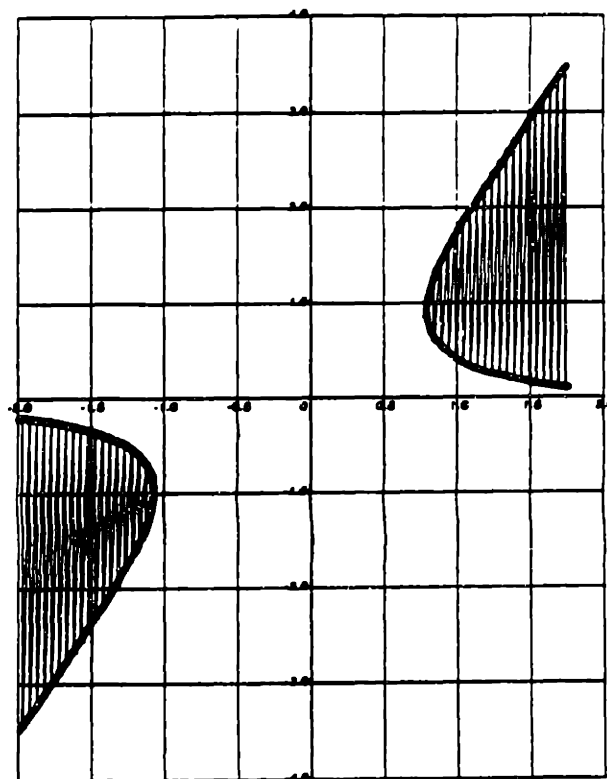
The resulting motion field, shown in Figure 6-5c (for a 27 degrees field of view), is given by

$$\mathbf{r}_t = \begin{pmatrix} -1 + .05 x - y - 0.5 x^2 - 10 xy - 0.5 y^2 \\ -2 + .1 x - 2 y - x^2 - 20 xy - y^2 \\ 0 \end{pmatrix}.$$

As expected, the motion field is a second-order flow field and consists of parallel vectors that either emanate from the focus of expansion in the region with positive depth values or point toward the focus of expansion in the region with negative depth values.



(a)



(b)

Figure 6-5. The boundaries of discontinuity for two hyperboloids of one sheet that give rise to the same motion field when viewed by an observer moving perpendicular to the viewing direction

6.9.2.1 Simulation Results

Table 6.2a shows the results of several simulations of the iterative scheme. We have not included those simulations that converged to the spurious solution. These results show that good estimates can be obtained with as much as 30% noise in the data. The inaccuracies in the estimates of the rotational motion can be attributed to the inherent ambiguity in the interpretation of 3-D motion in the presence of noisy data; this consists of interpreting a rotation about positive x -axis (or positive y -axis) as a translation along the negative y -axis (or positive x -axis) scaled by the distance of the object from the viewer. More precisely, the ambiguity is such that

$$\delta\omega_1 \approx d_0\delta t_2 = \delta t_2 \quad \text{and} \quad \delta\omega_2 \approx -d_0\delta t_1 = -\delta t_1,$$

where $\delta(\cdot)$ denotes the error in the estimate of (\cdot) . For example, with 5.7% noise in the data, we have

$$\delta\omega_1 = 0 - (-.1915) = .1915, \quad \delta t_2 = (2 - 1.7705) = .2295,$$

$$\delta\omega_2 = 0 - .1271 = -.127, \quad \delta t_1 = (1 - .8694) = .1304.$$

Similarly, with 13.8% noise in the data, we obtain

$$\delta\omega_1 = 0 - .2761 = -.2761, \quad \delta t_2 = (2 - 2.258) = -.258,$$

$$\delta\omega_2 = 0 - (-.1717) = .1717, \quad \delta t_1 = 1 - 1.1534 = -.1537.$$

We should mention that for some initial conditions, the algorithm converged to a solution near

$$\boldsymbol{\omega} = [-1.1164, 20.27, -0.0410]^T, \quad \mathbf{t} = [-19.31, 0.8981, -1.418]^T,$$

$$\mathbf{n} = [0.1177, -0.0597, 1.000]^T, \quad \mathbf{d} = [2.053, -0.4612, -0.0508]^T,$$

that seems to be a local minimum.

Table 6.2b shows the results obtained using the analytical method. Again, we had to assume that $t_3 = 0$ in order to use the correct solution procedure (as explained in the previous example). With noisy data, the constraint $m'_{11} = 0$ was violated in every case (we used a threshold of .1 for 5.7% noise, .25 for 13.8% noise, and .5 for 28.8% noise). The results, with the exception of the estimates of t_3 , are comparable to those obtained from the iterative method; yet, the magnitude of the total error is much larger due to bad estimates of t_3 . Note that the magnitude of the error for the true solution is consistently larger than that for the spurious solution. Interestingly, with 28.1% noisy in the data, a third solution is obtained near the local minimum that was observed in the simulation results of the iterative method. In fact, this solution has a smaller error than the true solution.

6.9.3 The Ambiguity of a Second-Order Flow Analysis: Example One

We now present an example of an ambiguity that arises when we restrict ourselves to a second-order analysis of the flow field. The motion parameters are given by $\omega = [0, 0, 0]^T$ and $t = [1, 2, 3]^T$. The normal of the tangent plane is $n = [0, 0, 1]^T$ and the curvature parameters are given by $d = [1, 10, 1]^T$.

The resulting motion field, shown in Figure 6-6a for a 27 degrees field of view, is given by

$$r_t = \begin{pmatrix} -1 + 3x - 0.5x^2 - 10xy - 0.5y^2 + 1.5x^3 + 30x^2y + 1.5xy^2 \\ -2 + 3y - x^2 - 20xy - y^2 + 1.5x^2y + 30xy^2 + 1.5y^3 \\ 0 \end{pmatrix}.$$

Since the motion field is cubic, we immediately conclude that the solution is unique. If we only exploit the second order terms of the motion field, we find that two spurious solutions are possible. These are

$$\tilde{\omega}_1 = [-1.025, 20.45, 0]^T, \quad \tilde{t}_1 = [-19.45, .9750, 3]^T,$$

$$\tilde{n}_1 = [0, 0, 1]^T, \quad \tilde{d}_1 = [2.051, -.4614, -.0514]^T,$$

and

$$\tilde{\omega}_2 = [-10.97, .5501, 0]^T, \quad \tilde{t}_2 = [.4499, -8.975, 3]^T,$$

$$\tilde{n}_2 = [0, 0, 1]^T, \quad \tilde{d}_2 = [-.2228, -2.167, 2.223]^T.$$

For the first spurious solution, the motion field is given by

$$\tilde{r}_t = \begin{pmatrix} -1 + 3x - 0.5x^2 - 10xy - 0.5y^2 + 3.08x^3 - 0.69x^2y - 0.08xy^2 \\ -2 + 3y - x^2 - 20xy - y^2 + 3.08x^2y - 0.69xy^2 - 0.08y^3 \\ 0 \end{pmatrix}$$

which is shown in figure 6-6b. Similarly, we obtain

$$\tilde{r}_t = \begin{pmatrix} -1 + 3x - 0.5x^2 - 10xy - 0.5y^2 + 0.33x^3 - 3.25x^2y + 3.33xy^2 \\ -2 + 3y - x^2 - 20xy - y^2 + 0.33x^2y - 3.25xy^2 + 3.33y^3 \\ 0 \end{pmatrix}$$

for the second spurious solution. This is shown in Figure 6-6c. In fact, the solution is unique since the cubic terms of the motion field are different for the three solutions; however, the three motion fields are indistinguishable if we restrict ourselves to a small field of view (a small region in the center of the image).

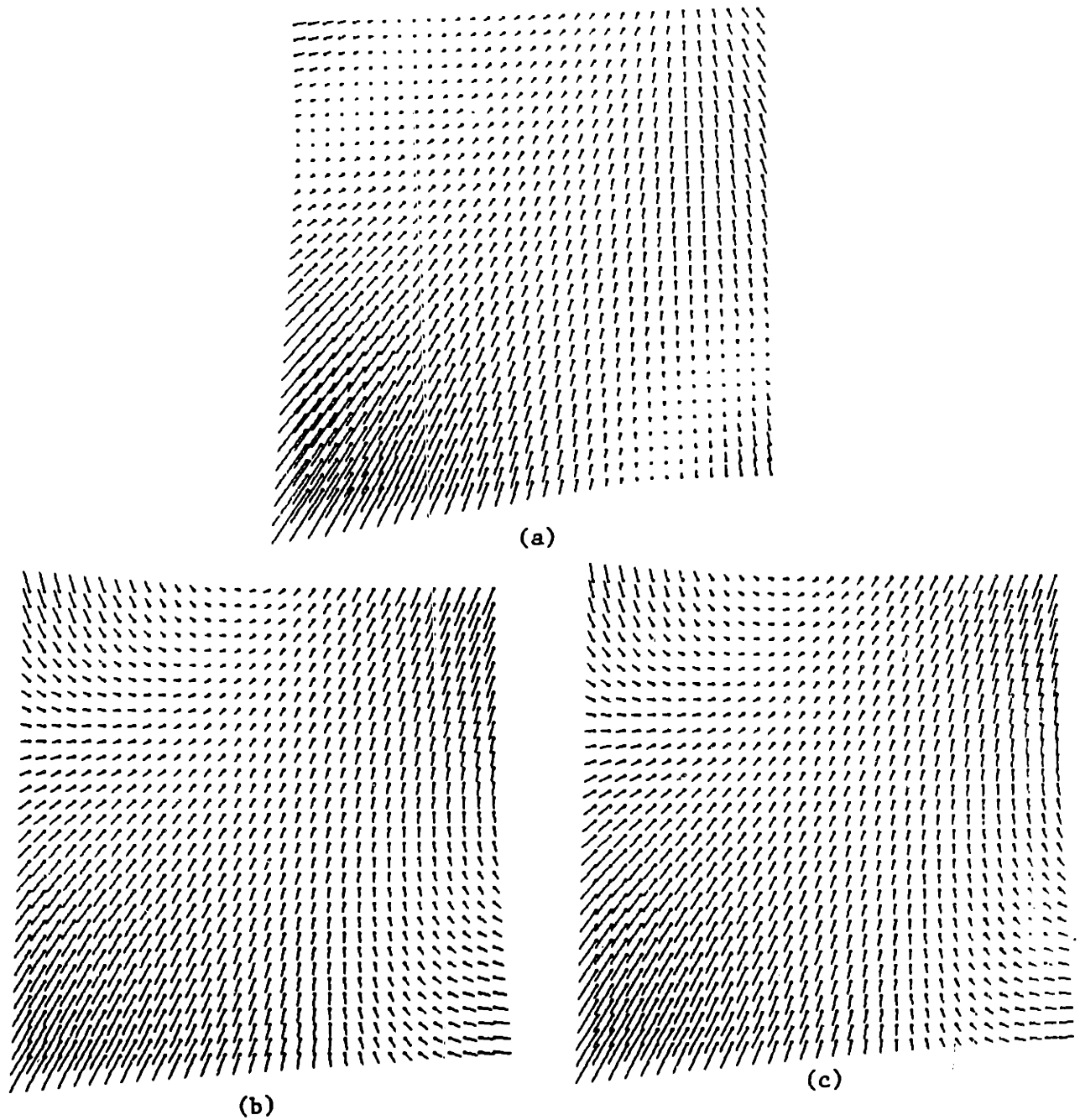


Figure 6-6. The ambiguity in the interpretation of a motion field is usually restricted to local regions. Three surfaces undergoing different motions can give rise to motion fields that are similar up to second-order terms.

6.9.3.1 Simulation Results

The results of several simulations of the iterative scheme and the analytical method are given in Tables 6.3a and 6.3b. These results show that we obtain better estimates from the iterative scheme; however, in the presence of noise, the solution is not as satisfactory as in the previous two cases even though the convergence is faster. For example, the component of rotation about the optical axis cannot be determined accurately even with a large field of view. In addition, the results deteriorate when the noise is increased to about 30%. The error is as large as 50% in certain components, for example in d_{11} , d_{22} , ω_2 , ω_3 and t_1 . Actually, the errors in ω_2 and t_1 are due to the ambiguity in the interpretation of translation along positive x -axis as rotation about positive y -axis scaled appropriately by the distance of the object from the viewer ($\delta\omega_2 = -d_0\delta t_1$). For example, with 28.4% noise in the data, we have

$$\delta\omega_2 = 0 - .4482 = -.4482, \quad \text{and} \quad \delta t_1 = 1 - .5391 = .4609.$$

6.9.4 The Ambiguity of a Second-Order Flow Analysis: Example Two

Another ambiguity that arises when we restrict ourselves to a second-order analysis of the motion field is shown through the following example. The motion parameters are given by $\omega = [0, 0, 0]^T$ and $\mathbf{t} = [1, 2, 5]^T$. The normal of the tangent plane is $\mathbf{n} = [1, 2, 1]^T$ and the curvature parameters are given by $\mathbf{d} = [1, 10, 1]^T$. The resulting motion field is a cubic polynomial given by

$$\mathbf{r}_t = \begin{pmatrix} -1.0 + 4.0x - 2.0y + 4.5x^2 - 0.5y^2 + 2.5x^3 + 50.0x^2y + 2.5xy^2 \\ -2.0 - 2.0x + y - x^2 - 15xy + 9y^2 + 2.5x^2y + 50.0xy^2 + 2.5y^3 \\ 0 \end{pmatrix}.$$

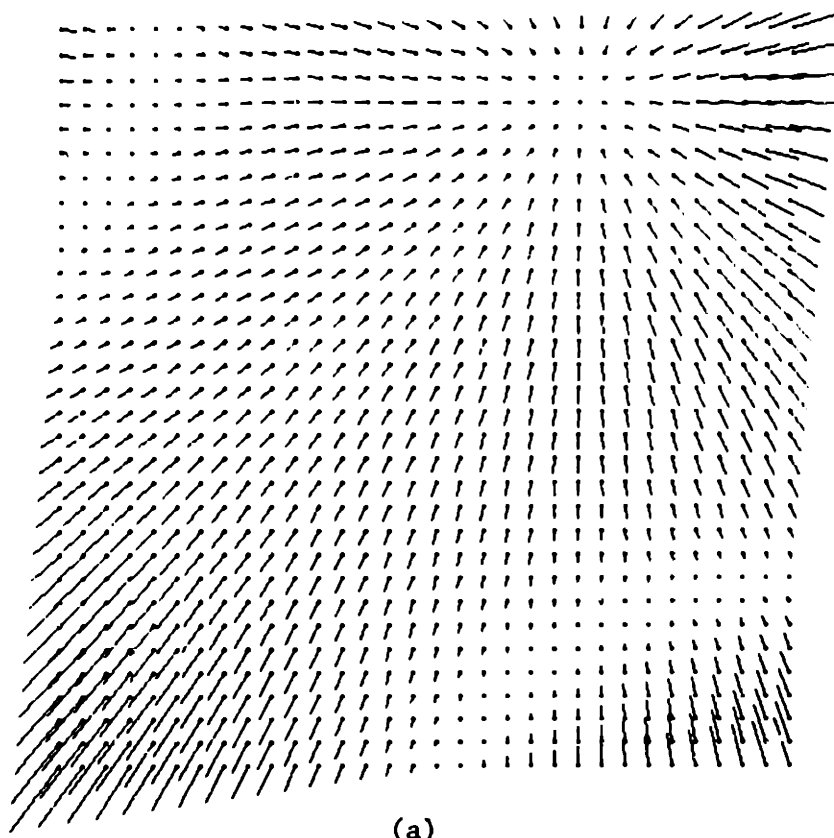
Figure 6-7a shows a needle-map diagram of the motion field with a 27 degrees half-angle field of view. If we ignore the cubic terms of the flow field, we find that there is a spurious solution,

$$\tilde{\omega} = [8, -4, 0]^T, \quad \tilde{\mathbf{t}} = [5, 10, 5]^T, \quad \tilde{\mathbf{n}} = [.2, .4, 1]^T, \quad \text{and} \quad \tilde{\mathbf{d}} = [.2, 2, .2]^T.$$

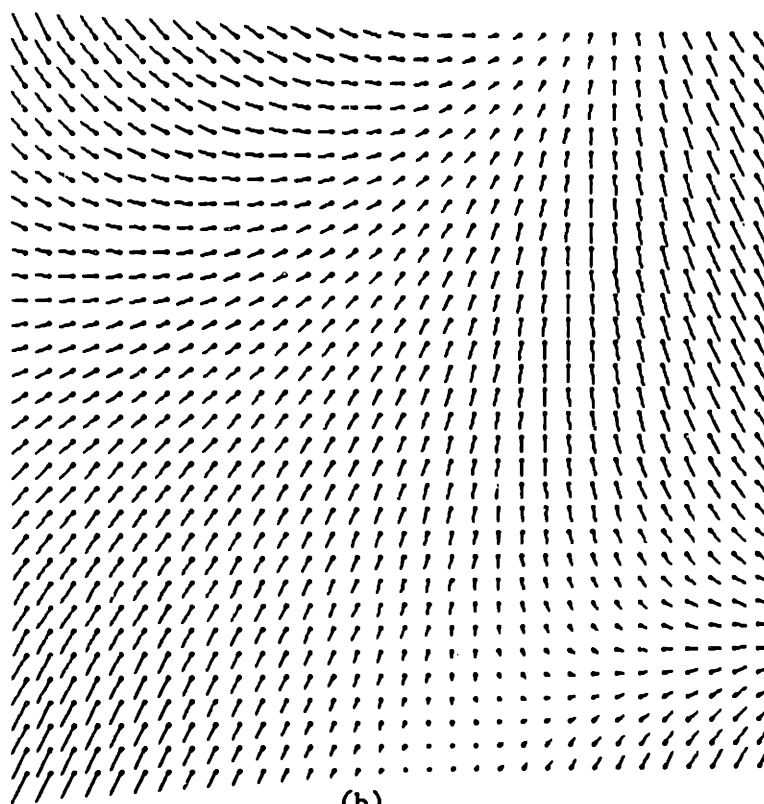
The motion field for the spurious solution, shown in Figure 6-7b, is given by

$$\mathbf{r}_t = \begin{pmatrix} -1.0 + 4.0x - 2.0y + 4.5x^2 - 0.5y^2 + 0.5x^3 + 10.0x^2y + 0.5xy^2 \\ -2.0 - 2.0x + y - x^2 - 15.0xy + 9.0y^2 + 0.5x^2y + 10.0xy^2 + 0.5y^3 \\ 0 \end{pmatrix}.$$

Again, we conclude that the solution is unique since the spurious solution differs from the true solution in the cubic terms of the motion field. The same conclusion is made when we compare the diagrams that show the two motion fields; however, it is very difficult to distinguish between the two motion fields if the field of view is small.



(a)



(b)

Figure 6-7. An example of a two-fold ambiguity when the analysis is restricted to a local region of the image.

6.9.4.1 Simulation Results

Tables 6.4a and 6.4b show the results of several simulations of the iterative method after 60 iterations and the solutions obtained using the analytical method, respectively. For the iterative method, the estimate of every parameter, with the exception of ω_3 , is within 20% of the true value with as much as 30% noise in the data. For the analytical method, the estimates of the rotational motion are very inaccurate. With noisy data, most of the constraints had to be relaxed to obtain the solution.

6.9.5 Example 5: A Non-ambiguous case

This example demonstrates a situation that has been labeled incorrectly as an ambiguous case in Waxman et al [1986]. The motion parameters are given by $\omega = [0, 0, 0]^T$ and $t = [1, 2, 0]^T$. The normal of the tangent plane is $n = [1, 1, 1]^T$ and the curvature parameters are given by $d = [1, 10, 1]^T$ (that is, the surface has a negative Gaussian curvature, the mean curvature is equal to one, and the translation along the line of sight vanishes). The equation of the surface,

$$d = 1 + x + y + 0.5 x^2 + 10 xy + 0.5 y^2,$$

can be written

$$\frac{1}{2} \mathbf{R}^T \mathbf{D} \mathbf{R} + \hat{\mathbf{z}}^T \mathbf{R} = 0,$$

where

$$\mathbf{D} = \begin{pmatrix} 1 & 10 & 1 \\ 10 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix}, \quad \hat{\mathbf{z}} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}.$$

A second solution, if it exists, can be determined from the eigenvalue-eigenvector decomposition of \mathbf{D} . The eigenvalues of \mathbf{D} are -9, 1.78, and 11.22. Clearly, these do not satisfy the constraint among the eigenvalues of a critical surface pair (the middle eigenvalue should be equal to the sum of the other two eigenvalues). The corresponding eigenvectors are

$$\mathbf{u}_1 = [0.707, -0.707, 0]^T, \quad \mathbf{u}_2 = [0.107, 0.107, -0.988]^T, \quad \text{and} \quad \mathbf{u}_3 = [0.699, 0.699, 0.152]^T.$$

Using the equations given earlier, the possible solutions for $\tilde{\mathbf{t}}$ and $\tilde{\omega}$ are

$$\tilde{\mathbf{t}} = \pm \|\tilde{\mathbf{t}}\| [0.994, -0.049, 0.103]^T, \quad \tilde{\omega} = \pm \|\tilde{\omega}\| [-0.049, 0.994, 0.103]^T,$$

and

$$\tilde{\mathbf{t}} = \pm \|\tilde{\mathbf{t}}\| [-0.049, 0.994, 0.103]^T, \quad \tilde{\omega} = \pm \|\tilde{\omega}\| [0.994, -0.049, 0.103]^T.$$

None of the four solutions can be a possible motion for a second solution since the translational vector have a component along the optical axis (we showed that the component of translation along the optical axis should be the same for the true and spurious solutions; that is, it should be zero).

6.9.5.1 Simulation Results

The simulation results using the iterative and analytical methods are given in Tables 6.5a and 6.5b. The iterative results were obtained using random initial conditions and after 100 iterations of the algorithm. Most of the parameters have been estimated rather accurately (these are within 20% of the true value), except for the curvature parameters (when the noise is increased to about 30%). The most accurate estimates are those obtained for the components of motion in the direction of the optical axis; this is usually the case when a relatively large field of view is assumed. Using the analytical method, the constraints $m'_{11} = 0$ and $m'_{111} = 0$ had to be relaxed to discount the noise effect. With 28.4% noise, three solutions are obtained; however, it is very difficult to determine which solution corresponds to the true solution since the estimates are very inaccurate.

6.10 Summary

We have addressed the issue of the uniqueness of the solution in motion vision. We have shown that an ambiguity in the solution of the motion vision problem can be at most a three-fold one. Extending the results derived by Horn [1986b] that we summarized in a theorem referred to as *Horn's theorem of critical surfaces* we have derived analytical expressions for the relationship among multiple solutions.

Using a special representation of smooth curved surfaces, when the inverse depth function is a polynomial in the image coordinates, we have shown that only certain quadratic surfaces can give rise to an ambiguity; no other smooth surface can give rise to an ambiguity in the interpretation of 3-D motion. Again, the ambiguity is at most three-fold and is characterized by an observer moving in parallel to the image plane (perpendicular to the viewing direction). For example, a three-fold ambiguity arises if the surface gradient vanishes, the Gaussian curvature is negative, and the mean curvature is equal to one at the fixation point. In this case, the surface is a hyperboloid of one sheet. The ambiguity degenerates to a two-fold one when the Gaussian curvature is zero with all other conditions being the same; in this case, the surface is a circular cylinder. There is also a two-fold ambiguity when the Gaussian curvature is non-positive, the mean curvature is unity, and the surface normal, the optical axis, and one of the asymptotic directions (which is the only one if the Gaussian curvature is zero) are in the same plane at the fixation point. Again, the surface is a hyperboloid of one sheet.

In several examples of ambiguous cases, we showed a counter-intuitive behavior of the resulting motion field where the image velocity vectors are all parallel. This has the intuitive interpretation that an observer is translating parallel to the image plane in addition to two other non-intuitive interpretations that involve both rotation and translation of the viewer. These two interpretations are characterized by a rotation about one of the asymptotic directions and a translating along the other asymptotic direction of the intuitive solution. The corresponding surface has two asymptotic lines:

one is parallel to the true direction of translation and the other is parallel to the axis of rotation of the observer viewing the spurious surface. We showed that when an ambiguity exists, it is possible to distinguish the correct solution using the fact that an acceptable solution should give rise to positive depth values over the relevant region of the image. To do this, we need to have a large field of view. We derived a mathematical expression for the minimum size of the field of view.

Finally, we presented selected examples to compare the behavior of the solution of the iterative scheme given in Chapter 5 and the analytic method (this is, the method used to present the uniqueness results) in the presence of noise in the data. We concluded that the iterative scheme gives better estimates in every case.

Table 6.1a: Simulation Results of the Iterative Scheme after 120 Iterations

True Solution	Spurious Solution (1)	Spurious Solution (2)
$\omega = [0, 0, 0]^T$	$\omega = [-10.97, 0.5501, 0]^T$	$\omega = [-1.025, 20.45, 0]^T$
$t = [1, 2, 0]^T$	$t = [0.4499, -8.975, 0]^T$	$t = [-19.45, 0.9750, 0]^T$
$n = [0, 0, 1]^T$	$n = [0, 0, 1]^T$	$n = [0, 0, 1]^T$
$d = [1, 10, 1]^T$	$d = [-0.2228, -2.167, 2.223]^T$	$d = [2.051, -0.4614, -0.0514]^T$

Initial Condition: $n = [0.0062, 0.0428, -0.0486]^T$ $d = [0.0142, -0.0487, 0.0072]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
0%	0%	0	0.003

	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	-0.0290	0.0160	0.0001	0.9836	1.971	-0.0001
Surface Parameters:	-0.0003	-0.0001	1.000	1.009	10.14	0.9918

Initial Condition: $n = [0.0132, 0.0408, -0.0534]^T$ $d = [0.0046, -0.0562, 0.0014]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
10%	5.7%	36.9	34.7

	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	-0.0229	-0.0088	0.0405	0.9929	2.020	-0.0013
Surface Parameters:	-0.0488	-0.0181	1.000	0.8868	9.820	0.7980

Initial Condition: $n = [-0.0248, 0.0108, 1.887]^T$ $d = [2.340, 22.12, 1.842]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
25%	14.2%	188.	186.

	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	0.1040	-0.0689	-0.1055	1.041	2.050	0.0211
Surface Parameters:	-0.0074	0.0053	1.000	1.088	9.761	1.082

Initial Condition: $n = [0.0443, -0.0169, -0.0162]^T$ $d = [-0.0481, -0.0384, -0.0108]^T$

Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error
50%	28.1%	865.	826.

	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	0.3650	-0.2536	0.1152	1.091	2.197	-0.0660
Surface Parameters:	-0.1072	-0.0242	1.000	1.159	8.53	1.301

Table 6.1b: Simulation Results of the Analytical Method

True Solution	Spurious Solution (1)	Spurious Solution (2)
$\omega = [0, 0, 0]^T$	$\omega = [-10.97, 0.5501, 0]^T$	$\omega = [-1.025, 20.45, 0]^T$
$t = [1, 2, 0]^T$	$t = [0.4499, -8.975, 0]^T$	$t = [-19.45, 0.9750, 0]^T$
$n = [0, 0, 1]^T$	$n = [0, 0, 1]^T$	$n = [0, 0, 1]^T$
$d = [1, 10, 1]^T$	$d = [-0.2228, -2.167, 2.223]^T$	$d = [2.051, -0.4614, -0.0514]^T$
Max. Image Noise	S.D. Noise	True Solution Error
10%	5.7%	36.9
Rotation/Surface Normal		Translation/Curvature Parameters Error
Motion Parameters:	-1.087 20.26 -0.0370	-19.20 0.9155 -0.0893 36.6
Surface Parameters:	0.0027 0.0036 1.000	2.074 -0.4533 -0.0296
Motion Parameters:	-0.1736 0.1896 0.0737	0.8698 1.829 0.0035 36.9
Surface Parameters:	-0.0576 0.0509 1.000	0.9123 10.79 0.7610
Motion Parameters:	-10.37 0.8149 0.0238	0.2445 -8.371 0.0546 37.8
Surface Parameters:	0.0062 -0.0110 1.000	-0.2298 -2.271 2.264
Max. Image Noise	S.D. Noise	True Solution Error
25%	14.2%	188.
Rotation/Surface Normal		Translation/Curvature Parameters Error
Motion Parameters:	-10.91 0.7234 -0.1357	0.3726 -9.026 -0.1068 194.
Surface Parameters:	-0.0021 -0.0193 1.000	-0.2690 -2.130 2.177
Motion Parameters:	-0.7378 20.46 -0.1158	-19.37 1.150 -0.1758 197.
Surface Parameters:	-0.0054 -0.0003 1.000	2.081 -0.4827 -0.0395
Motion Parameters:	-0.0272 0.1909 -0.1018	0.9051 1.861 -0.1107 221
Surface Parameters:	0.0180 0.0825 1.000	1.091 10.71 1.091
Max. Image Noise	S.D. Noise	True Solution Error
50%	28.1%	865.
Rotation/Surface Normal		Translation/Curvature Parameters Error
Motion Parameters:	-9.864 0.7240 0.1760	0.3732 -8.166 0.1929 843.
Surface Parameters:	0.0377 0.0145 1.000	-0.3758 -2.178 2.116
Motion Parameters:	-0.2897 19.04 0.1435	-17.94 1.409 0.1447 852.
Surface Parameters:	0.0098 0.0173 1.000	2.125 -0.4841 -0.0354
Motion Parameters:	0.0976 0.2894 -0.1476	0.8078 1.796 0.0487 866
Surface Parameters:	-0.1707 -0.0618 1.000	1.260 10.32 1.387

Table 6.2a: Simulation Results of the Iterative Scheme after 100 Iterations

True Solution			Spurious Solution			
$\omega = [0, 0, 0]^T$			$\omega = [-10.97, 0.5501, -1.100]^T$			
$t = [1, 2, 0]^T$			$t = [0.4499, -8.975, 0]^T$			
$n = [-.0501, 0, 1]^T$			$n = [-0.1114, -0.2228, 1]^T$			
$d = [1, 10, 1]^T$			$d = [-0.2228, -2.167, 2.223]^T$			
Initial Condition: $n = [-0.0009, 0.0222, 0.0183]^T$ $d = [0.0208, 0.2203, 0.0160]^T$						
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error			
0%	0%	0	0.008			
	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	-0.0503	0.0283	-0.0030	0.9713	1.949	0.0001
Surface Parameters:	-0.0500	1.026	1.000	1.016	10.24	0.9851
Initial Condition: $n = [0.0105, 0.2075, 0.0542]^T$ $d = [0.1977, 0.7601, 0.0521]^T$						
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error			
10%	5.7%	33.8	32.9			
	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	-0.1915	0.1271	-0.1148	0.8694	1.771	0.0059
Surface Parameters:	0.0196	1.073	1.000	1.237	11.18	0.9541
Initial Condition: $n = [0.0002, 0.0223, 0.0209]^T$ $d = [0.2246, 0.9022, .0190]^T$						
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error			
25%	13.8%	262.	259.			
	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	0.2761	-0.1717	0.0484	1.153	2.258	0.0290
Surface Parameters:	-0.0221	0.9072	1.000	0.7884	8.831	1.034
Initial Condition: $n = [-0.0006, 0.0300, 0.0636]^T$ $d = [0.0191, 0.0239, 0.0002]^T$						
Max. Image Noise	S.D. Noise	True Solution Error	Optimal Solution Error			
50%	28.8%	897.	867.			
	Rotation/Surface Normal		Translation/Curvature Parameters			
Motion Parameters:	0.6243	-0.2802	0.1191	1.149	2.383	-0.0075
Surface Parameters:	0.0216	0.9127	1.000	1.172	8.190	1.285

Table 6.2b: Simulation Results of the Analytical Method

True Solution			Spurious Solution				
$\omega = [0, 0, 0]^T$			$\omega = [-10.97, 0.5501, -1.100]^T$				
$t = [1, 2, 0]^T$			$t = [0.4499, -8.975, 0]^T$				
$n = [-.0501, 0, 1]^T$			$n = [-0.1114, -0.2228, 1]^T$				
$d = [1, 10, 1]^T$			$d = [-0.2228, -2.167, 2.223]^T$				
Max. Image Noise			S.D. Noise		True Solution Error		
10%			5.7%		33.8		
Rotation/Surface Normal			Translation/Curvature Parameters			Error	
Motion Parameters:	-10.73	0.6376	-1.119	0.3711	-8.774	-0.1747	38.5
Surface Parameters:	-0.1174	-0.2242	1.000	-0.2525	-2.199	2.209	
Motion Parameters:	-0.1787	0.1384	-0.1318	0.8703	1.777	-0.1061	55.4
Surface Parameters:	0.0519	1.095	1.000	1.230	11.15	0.9613	
Max. Image Noise			S.D. Noise		True Solution Error		
25%			14.2%		262.		
Rotation/Surface Normal			Translation/Curvature Parameters			Error	
Motion Parameters:	-11.00	0.4866	-1.145	0.5105	-9.035	-0.3266	286.
Surface Parameters:	-0.1296	-0.2488	1.000	-0.2011	-2.136	2.238	
Motion Parameters:	0.2571	-0.1342	-0.0131	1.131	2.231	-0.2302	350.
Surface Parameters:	0.0674	0.9871	1.000	0.7867	8.938	1.028	
Max. Image Noise			S.D. Noise		True Solution Error		
50%			28.1%		897.		
Rotation/Surface Normal			Translation/Curvature Parameters			Error	
Motion Parameters:	-10.19	0.2458	-1.058	0.6417	-8.443	0.4754	949.
Surface Parameters:	-0.1584	-0.1913	1.000	-0.3278	-2.249	2.203	
Motion Parameters:	-0.3964	19.76	0.0338	-18.87	1.3588	-1.683	993.
Surface Parameters:	0.0308	-0.0455	1.000	2.056	-0.4441	-0.0708	
Motion Parameters:	0.6143	-0.2474	0.1183	1.135	2.369	0.6117	1347.
Surface Parameters:	-0.0676	0.7287	1.000	1.158	8.235	1.271	

Table 6.3a: Simulation Results of the Iterative Scheme after 60 Iterations

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 3]^T$ $n = [0, 0, 1]^T$ $d = [1, 10, 1]^T$

Initial Condition: $n = [-0.0028, 0.0007, -0.0110]^T$ $d = [-0.0534, 0.0621, 0.0281]^T$

Max. Image Noise S.D. Noise True Solution Error Optimal Solution Error
0% 0% 0 0.000

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	0.0000	0.0000	0.0000	1.000	2.000	3.000
Surface Parameters:	0.0000	0.0000	1.000	1.000	10.00	1.000

Initial Condition: $n = [0.0452, 0.0605, -0.0565]^T$ $d = [0.0236, -0.0130, 0.0387]^T$

Max. Image Noise S.D. Noise True Solution Error Optimal Solution Error
10% 5.7% 143. 132.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0941	-0.0987	0.2734	0.9950	1.863	2.844
Surface Parameters:	-0.0653	-0.0212	1.000	0.8484	9.456	0.8429

Initial Condition: $n = [0.0408, 0.0126, 0.0581]^T$ $d = [-0.0037, 0.0080, -0.0624]^T$

Max. Image Noise S.D. Noise True Solution Error Optimal Solution Error
25% 14.3% 676. 662.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	0.0129	0.0932	-0.3552	0.9477	1.953	2.898
Surface Parameters:	0.1188	-0.0630	1.000	0.1371	10.23	0.5272

Initial Condition: $n = [-0.0555, -0.0008, 0.0340]^T$ $d = [0.0277, -0.0564, -0.0163]^T$

Max. Image Noise S.D. Noise True Solution Error Optimal Solution Error
50% 28.4% 3653. 3520.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0108	0.4882	-0.5288	0.5391	1.385	2.065
Surface Parameters:	0.1572	-0.1502	1.000	2.219	15.22	3.207

Table 6.3b: Simulation Results of the Analytical Method

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 3]^T$ $n = [0, 0, 1]^T$ $d = [1, 10, 1]^T$

Max. Image Noise S.D. Noise True Solution Error
 10% 5.7% 143.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	-0.3657 0.0264 0.2440	0.8618 1.620 2.838	358
Surface Parameters:	-0.0680 0.2548 1.000	1.099 12.29 1.915	

Max. Image Noise S.D. Noise True Solution Error
 25% 14.2% 676.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	-0.6200 0.3660 0.1153	0.7070 1.528 3.485	3299.
Surface Parameters:	0.0212 0.2048 1.000	0.2241 12.79 0.8422	

Max. Image Noise S.D. Noise True Solution Error
 50% 28.1% 3653.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	-0.076 -0.8035 0.8997	1.571 2.398 3.551	4221.
Surface Parameters:	-0.6322 0.6414 1.000	-1.219 7.343 2.341	

Table 6.4a: Simulation Results of the Iterative Scheme after 60 Iterations

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 5]^T$ $n = [1, 2, 1]^T$ $d = [1, 10, 1]^T$

Initial Condition: $n = [0.0359, 0.0079, -0.0349]^T$ $d = [-0.0352, 0.0093, 0.0476]^T$

Max. Image Noise 0% S.D. Noise 0% True Solution Error 0 Optimal Solution Error 0.000

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	0.0000	0.0000	0.0000	1.000	2.000	5.000
Surface Parameters:	1.0000	2.0000	1.000	1.000	10.00	1.000

Initial Condition: $n = [-0.0304, 0.0201, 0.0258]^T$ $d = [-0.0462, 0.0470, 0.0389]^T$

Max. Image Noise 10% S.D. Noise 5.7% True Solution Error 191. Optimal Solution Error 185.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0907	-0.0655	0.0927	0.9720	1.835	4.737
Surface Parameters:	0.9798	2.103	1.000	1.222	10.46	1.051

Initial Condition: $n = [0.0470, 0.0511, 0.0353]^T$ $d = [0.0240, 0.0492, 0.0029]^T$

Max. Image Noise 25% S.D. Noise 14.0% True Solution Error 1060. Optimal Solution Error 1037.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0442	0.1055	-0.3038	0.9887	2.123	5.360
Surface Parameters:	0.9564	1.911	1.000	0.8244	9.241	0.6446

Initial Condition: $n = [-0.0145, -0.0596, -0.0071]^T$ $d = [-0.0310, -0.0361, 0.0085]^T$

Max. Image Noise 50% S.D. Noise 28.7% True Solution Error 4538. Optimal Solution Error 4396.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	0.0466	-0.0799	-0.5149	0.7977	1.8542	4.245
Surface Parameters:	1.078	2.247	1.000	1.263	11.03	1.679

Table 6.4b: Simulation Results of the Analytical Method

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 5]^T$ $n = [1, 2, 1]^T$ $d = [1, 10, 1]^T$

Max. Image Noise S.D. Noise True Solution Error
 10% 5.7% 191.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	-0.3836 -0.1504 0.1894	0.9214 1.624 4.681	633.
Surface Parameters:	0.9596 2.347 1.000	1.100 11.79 2.088	

Max. Image Noise S.D. Noise True Solution Error
 25% 14.2% 1060.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	0.1090 0.3099 -0.7275	0.7573 1.8464 4.929	1385.
Surface Parameters:	1.148 1.903 1.000	1.129 10.98 1.515	

Max. Image Noise S.D. Noise True Solution Error
 50% 28.1% 4538.

	Rotation/Surface Normal	Translation/Curvature Parameters	Error
Motion Parameters:	0.1936 -0.3724 -1.500	0.9598 1.8786 4.0745	5376.
Surface Parameters:	1.135 2.376 1.000	1.591 10.47 2.453	

Table 6.5a: Simulation Results of the Iterative Scheme after 100 Iterations

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 0]^T$ $n = [1, 1, 1]^T$ $d = [1, 10, 1]^T$

Initial Condition: $n = [-0.0126, 0.0454, 0.0168]^T$ $d = [-0.0179, -0.0488, 0.0093]^T$

Max. Image Noise 0% S.D. Noise 0% True Solution Error 0 Optimal Solution Error 0.001

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0166	0.0092	-0.0019	0.9906	1.983	0.0002
Surface Parameters:	1.010	1.008	1.000	1.005	10.07	0.9949

Initial Condition: $n = [0.0105, 0.0341, 0.0577]^T$ $d = [0.0335, 0.3142, 0.1008]^T$

Max. Image Noise 10% S.D. Noise 5.6% True Solution Error 36.1 Optimal Solution Error 35.7

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.0208	0.0108	0.0561	0.9870	1.975	-0.0106
Surface Parameters:	0.9380	1.018	1.000	0.9537	10.08	1.007

Initial Condition: $n = [.2071, .1248, .3208]^T$ $d = [.1105, .8927, .4224]^T$

Max. Image Noise 25% S.D. Noise 14.3% True Solution Error 202. Optimal Solution Error 193.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	0.1781	-0.1249	-0.0783	1.136	2.287	-0.0282
Surface Parameters:	0.8793	0.8376	1.000	0.5611	8.508	0.9165

Initial Condition: $n = [.0424, .0113, .0055]^T$ $d = [.0143, .2173, .0134]^T$

Max. Image Noise 50% S.D. Noise 28.6% True Solution Error 987. Optimal Solution Error 886.

	Rotation/Surface Normal			Translation/Curvature Parameters		
Motion Parameters:	-0.1368	-0.0666	0.1500	1.200	2.412	-0.0286
Surface Parameters:	0.7877	0.8522	1.000	-0.3475	7.521	-0.2791

Table 6.5b: Simulation Results of the Analytical Method

True Solution: $\omega = [0, 0, 0]^T$ $t = [1, 2, 0]^T$ $n = [1, 1, 1]^T$ $d = [1, 10, 1]^T$

	Max. Image Noise	S.D. Noise	True Solution Error				
	10%	5.6%	36.1				
	Rotation/Surface Normal			Translation/Curvature Parameters			Error
Motion Parameters:	-0.1104	-0.0820	0.0496	0.9287	1.877	0.0340	38.4
Surface Parameters:	0.9814	1.060	1.000	0.9849	10.59	0.9932	
	Max. Image Noise	S.D. Noise	True Solution Error				
	25%	14.3%	202.				
	Rotation/Surface Normal			Translation/Curvature Parameters			Error
Motion Parameters:	0.2383	-0.2140	-0.1582	1.192	2.360	-0.1205	204.
Surface Parameters:	0.9112	0.7873	1.000	0.5786	8.269	0.9433	
	Max. Image Noise	S.D. Noise	True Solution Error				
	50%	28.6%	987.				
	Rotation/Surface Normal			Translation/Curvature Parameters			Error
Motion Parameters:	-6.441	2.046	-0.6596	-0.5109	-4.107	-0.3514	1033.
Surface Parameters:	-0.6696	-0.4848	1.000	0.2363	-3.941	3.040	
	Rotation/Surface Normal			Translation/Curvature Parameters			Error
Motion Parameters:	-1.830	1.359	-0.1383	0.1760	0.5028	0.0770	1064.
Surface Parameters:	4.300	3.988	1.000	-3.107	34.28	-6.790	
	Rotation/Surface Normal			Translation/Curvature Parameters			Error
Motion Parameters:	-2.463	18.06	2.058	-16.53	-0.1306	-1.974	1204.
Surface Parameters:	-0.0416	-0.1777	1.000	2.273	-0.4031	0.1013	

Depth Positivity as a Constraint to Recover Motion

We have shown how we can recover the observer motion in certain special cases using the brightness change constraint equation alone. When the motion is purely rotational, the solution is obtained from three linear equations. When either the depth values for a minimum of six image points are known or the surface is a frontal plane, the motion can be determined from six linear equations. When the surface of the scene is planar at an unknown orientation, the motion parameters and the surface of the scene are determined from eight non-linear equations; a least-squares solution can be determined either iteratively or in closed form. When the surface is quadratic, 11 non-linear equations have to be solved iteratively. Alternatively, a non-robust estimate may be obtained in closed form.

In the general case, we cannot recover motion and shape uniquely using the brightness change equation alone. Suppose the pair $\{\omega', t'\}$ is chosen arbitrarily as a solution. Then the depth values can be determined using the brightness change equation

$$c + \mathbf{v} \cdot \boldsymbol{\omega}' + \frac{1}{Z'} \mathbf{s} \cdot \mathbf{t}' = 0,$$

that is,

$$Z' = \frac{\mathbf{s} \cdot \mathbf{t}'}{c + \mathbf{v} \cdot \boldsymbol{\omega}'},$$

provided that the denominator is not zero. This may suggest that for any choice of the pair $\{\omega', t'\}$, we can determine the depth values such that the brightness change equation is satisfied at every image point. Therefore, many solutions are possible since the motion parameters can be chosen arbitrarily. Again, we conclude that we need other constraints to recover motion uniquely.

Since the surface has to be in front of the camera for it to be seen, the depth values are constrained to be positive. In theory, any motion pair $\{\omega', t'\}$ that gives rise to negative depth values is not a solution. Therefore, the problem is to determine the pair $\{\omega, t\}$ that gives rise to positive depth values ($Z > 0$) over the whole image. We also need to address the uniqueness issue; that is, given that the brightness change equation is satisfied for the motion $\{\omega, t\}$ and the surface $Z > 0$, is there another motion $\{t', \omega'\}$ and another surface $Z' > 0$ that satisfies the brightness change equation at every point in the image? In general, this is clearly so since an image of uniform brightness could correspond to an arbitrary uniform surface moving in an arbitrary way. Hence, the brightness gradients or lack of brightness gradients can conspire to make the problem highly ambiguous.

We need to use the constraint from the whole image region since the problem may remain under-constrained if we restrict ourselves to information from only a local region. In practice, with noisy images, it is more likely that there is no solution; that is, the depth values for some points in the image may be negative for every possible motion including the correct one. We then have to improvise a method for selecting a solution that is most consistent with the data. In this chapter, we address the problem of recovering motion from the brightness change equation by imposing the positivity constraint. Unfortunately, the problem is difficult when both rotation and translation are unknown. We first restrict ourselves to the special case when either rotation is zero or it is known. We then show how the method proposed for this special case can be extended to address the general problem. By incorporating the depth positivity constraint, Horn & Weldon [1986] also present several direct methods for estimating the translation of the viewer.

7.1 Pure Translation or Known Rotation

Suppose the rotational component of motion is known. Then we can write the brightness change equation in the form

$$\tilde{c} + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) = 0,$$

where $\tilde{c} = c + \mathbf{v} \cdot \boldsymbol{\omega}$. For simplicity, let c replace \tilde{c} .

The question regarding the uniqueness is now as follows: Given that the brightness change equation is satisfied for the translational motion \mathbf{t} and the surface $Z > 0$, is there another translational motion \mathbf{t}' and another surface $Z' > 0$ that satisfies the brightness change equation at every point in the image? It should be mentioned that, given the motion field over some region in the image, the solution can be determined uniquely when the motion is pure translation (for example, see Bruss & Horn [1983]). As we see, the uniqueness proof we present actually serves as a direct method for determining the translation vector \mathbf{t} . Before we address this problem, we show how a simplified constraint can be used to recover the translation vector provided that certain feature points can be identified.

7.1.1 Feature Points

An image point where $c = 0$ will be referred to as a *feature point*. In the case of pure translation ($\omega = 0$), this implies that the time derivative of brightness is zero at a feature point. In order to exclude regions of uniform brightness from consideration, we restrict ourselves to points with non-zero brightness gradient ($E_r \neq 0$).

When $c = 0$, the brightness change equation reduces to

$$\frac{1}{Z} (\mathbf{s} \cdot \mathbf{t}) = 0.$$

Provided that depth is finite we must have

$$(\mathbf{s} \cdot \mathbf{t}) = 0.$$

Since Z drops out of the equation, we conclude that the depth value cannot be computed at a feature point. These points, however, provide strong constraints on the location of the FOE.

Ideally, just two non-parallel vectors \mathbf{s}_1 and \mathbf{s}_2 at two image points define the translational vector \mathbf{t} . The solution is given by

$$\mathbf{t} = k (\mathbf{s}_1 \times \mathbf{s}_2),$$

where k represents the scale factor ambiguity. Because of noise in the images and quantization error, the earlier constraint equation will not be satisfied exactly. Therefore, we instead minimize the sum of the square of the error in this equation for every feature point; that is, we minimize

$$\sum_{i=1}^n (\mathbf{s}_i \cdot \mathbf{t})^2.$$

Since only the direction of \mathbf{t} can be determined, we further impose the constraint that $\|\mathbf{t}\| = 1$. This is a constrained minimization problem. We can reduce it to an unconstrained minimization problem with a closed-form solution by introducing a lagrange multiplier. In the new formulation, we instead minimize

$$J = \sum_{i=1}^n (\mathbf{s}_i \cdot \mathbf{t})^2 + \lambda(1 - \mathbf{t}^T \mathbf{t}) = \mathbf{t}^T \left(\sum_{i=1}^n \mathbf{s}_i \mathbf{s}_i^T \right) \mathbf{t} + \lambda(1 - \mathbf{t}^T \mathbf{t}).$$

Necessary conditions for optimality are

$$\frac{\partial J}{\partial \mathbf{t}} = 0 \quad \text{and} \quad \frac{\partial J}{\partial \lambda} = 0.$$

Performing the above differentiation we arrive at

$$\left(\sum_{i=1}^n \mathbf{s}_i \mathbf{s}_i^T \right) \mathbf{t} = \lambda \mathbf{t} \quad \text{and} \quad \mathbf{t}^T \mathbf{t} = 1.$$

This is an eigenvalue-eigenvector problem; that is, $\{\mathbf{t}, \lambda\}$ is an eigenvector-eigenvalue pair of the 3×3 matrix

$$\sum_{i=1}^n \mathbf{s}_i \mathbf{s}_i^T.$$

Even though this matrix generally has three eigenvector-eigenvalue pairs, it is easy to show that we want \mathbf{t} to be the eigenvector associated with the smallest eigenvalue. This follows simply by substituting the solution into the expression for J ,

$$J = \mathbf{t}^T(\lambda \mathbf{t}) + \lambda(1 - \mathbf{t}^T \mathbf{t}) = \lambda \mathbf{t}^T \mathbf{t} + \lambda - \lambda \mathbf{t}^T \mathbf{t} = \lambda.$$

Clearly, J is minimized if we choose the smallest eigenvalue.

It should be noted that with two feature points, the 3×3 matrix has at most rank two since it is the sum of two dyadic products. Then the solution is the eigenvector corresponding to the zero eigenvalue. Geometrically, this is the vector normal to the plane formed by \mathbf{s}_1 and \mathbf{s}_2 as shown earlier. The matrix degenerates into a rank-one matrix if the vectors \mathbf{s}_1 and \mathbf{s}_2 are parallel; in this case, there are two zero eigenvalues and the solution is non-unique. In this degenerate case, the solution is constrained to lie in the plane with normal \mathbf{s}_1 (or \mathbf{s}_2) but it is not uniquely defined since any vector \mathbf{t} in this plane satisfies

$$\mathbf{s}_1 \cdot \mathbf{t} = 0 \quad \text{and} \quad \mathbf{s}_2 \cdot \mathbf{t} = 0.$$

7.1.2 Constraints Imposed by Brightness Gradient Vectors

We first assume that two translational motions and two surfaces satisfy the brightness change equation; that is, we have

$$c + \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) = 0 \quad \text{and} \quad c + \frac{1}{Z'}(\mathbf{s} \cdot \mathbf{t}') = 0.$$

Here, $\{Z, \mathbf{t}\}$ denotes the true solution and $\{Z', \mathbf{t}'\}$ denotes a spurious (or assumed) solution. We will show that we must have $Z = Z'$ and $\mathbf{t} = \mathbf{t}'$ and, therefore, the solution is unique.

Solving for Z and Z' we obtain

$$Z = -\frac{1}{c}(\mathbf{s} \cdot \mathbf{t}) \quad \text{and} \quad Z' = -\frac{1}{c}(\mathbf{s} \cdot \mathbf{t}').$$

The depth value cannot be computed at a point where $c = 0$; that is, at a feature point. Since we already know how to exploit the information at these points, we exclude them from further consideration and assume that $c \neq 0$ from now on.

Since the surface of the scene has to be in front of the imaging system for it to be viewed, both Z and Z' should be non-negative for every point in the image. Since Z is the true solution, we have $Z > 0$. If $\{Z', \mathbf{t}'\}$ is an acceptable solution, we must have

$$ZZ' = \frac{1}{c^2}(\mathbf{s} \cdot \mathbf{t})(\mathbf{s} \cdot \mathbf{t}') > 0.$$

We can write $\mathbf{s} = (\mathbf{E}_r \times \hat{\mathbf{z}}) \times \mathbf{r}$ in the form

$$\mathbf{s} = (\mathbf{r} \cdot \mathbf{E}_r) \hat{\mathbf{z}} - (\mathbf{r} \cdot \hat{\mathbf{z}}) \mathbf{E}_r.$$

Noting that $\mathbf{r} \cdot \hat{\mathbf{z}} = 1$, this can be simplified to

$$\mathbf{s} = (\mathbf{r} \cdot \mathbf{E}_r) \hat{\mathbf{z}} - \mathbf{E}_r.$$

Since the first two components of $\hat{\mathbf{z}}$ and the last component of \mathbf{E}_r are zero, we can write

$$\mathbf{s} = \begin{pmatrix} -\tilde{\mathbf{E}}_r \\ \tilde{\mathbf{r}} \cdot \tilde{\mathbf{E}}_r \end{pmatrix},$$

where $\tilde{\mathbf{E}}_r = (E_x, E_y)$ and $\tilde{\mathbf{r}} = (x, y)$ are 2-D vectors. Similarly, assuming that t_3 is non-zero (we deal with the case $t_3 = 0$ later) we can write

$$\mathbf{t} = t_3 \begin{pmatrix} \tilde{\mathbf{t}} \\ 1 \end{pmatrix}.$$

The vector $\tilde{\mathbf{t}}$ gives the location of the focus of expansion/contraction for an approaching/departing motion. This is the point where the axis of translation intersects the image plane and will be referred to as the FOE/FOC. Substituting for \mathbf{s} and \mathbf{t} in terms of $\tilde{\mathbf{E}}_r$, $\tilde{\mathbf{r}}$, and $\tilde{\mathbf{t}}$, we arrive at

$$\mathbf{s} \cdot \mathbf{t} = t_3 (\tilde{\mathbf{r}} \cdot \tilde{\mathbf{E}}_r - \tilde{\mathbf{t}} \cdot \tilde{\mathbf{E}}_r) = t_3 [\tilde{\mathbf{E}}_r \cdot (\tilde{\mathbf{t}} - \tilde{\mathbf{r}})].$$

Similarly, we can write

$$\mathbf{s} \cdot \mathbf{t}' = t'_3 [\tilde{\mathbf{E}}_r \cdot (\tilde{\mathbf{t}}' - \tilde{\mathbf{r}})],$$

where

$$\mathbf{t}' = t'_3 \begin{pmatrix} \tilde{\mathbf{t}}' \\ 1 \end{pmatrix}.$$

The vector $\tilde{\mathbf{t}}'$ gives the coordinates of the FOE corresponding to the translational vector \mathbf{t}' . Substituting these into the earlier equation we obtain

$$ZZ' = \frac{1}{c^2} t_3 t'_3 ((\tilde{\mathbf{t}} - \tilde{\mathbf{r}}) \cdot \tilde{\mathbf{E}}_r) ((\tilde{\mathbf{t}}' - \tilde{\mathbf{r}}) \cdot \tilde{\mathbf{E}}_r) > 0.$$

If we define the function S to be

$$S(\mathbf{t}', \tilde{\mathbf{r}}) = t_3 t'_3 ((\tilde{\mathbf{t}} - \tilde{\mathbf{r}}) \cdot \tilde{\mathbf{E}}_r) ((\tilde{\mathbf{t}}' - \tilde{\mathbf{r}}) \cdot \tilde{\mathbf{E}}_r),$$

then $S(\mathbf{t}', \tilde{\mathbf{r}})$ should be non-negative for every image point $\tilde{\mathbf{r}}$ if \mathbf{t}' is an acceptable solution. If we let

$$\tilde{\mathbf{x}} = (\tilde{\mathbf{t}} - \tilde{\mathbf{r}}) \quad \text{and} \quad \tilde{\mathbf{x}}' = (\tilde{\mathbf{t}}' - \tilde{\mathbf{r}}),$$

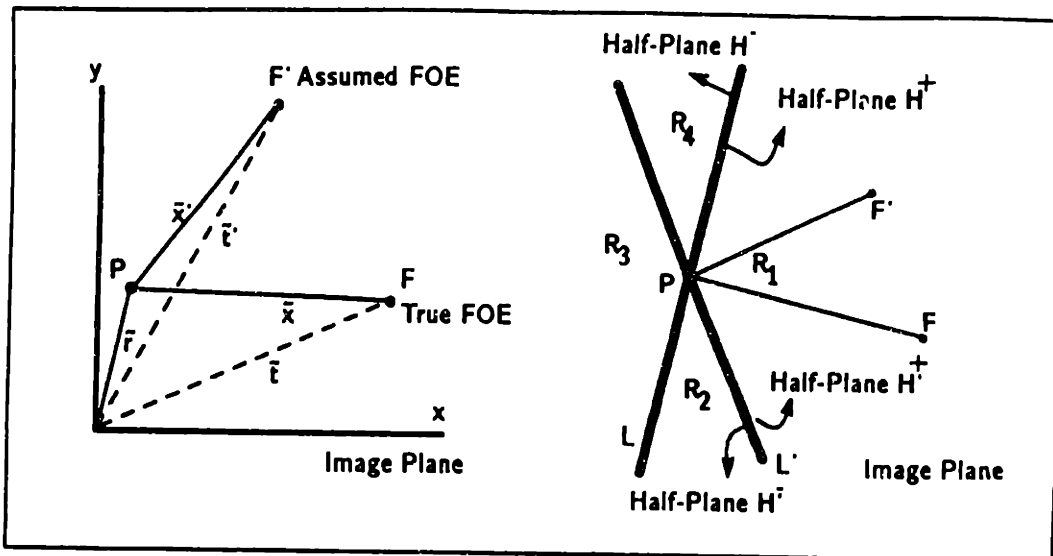


Figure 7-1. True and spurious (assumed) FOE's define the positive and negative half-planes.

we finally arrive at

$$S = t_3 t'_3 (\tilde{x} \cdot \tilde{E}_r)(\tilde{x}' \cdot \tilde{E}_r) > 0.$$

The vectors \tilde{x} and \tilde{x}' represent the line segments from a point P (with coordinates $\tilde{r} = (x, y)$) to the true and spurious FOE's, respectively. These are the line segments PF and PF' in Figure 7-1a. The scalar product $(\tilde{x} \cdot \tilde{E}_r)$ is positive if the angle between \tilde{x} and the brightness gradient vector at point P is less than $\pi/2$, and it is negative when the angle is greater than $\pi/2$. Clearly, it is zero when \tilde{x} is orthogonal to the gradient vector. Similarly, the dot product $(\tilde{x}' \cdot \tilde{E}_r)$ is positive, zero, or negative when the angle between \tilde{x}' and the gradient vector at point P is less than, equal to, or greater than $\pi/2$.

Now, for any two acceptable solutions, $\{t, Z\}$ and $\{t', Z'\}$, the product

$$p = (\tilde{x} \cdot \tilde{E}_r)(\tilde{x}' \cdot \tilde{E}_r)$$

is always either positive if $t_3 t'_3 > 0$ or negative if $t_3 t'_3 < 0$. Without loss of generality let us assume that $t_3 t'_3 > 0$. We use a simple geometric argument to show that the product p changes sign unless $\tilde{x} = \tilde{x}'$ which implies that $\tilde{t} = \tilde{t}'$. Consequently, we must have $t = (t_3/t'_3)t' = kt'$ for some non-zero constant k . Since we can recover the translational motion only up to a scale factor, we will conclude that the solution is unique up to the scale factor ambiguity.

Let us denote the two lines orthogonal to vectors \tilde{x} and \tilde{x}' by L and L' (see Figure 7-1b). Each line divides the image region into two half-planes. Let H^+ and H^- denote the half-planes corresponding to the line segment L and let H'^+ and H'^- be the half-planes corresponding to the line segment L' . The half-planes H^+ and H'^+ include

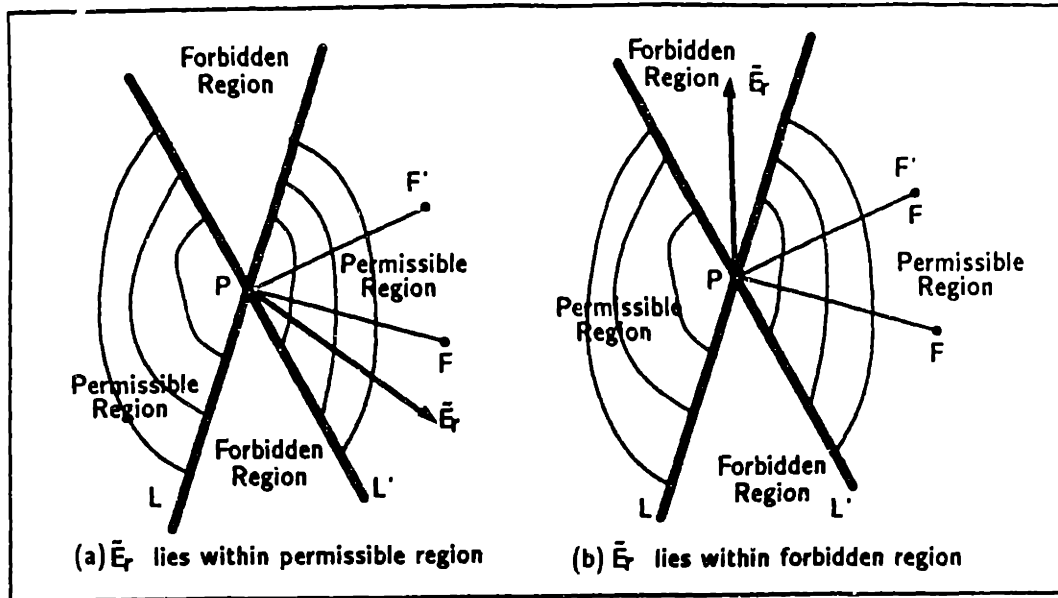


Figure 7-2. Permissible and forbidden regions.

regions of the image that correspond to positive values for $(\tilde{x} \cdot \tilde{E}_r)$ and $(\tilde{x}' \cdot \tilde{E}_r)$, respectively. Conversely, the scalar products $(\tilde{x} \cdot \tilde{E}_r)$ and $(\tilde{x}' \cdot \tilde{E}_r)$ are negative in H^- and H'^- , respectively.

The two line segments L and L' divide the image plane into four regions, R_1, \dots, R_4 . Regions R_1 and R_3 comprise the intersections of the half-planes H^+ with H'^+ and H^- with H'^- , respectively; that is,

$$R_1 = H^+ \cap H'^+ \quad \text{and} \quad R_3 = H^- \cap H'^-$$

Similarly, regions R_2 and R_4 comprise the intersections of the half-planes H^+ with H'^- and H^- with H'^+ , respectively; that is,

$$R_2 = H^+ \cap H'^- \quad \text{and} \quad R_4 = H^- \cap H'^+$$

Suppose \tilde{E}_r lies in R_1 . Then, the product p is positive since both $(\tilde{x} \cdot \tilde{E}_r)$ and $(\tilde{x}' \cdot \tilde{E}_r)$ are positive. It is also positive if \tilde{E}_r lies in the region R_3 since both $(\tilde{x} \cdot \tilde{E}_r)$ and $(\tilde{x}' \cdot \tilde{E}_r)$ are negative. Since Z (the true solution) is positive and $S = c^2 Z Z' = c^2 t_3 t_3' p > 0$ if \tilde{E}_r lies in $R_1 \cup R_3$, we conclude that

$$Z' > 0 \quad \text{if } \tilde{E}_r \text{ lies in } R_1 \cup R_3.$$

We will term $R_1 \cup R_3$ the *permissible region* because \tilde{E}_r has to lie in this region in order to satisfy the constraint $Z' > 0$. Conversely, the region $R_2 \cup R_4$ will be termed the *forbidden region* since \tilde{E}_r cannot lie in this region in order to have $Z' > 0$ (see Figure 7-2). Note that when $t_3 t_3' < 0$, the permissible region will be the region consisting of R_2 and R_4 and the forbidden region will consist of R_1 and R_3 .

We now show that the vector \tilde{E}_r has to lie in the forbidden region for some image points if $\tilde{t} \neq \tilde{t}'$. Consequently, we obtain

$$S = ZZ' = t_3 t_3' p < 0 \iff Z' < 0.$$

Therefore, Z' is not an acceptable solution.

Suppose the point P lies along the direction defined by the line segment FF' which we refer to as a *FOE constraint line*. Then we have

$$\tilde{r} = (1 - \alpha)\tilde{t} + \alpha\tilde{t}' \quad \text{for some } \alpha.$$

When $0 < \alpha < 1$, the point P lies on the segment between the points F and F' . When $\alpha < 0$, it lies on the ray emanating from F (segment FX) and when $\alpha > 1$, it lies on the ray emanating from F' (segment $F'X'$). For these points, we must have

$$\tilde{x} = \tilde{t} - \tilde{r} = \alpha(\tilde{t} - \tilde{t}') \quad \text{and} \quad \tilde{x}' = \tilde{t}' - \tilde{r} = (\alpha - 1)(\tilde{t} - \tilde{t}').$$

The product p is then given by

$$p = (\tilde{x} \cdot \tilde{E}_r)(\tilde{x}' \cdot \tilde{E}_r) = \alpha(\alpha - 1) ((\tilde{t} - \tilde{t}') \cdot \tilde{E}_r)^2.$$

It is clear that when $0 < \alpha < 1$, the product p will be negative unless the gradient vector is orthogonal to FF' (note that FF' is the vector $(\tilde{t} - \tilde{t}')$); that is, the point is a feature point which has been excluded from consideration. Conversely, the product is positive when $\alpha < 0$ or $\alpha > 1$. Since

$$S = c^2 ZZ' = c^2 t_3 t_3' p,$$

this implies that, for points on the line segment FF' , the depth values Z' are guaranteed to be negative (unless the point is a feature point). Conversely, for points along the rays FX and $F'X'$, the depth values are guaranteed to be positive (unless the point is a feature point).

A probability value can be assigned to each image point as a measure of the likelihood that $Z' < 0$ at that image point. Since $Z' < 0$ if the gradient vector lies outside the permissible region, it is concluded that the probability distribution function depends on θ , the angle between the vectors \tilde{x} and \tilde{x}' , as well as on the distribution of the brightness gradient vectors.

When θ is small, the permissible region contains a larger portion of the image plane (see Figure 7-3a). Therefore, the points with small θ values are more likely to have positive depth values even for an incorrect translational vector t' . These are points that are either far from the FOE constraint line or points that are in the vicinity of or on the two line segments FX and $F'X'$.

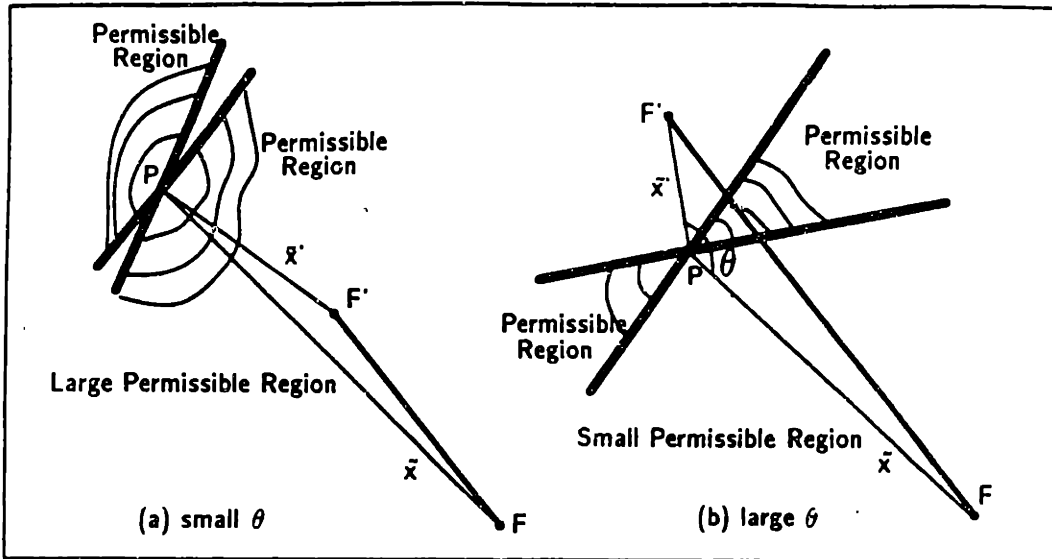


Figure 7-3. Relationship between the size of the permissible region and the relative position of an image point with respect to the FOE constraint line.

Conversely, when θ is large, the permissible region comprises a smaller portion of the image plane (see Figure 7-3b). Therefore, it is more likely that the brightness gradient lies outside this region giving rise to a negative depth value. In the extreme case when $\theta = \pi$ (that is, the point lies along FF') the depth values are guaranteed to be negative. Interestingly, the forbidden region for a point on FF' is the whole image plane not including the line orthogonal to FF' at that image point. If the gradient vector lies in that direction, then the point has to be a feature point.

Suppose that the probability distribution of the gradient vectors is independent of the image position and is rotationally symmetric; that is, all directions of brightness gradients are equally likely. It is not difficult to see that the probability distribution function for the likelihood that a point in the image plane gives rise to a negative depth value is given by

$$\text{Prob}(S < 0) = \text{Prob}(Z' < 0) = \frac{\theta}{\pi}.$$

The constant probability loci are circles that pass through F and F' . Also, there is a symmetry about the FOE constraint line as shown in Figure 7-4.

Let h denote the distance from an image point P to the FOE constraint line and f the length of the line segment FF' (see Figure 7-5a). If Q is the projection of the point on XX' and if we define

$$\alpha = \frac{\|FQ\|}{\|FF'\|},$$

then it is easy to show that

$$\theta = \tan^{-1}\left(\frac{\alpha f}{h}\right) + \tan^{-1}\left(\frac{(1-\alpha)f}{h}\right)$$

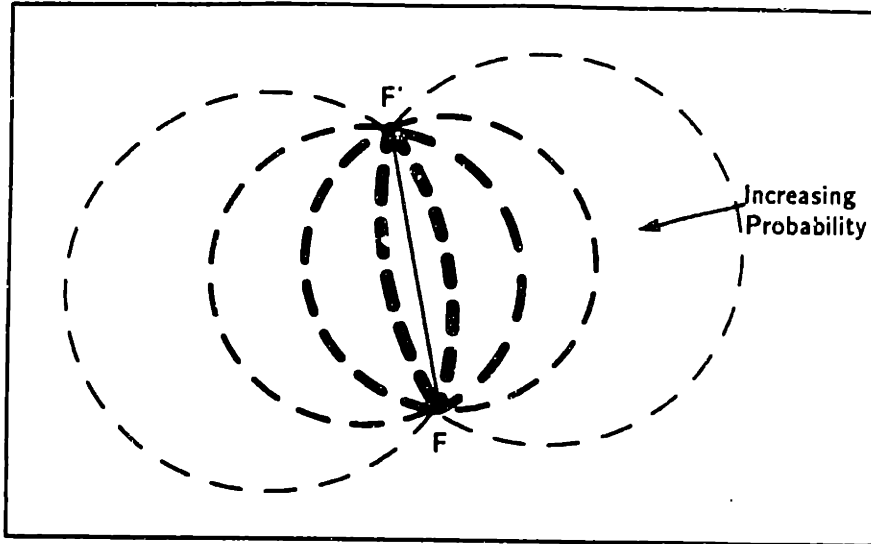


Figure 7-4. Constant probability loci.

which can be simplified to

$$\theta = \tan^{-1}\left(\frac{hf}{h^2 - \alpha(1 - \alpha)f^2}\right).$$

Therefore, we have

$$\text{Prob}(Z' < 0) = \frac{1}{\pi} \left[\tan^{-1}\left(\frac{\alpha f}{h}\right) + \tan^{-1}\left(\frac{(1 - \alpha)f}{h}\right) \right] = \frac{1}{\pi} \left[\tan^{-1}\left(\frac{hf}{h^2 - \alpha(1 - \alpha)f^2}\right) \right].$$

This shows that a point close to FF' (small h) with $0 < \alpha < 1$ (that is, its projection on XX' lies on FF') is very likely to have a negative depth value as explained earlier. At a constant distance h from FF' , the angle θ is a maximum when the point P is equidistant from F and F' . Assuming that the brightness gradient vectors are distributed more or less randomly, we expect that the cluster of negative depth values around FF' has a shape similar to that shown in Figure 7-5b.

To summarize, there are points in the image that give rise to a negative depth value when the incorrect translation vector is assumed. These points are more likely to be found in the vicinity of the line segment that connects the incorrect focus of expansion to the true one; that is, the FOE constraint line (later this is exploited to locate the true FOE). As F' approaches F , the region around FF' that is likely to contain points with negative depth values shrinks in size. In the limit when F' coincides with F (that is, $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}'$ and, therefore, $\tilde{\mathbf{t}} = k\tilde{\mathbf{t}}'$ for some non-zero k), all of the depth values will be non-negative since

$$S = t_3 t_3' (\tilde{\mathbf{x}} \cdot \tilde{\mathbf{E}}_r) (\tilde{\mathbf{x}}' \cdot \tilde{\mathbf{E}}_r) = t_3 t_3' [(\tilde{\mathbf{x}} \cdot \tilde{\mathbf{E}}_r)]^2 > 0.$$

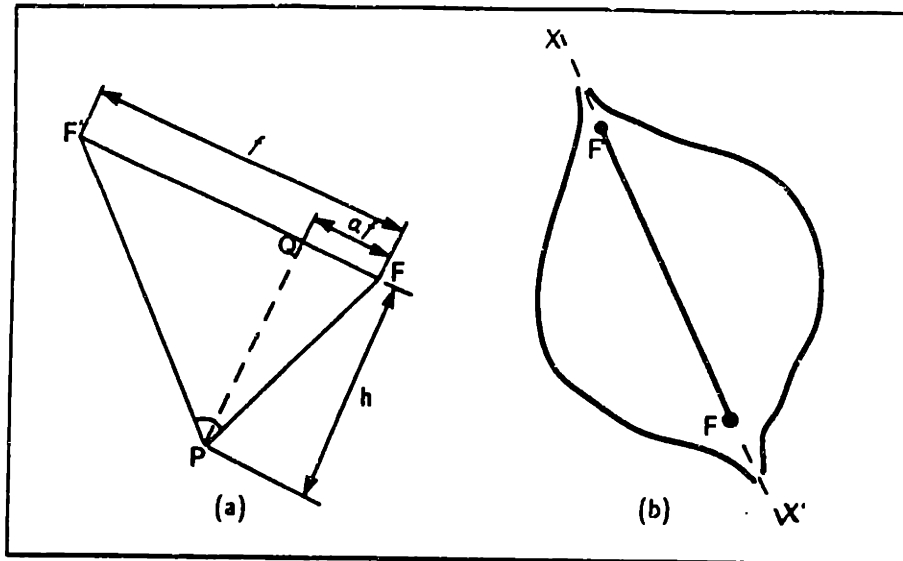


Figure 7-5. Expected shape of the negative depth cluster.

We have a similar situation when $t_3 = 0$. In this case, the true FOE is at infinity. From symmetry, we must have $t'_3 = 0$ for any spurious solution. If we let

$$\mathbf{t} = \begin{pmatrix} \tilde{\mathbf{t}} \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{t}' = \begin{pmatrix} \tilde{\mathbf{t}}' \\ 0 \end{pmatrix},$$

then we can write

$$\mathbf{s} \cdot \mathbf{t} = \tilde{\mathbf{t}} \cdot \tilde{\mathbf{E}}_r \quad \text{and} \quad \mathbf{s} \cdot \mathbf{t}' = \tilde{\mathbf{t}}' \cdot \tilde{\mathbf{E}}_r.$$

Using these in S , we obtain

$$S = (\tilde{\mathbf{t}} \cdot \tilde{\mathbf{E}}_r)(\tilde{\mathbf{t}}' \cdot \tilde{\mathbf{E}}_r).$$

The half-planes $\{H^+, H^-\}$ and $\{H'^+, H'^-\}$ are now defined by the vector $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{t}}'$ (instead of $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ for the case $t_3 \neq 0$). We need to replace $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ by $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{t}}'$, respectively, in our earlier analysis. Since these vectors are constants, we conclude that θ (in this case, this becomes the angle between the two vectors $\tilde{\mathbf{t}}$ and $\tilde{\mathbf{t}}'$) is the same for every image point. If the distribution of brightness gradient vectors is rotationally symmetric and independent of the image position, each image point can give rise to a negative depth value with probability equal to θ/π . Therefore, the depth values will be negative for some image points unless $\tilde{\mathbf{t}} = \tilde{\mathbf{t}}'$.

7.1.2.1 Locating Focus of Expansion using Gradient Vectors

First, we consider the case when the FOE is located in the image. As before we denote the true FOE by F and an assumed FOE by F' . The continuation of FF' beyond F is denoted by FX and beyond F' by $F'X'$. We showed that the line segment connecting an assumed (incorrect) FOE to the true FOE, FF' , is the locus of points that are guaranteed

to have negative depth values. Furthermore, the continuation of this line beyond F and F' (FX and $F'X'$) is where the depth values are expected to be positive. Therefore, we expect to find a cluster of negative depth values around FF' and two clusters of positive depth values around FX and $F'X'$ (even though these may be outside the image plane if F or F' are either outside or close to the boundaries of the image plane). In order to locate the FOE, we need to determine the direction toward and the distance to the true FOE from the assumed FOE.

To determine the direction, we need to identify the clusters of positive and negative depth values. Then we can choose the axis of symmetry (or the axis of least inertia) of these clusters as the FOE constraint line (see Figure 7-6a). Alternatively, we may do the following: consider a circle of some radius r centered at F' that does not contain the true FOE and is located totally within the image plane (see Figure 7-6b). We can construct a histogram of the number of image points within the circle with a negative depth value for all possible directions $0 \leq \phi < 2\pi$; obviously, we cannot obtain every angle because of quantization. In general, the peak of the histogram corresponds to the direction toward the FOE (that is, the orientation of the FOE constraint line). Furthermore, if we denote the corresponding angle by ϕ_f , then $\phi_f \pm \pi$ (depending on which sign gives an angle between 0 and 2π) should correspond to a minimum in the histogram (see Figure 7-6c). To obtain more robust results, we can take advantage of this fact as follows. Let us denote the number of negative and positive depth values in a certain direction ϕ by $N^-(\phi)$ and $N^+(\phi)$, respectively, and let $M(\phi) = N^-(\phi) - N^+(\phi)$ denote the difference between the number of negative and positive depth values in that direction. Then, we can instead construct the histogram of $(M(\phi) - M(\pi + \phi))$ for all possible directions $0 \leq \phi < \pi$ (see Figure 7-6d). Again, the peak of the histogram should correspond to the direction toward the true FOE; however, the new histogram should have a stronger peak.

The size of r , the radius of the circle, is important. If r is too large, then the circle may contain the FOE and the points within the positive cluster beyond FOE (that is, around FX) will subtract from the total and, therefore, will flatten the peak. If r is too small, then the histogram may not have a distinct peak due to noise in the images and quantization.

Now, suppose we have determined the FOE constraint line either by choosing the axis of symmetry of the negative depth cluster around F' or through the histogram method. The correct FOE is expected to lie on the line $F'X$. To determine the location of the FOE (the distance from F' to F) we construct the histogram of $M_h(l)$, where

$$M_h(l) = N_h^-(l) - N_h^+(l).$$

$N_h^-(l)$ (or $N_h^+(l)$) is the number of negative (or positive) depth values within some distance h from the constraint line at a distance l along $F'X$ from F' (see Figure 7-6e). In general, the peak in the histogram roughly corresponds to half the distance to the true FOE. Another estimate is where the histogram crosses zero since $M_h(l)$ should become negative beyond the FOE.

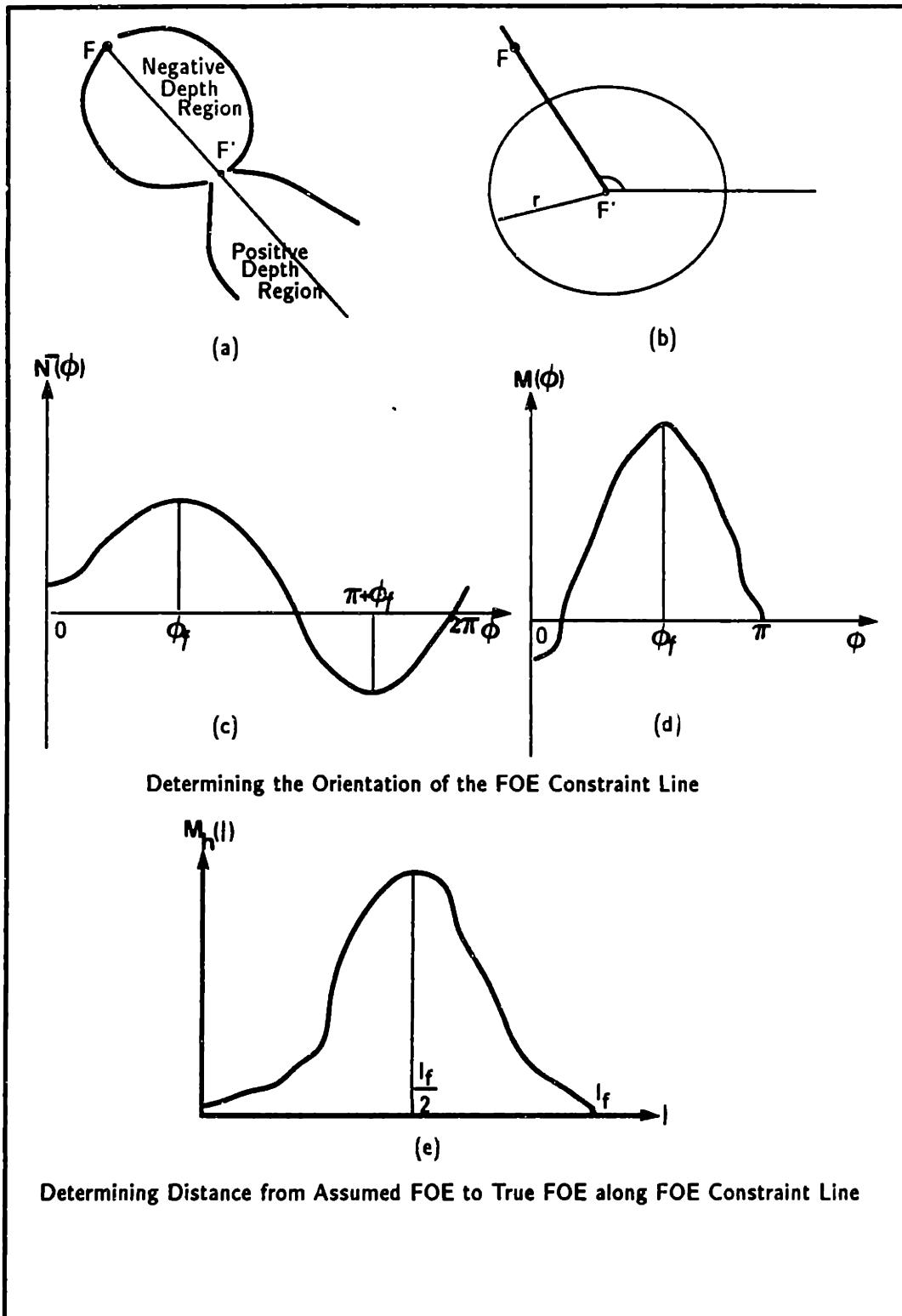


Figure 7-6. Determining the location of the FOE from a FOE constraint line.

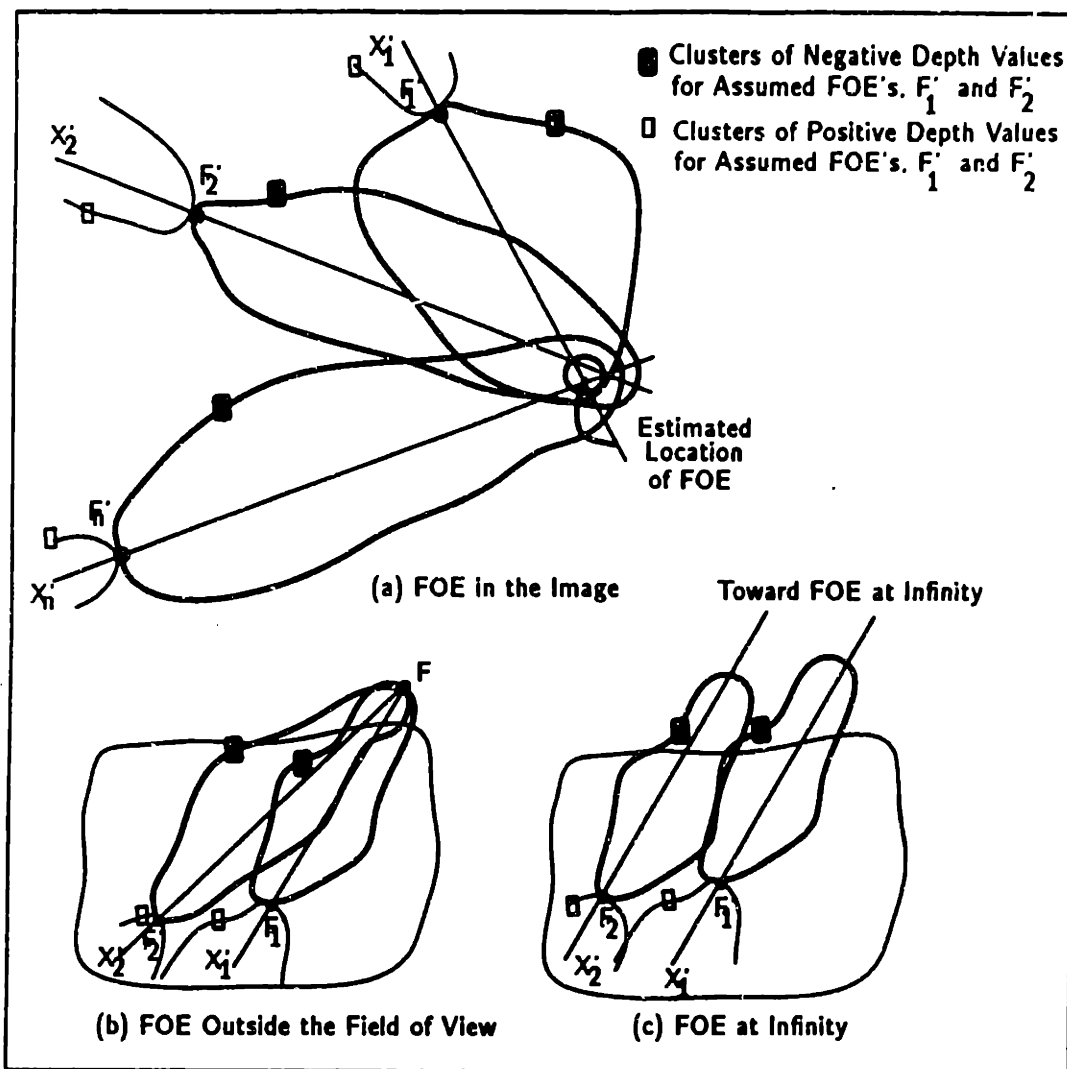


Figure 7-7. Determining the location of the FOE from several FOE constraint lines.

The procedure for determining the distance to the FOE will not work if the true FOE lies outside the field of view. To be able to deal with this case as well, we consider an alternative method. We first identify several constraint lines (corresponding to different F' 's). As shown in Figure 7-7a, the FOE is then chosen to be the best estimate of the common intersection of these lines; for example, the point with the least total square distance from the constraint lines (see Appendix E). This also has the advantage that the estimate is likely to be more robust with respect to image noise and quantization error.

The procedure is the same when the FOE is outside the image plane (including the case $t_3 = 0$ where the FOE is at infinity) as shown in Figures 7-7b and 7-7c. We should choose the point F' to be in the image plane. Otherwise, the whole FOE constraint line (FF') will be outside the field of view and so the clusters of negative or positive depth

values cannot be identified. The FOE is still determined from the best estimate of the common intersection of the FOE constraint lines; however, the intersection is outside the image plane.

The accuracy of the estimate of the location of the FOE will depend on the choice of F' and consequently on the shape and size of the clusters of negative and positive depth values. In general, more robust results are obtained if several F' 's are considered. Usually, once we determine one constraint line, it can lead us to choose the next F' 's more appropriately.

7.2 Unknown Rotation

The problem of locating the FOE from gradient vectors is similar to the problem of estimating the location of the FOE from optical flow data in the following sense. When the motion is only translation, the FOE can be determined rather easily from the intersection of the optical flow vectors (using the fact that these vectors point toward the FOE for a departing motion and emanate from the FOE for an approaching motion). In practice, these vectors do not intersect at a common point because of noise; Thus, the best estimate may be chosen to be the point with the least total square distance from the optical flow vectors. Unfortunately, these vectors do not intersect at the FOE when the rotational component is non-zero. Similarly, we expect that the FOE constraint lines do not intersect at a common point when the rotational component is non-zero and unknown. An intuitive approach is to assume some rotation vector in order to discount the contribution of the rotational component before we apply the method given for the case of a purely translational motion. Obviously, the estimate we obtain for the FOE may be very poor if the rotational component is not chosen accurately. This, however, is exactly the behavior we want if our method is to work in the general case. Thus, in order to have a distinct peak in the measure we use as a criterion for selecting the best estimate, we need to obtain a bad estimate for the FOE for an incorrect rotation. The measure of "badness" denoted $e(\omega)$ can be the total square distance of the estimated FOE from the constraint lines. Then, the best estimate of the motion parameters is the one that minimizes this error function. In general, it is difficult to compute this function for every possible rotation. An approach for dealing with this problem follows.

Suppose, an upper bound for each component of the rotational vector ω is available; for example, it is known that $|\omega_i| \leq \omega_i^{\max}$. If each interval from $-\omega_i^{\max}$ to ω_i^{\max} is divided into n smaller intervals, we can restrict the search to the n^3 discrete points in ω -space. Let us denote a point in this space by ω_{ijk} for $i, j, k = 1, 2, \dots, n$. For each possible point in this space (that is, for each ω_{ijk}) we estimate the location of the FOE using the method given earlier keeping the value of the error $e(\omega_{ijk})$. The best estimate corresponds to a minimum of the error function. We may then perform a local search in the neighborhood of ω_{ijk} to obtain a better estimate.

With perfect data, the minimum of the error function should give the correct estimate

of the motion parameters. This, of course, may not be the case for several reasons. First, the situations that give rise to an ambiguity in the interpretation of the motion field, as discussed in Chapter 6, may remain ambiguous for a small field of view since the depth values for every possible surface (up to three solutions) may be positive for every image point. It is also likely that we have to deal with the types of inherent ambiguities that arise when noisy optical flow data are used to recover 3-D motion. This includes interpreting a rotation about the positive x -axis (or positive y -axis) as a translation in the negative y direction (or positive x direction) scaled by the distance from the viewer to the object or vice versa. Finally, the image brightness gradient vectors can line up coincidentally in some directions that make the situation more ambiguous. In the extreme case, the lack of brightness gradients will make the problem highly ambiguous. Therefore, for any success, we are restricted to the images of objects that are strongly textured with no particular regularity in their texture patterns.

7.3 Selected Examples

Suppose an arbitrary point in the image with coordinate $\tilde{\mathbf{t}}'$ is chosen as a possible solution of the true FOE. The translation velocity corresponding to this point, the assumed FOE, is given by

$$\mathbf{t}' = t'_3 \begin{pmatrix} \tilde{\mathbf{t}}' \\ 1 \end{pmatrix}.$$

Assuming that the rotational motion is either known or zero, it is then possible to compute the depth values from

$$Z' = -\frac{1}{(c + \mathbf{v} \cdot \boldsymbol{\omega})} (\mathbf{s} \cdot \mathbf{t}').$$

Since we are only interested in the regions of positive and negative depth values, we only need to compute the sign of the depth values from the above equation; that is,

$$\text{sgn}(Z') = -\text{sgn}(c + \mathbf{v} \cdot \boldsymbol{\omega}) \cdot \text{sgn}(\mathbf{s} \cdot \mathbf{t}').$$

Through examples, we show that it is possible to determine the location of the true FOE from the distribution of the clusters of positive and negative depth values around F' .

In these examples, synthetic data were used so that the underlying motion is known exactly. The focal length is assumed to be unity and the image plane is a unit square quantized to 64 image points in both x and y directions. The positive x -axis points toward right and the positive y -axis points downward. The positive depth values and the spatial brightness derivatives were chosen randomly. The depth values vary in a range of one to nine units. The time derivative of the image brightness function was then computed from the brightness change constraint equation,

$$c = -(\mathbf{v} \cdot \boldsymbol{\omega} + \frac{1}{Z} (\mathbf{s} \cdot \mathbf{t})).$$

To simulate the noise effect, random noise was added to each of E_r and $c = E_t$. These were then used as image data.

7.3.1 Example One: Focus of expansion in the Image

In this example, we consider an observer approaching a scene; the motion parameters are $\omega = (0, 0, 0)$ and $t = (0, 0, 1)$; that is, the focus of expansion is at the center of the image plane. Figure 7-8 shows the regions of negative (white) and positive (black) depth values for several assumed FOE's. The diagrams shown in columns one through four show the results when the added noise has a mean of about 20%, 40%, 60%, and 80%, respectively. These plots show that the negative depth values are clustered around the line from the assumed FOE to the true FOE. Also, the positive depth values are clustered around the continuation of this line in either direction. Using these maps, it is possible to estimate the location of the true FOE with good accuracy with as much as 80% noise in the data.

7.3.2 Example Two: Focus of Expansion at Infinity

In this example, $\omega = (.1, 0, 0)$ and $t = (0, -1, 0)$, that is, the focus of expansion is at infinity along the negative y -axis. The rotational component of motion is assumed to be known. Figure 7-9 shows the regions of negative (white) and positive (black) depth values for several assumed FOE's with random noise added to the brightness derivatives. The diagrams shown in columns one through four show the results when the added noise has a mean of about 20%, 40%, 60%, and 80%, respectively. We can determine the direction toward the true FOE at infinity with good accuracy with as much as 40% noise in the data. The results deteriorate to some extent with 60% noise for some of the assumed FOE's. With 80% noise in the data, it is hard to define the clusters of positive and negative depth values. This implies that the histograms for the orientation of the FOE constraint lines may not have distinct peaks; thus, the FOE cannot be located accurately.

7.3.3 Example Three: Unknown Rotation

In this example, we investigate the sensitivity of the solution to local variations in the rotational parameters. The motion is toward the scene with no rotation (as in example one) so that the true FOE is at the origin of the image plane. The depth values vary in a range from one to nine units with an average of about five units. To study the dependency of the solution on the choice of the rotational vector, the procedure given in the previous examples was repeated for six values of the rotational vector with 40% noise in the data. The results are shown in Figure 7-10. Again, the regions with negative depth are shown in white and the regions with positive depth are in black.

The first column shows the results for an assumed rotation of $\omega' = (.05, 0, 0)$. These results show that the estimated FOE is located on the positive y -axis around $y = 0.25$ (since the axis of symmetry of the clusters of negative depth values for the assumed FOE's intersects the y -axis around $y = 0.25$). Interestingly, this is consistent with a translation of $t' = (0, .25, 1)$. Therefore, we have overestimated the rotation about the x -axis by 0.05 radians and the translation along y -axis by about 0.25 units. We explained that with

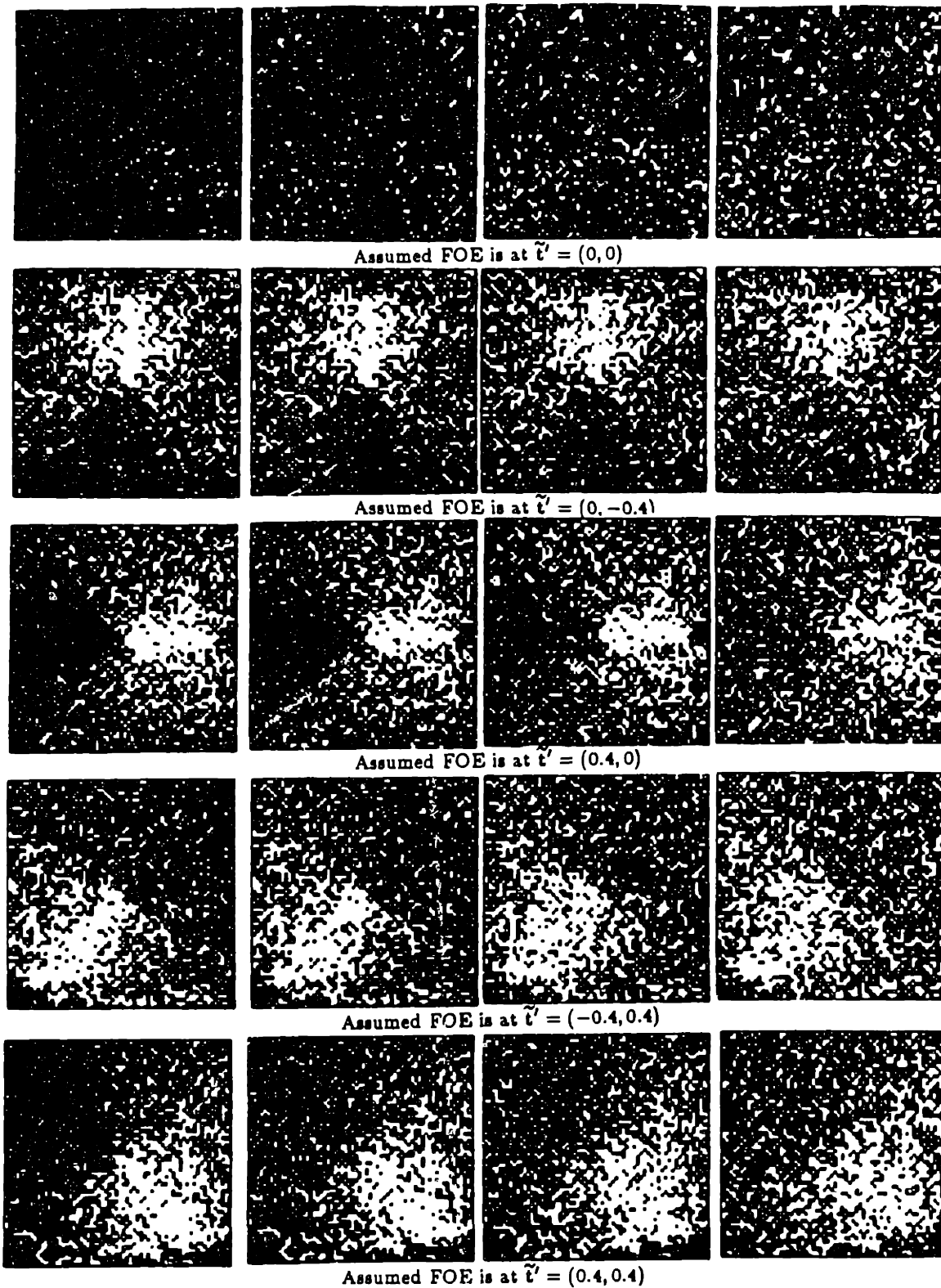


Figure 7-8. Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example one

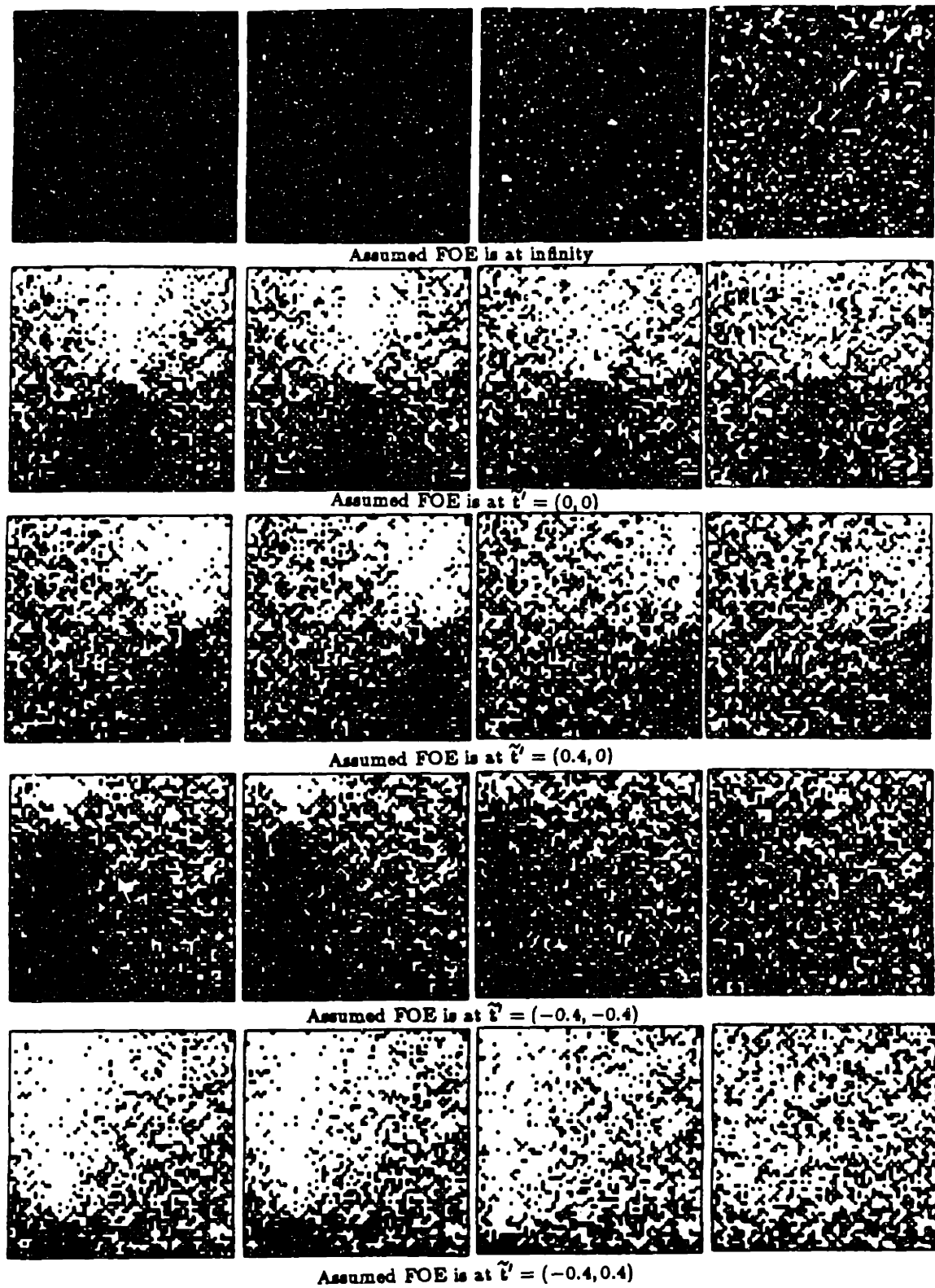


Figure 7-9. Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example two

noisy data, it is possible to interpret a rotation about positive x -axis as a translation in the negative y -axis scaled by the distance of the object from the viewer (note that the average of depth values is about five units). In this case, we need to add a translation in the positive y direction to offset the rotation about the positive x -axis.

The second column shows the results for an assumed rotation of $\omega' = (-.05, 0, 0)$. In this case, the estimated FOE is along the negative y -axis at a distance of about 0.25 units from the origin. This is consistent with a translation vector $t' = (0, -0.25, 1)$. The same conclusion as in the previous case can be made; we need to add a translation in the negative y direction to offset the rotation about the negative x -axis.

The third column shows the results for an assumed rotation of $\omega' = (0, 0.05, 0)$. In this case, the FOE constraint lines do not seem to have a common intersection point. This is good news since the conclusion is that the assumed rotation is an incorrect one. The situation is different for $\omega' = (0, -0.05, 0)$ (the results are shown in the fourth column). In this case, the axes of symmetry of the negative depth clusters seem to intersect around the point $\tilde{t}' = (0.25, 0)$. This is consistent with a translation of $t' = (0.25, 0, 1)$. Again, with noisy data, it is possible to interpret a rotation about negative y -axis as a translation in the negative x direction scaled by the distance of the object from the viewer. In this case, we need to add a translation in the positive x direction to offset the rotation about the negative y -axis.

The remaining plots from the leftmost column to the rightmost column (Figure 7-10, continued) are for an assumed rotation of $\omega = (0, 0, 0.05)$, $\omega = (0, 0, -0.05)$, $\omega = (0, 0, 0.1)$, and $\omega = (0, 0, -0.1)$, respectively.

A careful review of these plots reveals that, for each assumed rotation, the FOE constraint lines do not intersect at a common point, but seem to intersect on a circle centered at the origin with radius proportional to the assumed rotation about the optical axis. To explain this, we need to remember that a rotation about the optical axis generates motion field vectors that are tangent to concentric circles with center at the FOE (origin in this case). For a rotation of the viewer about the positive z -axis (the optical axis) the motion field vectors travel counterclockwise. Conversely, they are clockwise for a rotation about the negative z -axis. Take rotation about the positive z -axis, for example (results shown in the first and third columns). Along the negative y -axis (remember this points upward) the motion field vectors point from right to left (and increase in magnitude linearly with y). This is indicated by the shift in the negative depth cluster toward the negative x -direction (second row in the first and third columns). Along the positive x direction, these vectors point upward (and increase in magnitude linearly with x). This appears as an upward shift in the negative depth cluster (third row in the first and third columns). The same behavior is observed in the plots in the last two rows of the first and third columns. In each case, the negative depth cluster is shifted somewhat in the direction consistent with a rotation about z -axis. This implies, as mentioned earlier, that the axes of symmetry of these clusters do not really intersect at a common

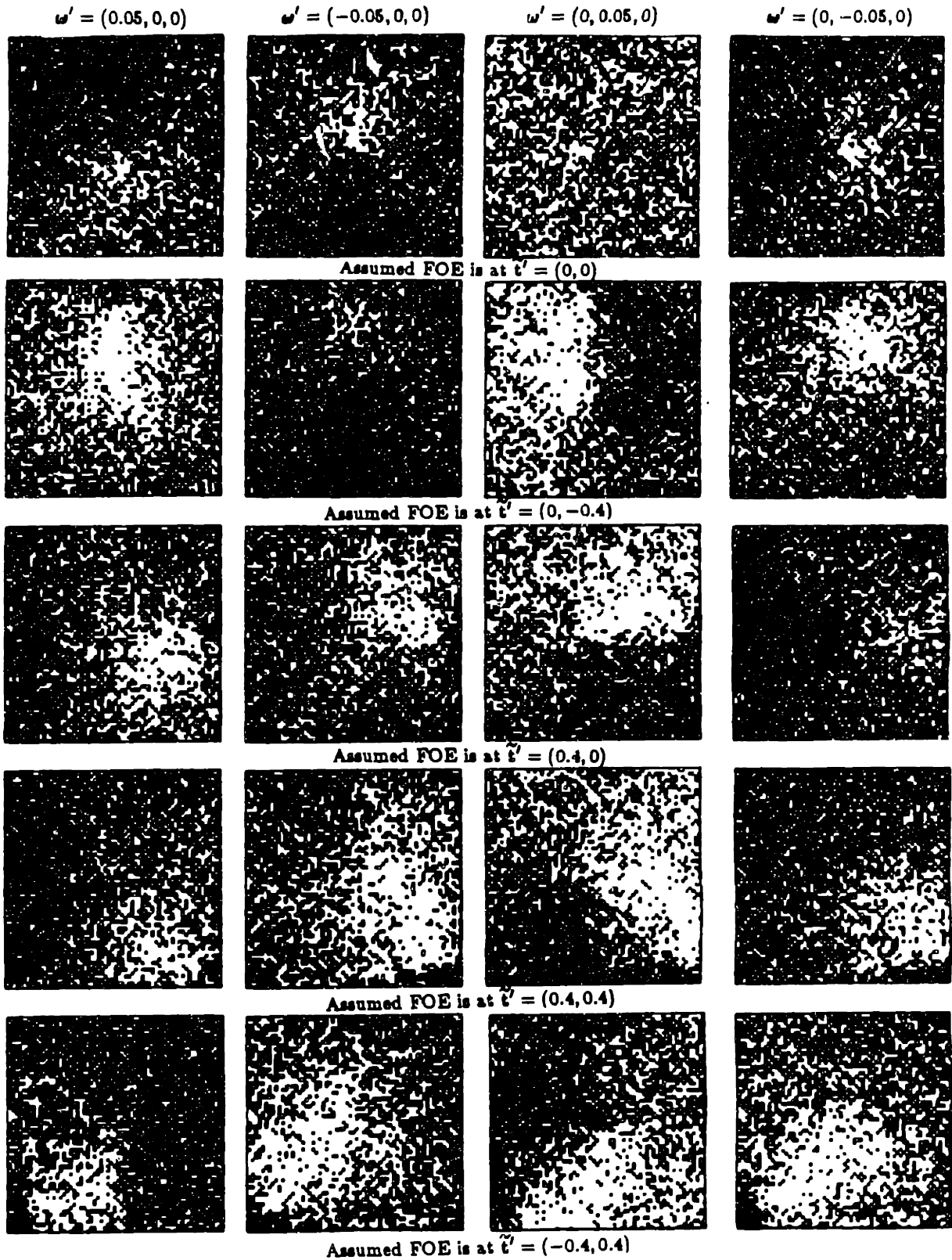


Figure 7-10. Positive (black) and negative (white) depth Regions with noise added to brightness derivatives; example three

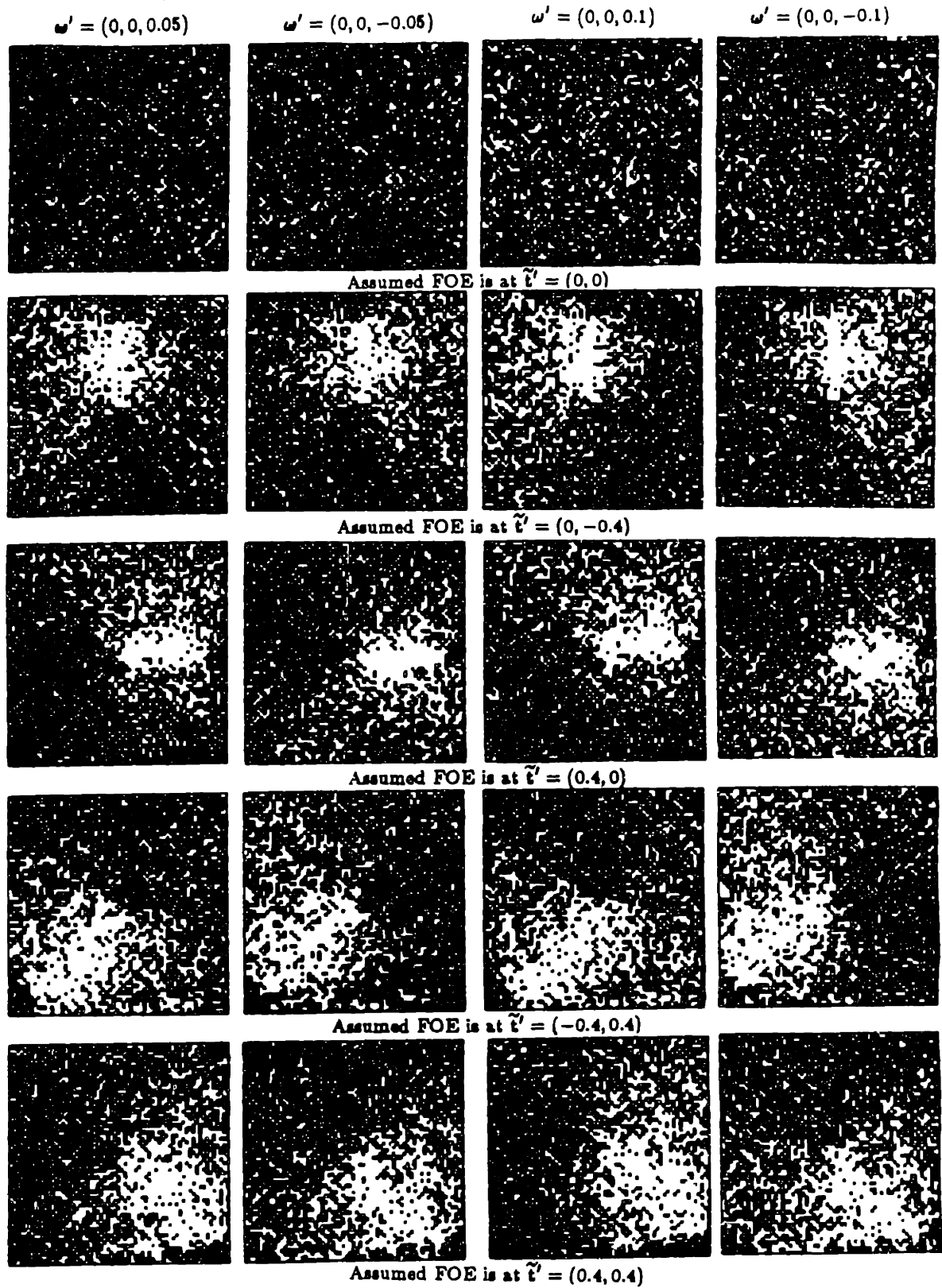


Figure 7-10. Continued.

point (the origin) because of the shifts, but rather intersect at several points that are located approximately on a circle with center at the origin and radius proportional to the magnitude of the assumed rotation. The plots in the second and fourth columns for a rotation about the negative z -axis show a similar behavior except that the shifts are now in the opposite directions. Therefore, we expect that the axes of symmetry of the clusters intersect almost at a common point when the magnitude of rotation about z -axis tends toward zero; that is, the correct rotation is assumed (see also the plots given in the first example).

7.4 Summary

In this section, we have shown how to exploit the positivity of depth as a constraint in order to estimate the location of the focus of expansion when the motion is either purely translational or the rotational component is known. The approach is based on the fact that when an arbitrary point in the image is chosen as the FOE, the depth values that are computed based on the assumed FOE tend to form clusters of positive and negative values around the line that connects the assumed FOE to the true FOE, the line that we referred to as an FOE constraint line. These clusters that are relatively symmetrical with respect to the FOE constraint line can be used to determine the direction toward the FOE; that is, the orientation of the FOE constraint line. By finding the common intersecting of several constraint lines, it is possible to obtain a reasonable estimate of the true FOE. In two selected examples, we showed that when the rotation is known, the method we suggested can give a good estimate of the location of the FOE in the presence of noisy data (noise with a mean of as much as 60%).

When the rotational component is not known (and is non-zero), these constraint lines need not have a common intersection point. This is reminiscent of the fact that motion field vectors do not intersect at a common point (FOE) when the viewer rotates about some axis through the viewing point as well as translates along some arbitrary direction. In this case, we proposed a method based on discounting the component due to rotation (by assuming some arbitrary rotation) before we apply the method of pure translation. Ideally, a reasonable estimate of FOE is obtained only when a correct rotation is assumed; this corresponds to a distinct optimum solution. We have not implemented the method to evaluate the accuracy of the solution; however, we presented an example to demonstrate the behavior of the solution, with noisy data, where the rotation vector was varied locally. The results showed some of the difficulties we have to deal with in estimating 3-D motion when the rotational component of motion is unknown. For example, several interpretations were possible (based on a qualitative analysis) related to the ambiguity in distinguishing rotation from translation (appropriately scaled by the distance of the viewer from the scene). The multiple interpretations, however, were consistent with those usually obtained from the corresponding noisy two-dimensional motion field.

Application of Regularization to Motion Vision

Suppose $y = Ax$ represents the relationship between the measurements y and an unknown parameter vector x . According to Hadamard [1923], the problem of recovering x from y is well-posed when its solution

- (1) exists,
- (2) is unique, and
- (3) depends continuously on the data.

Otherwise, the problem is ill-posed.

Most early vision problems (for example, shape from shading, shape from contours, stereomatching, computation of lightness or optical flow, or visual surface interpolation from sparse data) are ill-posed in the sense of Hadamard since the solution is usually non-unique and/or non-robust with noise in the images.

In the general case, the problem of motion vision is also ill-posed since we do not have enough constraints to determine the solution uniquely; at each point in the image, the brightness change constraint equation,

$$c + v \cdot \omega + d(s \cdot t) = 0,$$

provides one constraint on the six motion parameters (three components of each of ω and t) and the depth value $Z = 1/d$ (the component along the optical axis of the distance from the viewer to the point on the surface) corresponding to that image point. With n data points, we have n constraint equations. We seek to recover $n + 5$ unknowns, n depth values and six motion parameters up to the scale-factor ambiguity. A reasonable approach to resolve this problem is to identify physical constraints that make the problem solvable.

For example, we may assume that the surface of the scene is planar (either globally or piecewise) in order to recover the motion parameters (this problem was addressed in Chapter 4).

Another common approach pursued by some researchers in solving certain ill-posed early vision problems is the use of variational principles to incorporate certain physical constraints in the problem formulation. For example, the algorithms for computing optical flow (Horn & Schunck [1981], or Hildreth [1983]), shape from shading (Ikeuchi & Horn [1981], Horn & Brooks [1985]), lightness (Horn [1984]), or surface interpolation (Grimson [1981], Terzopoulos [1984a]) are based on determining the smoothest solution that is consistent with the available data. The smoothness assumption is incorporated in the formulation by minimizing a functional, the so-called *regularizing* (or *stabilizing*) *functional* (a functional is a mapping from the space of functions into the space of real numbers), that is some measure of departure from smoothness. Sometimes, the regularization is used to make the solution well-behaved and robust as in the formulation given by Yasumoto & Medioni [1985] for estimating 3-D motion from dense optical flow data. They extend the well-posed formulation of Bruss & Horn [1983] by introducing an additional term in the error function that is to be minimized; this term can be viewed as a regularization term. Poggio & Torre [1984] give a detailed review of the regularization theory and its application in some early vision problems. Here, we briefly show how the regularization method can be used to formulate a well-posed problem in our application.

8.1 Regularizing an Ill-Posed Problem

Suppose the problem of recovering the 2-D function $Z(x, y)$ from the measurements y_i , $i=1, 2, \dots, n$, is under-constrained, where

$$y_i = g_i(Z(x, y)), \quad i = 1, 2, \dots, n.$$

That is, suppose that there are not enough constraints to determine the solution uniquely. This implies that many solutions can be found that minimize some norm that is a measure of discrepancy between the observed data and that predicted by the solution,

$$\sum_{i=1}^n |y_i - g_i(Z)|.$$

In order to “regularize” the problem, we need to specify a regularizing functional,

$$\iint_I |f(Z(x, y))| dx dy,$$

where f is some operator (functional). The problem is then formulated as that of choosing Z which minimizes

$$e = \sum_{i=1}^n |y_i - g_i(Z)| + \lambda \iint_I |f(Z)| dx dy,$$

where λ is referred to as the *regularization parameter*. It is chosen to weigh the error due to the discrepancy between the observed and predicted data against that due to the regularizing functional. The choice of the operator f usually reflects some a priori knowledge of the nature of the solution. In most applications, the norm is chosen to be quadratic and the operator f is linear. The cost function is then written

$$e^2 = \sum_{i=1}^n |y_i - g_i(Z)|^2 + \lambda \iint_I |f(Z)|^2 dx dy.$$

For quadratic functionals, the solution is unique provided that certain conditions are satisfied (this usually requires that the appropriate boundary conditions be specified).

8.2 Imposing Surface Smoothness

When the local variations in depth are negligible relative to the absolute distance of the scene from the viewer, it is reasonable to assume that the surface of the scene varies smoothly. The discontinuities in depth usually correspond to occluding boundaries of different objects when the scene consists of more than one object. Therefore, we only need to assume that the surfaces in the scene vary smoothly within occluding boundaries. In most cases, however, the locations of the occluding boundaries are not known a priori. Thus, we somehow have to segment the image into regions within which depth is expected to vary smoothly. For example, we may initially assume that the occluding boundaries coincide with the intensity edges; that is, where the brightness gradient undergoes sharp changes or the Laplacian of the Gaussian operator applied to the image crosses zero. Of course, not every intensity edge (zero-crossing boundary) corresponds to an occluding boundary or depth discontinuity. Therefore, once the depth values are computed, it may be necessary to patch together neighboring segments that have "similar" depth values (this segmentation may be too fine since there is usually little information within zero-crossing boundaries).

The smoothness assumption can be imposed through the choice of an appropriate functional. More precisely, the measure of smoothness is governed by the specific choice of the regularizing functional. For example, if we want to minimize the local variations in depth in some region I , we can minimize the integral of the square of the gradient of depth in that region,

$$e_{r_1} = \iint_I \left(\left(\frac{\partial Z}{\partial X} \right)^2 + \left(\frac{\partial Z}{\partial Y} \right)^2 \right) dX dY.$$

To formulate the problem properly, we also need to specify the appropriate boundary condition (for example, $Z(X, Y)$ on the boundary of I). Alternatively, we may want to minimize the local surface curvature. If we choose the Euclidean (or Frobenius) norm of the Hessian of the surface as a measure of the local curvature, then we can minimize

$$e_{r_2} = \iint_I \left(\left(\frac{\partial^2 Z}{\partial X^2} \right)^2 + 2 \left(\frac{\partial^2 Z}{\partial X \partial Y} \right)^2 + \left(\frac{\partial^2 Z}{\partial Y^2} \right)^2 \right) dX dY.$$

In this case, the appropriate boundary conditions may include specifying Z and its first derivatives (gradient of Z) on the boundary of region I .

It is important to choose an appropriate functional that is consistent with the physical constraints of the problem. A detailed treatment of how to choose the appropriate functional for surface reconstruction is given in Grimson [1981]. Here, we simply show how these results can be incorporated in our formulation.

8.2.1 Solution Through Variational Principles

Suppose we seek a 2-D function Z that minimizes a certain functional. In our application, Z is the depth function. Since we are dealing with images, it is more convenient to represent Z as a function of the image coordinates x and y .

Let us assume that the functional depends only on Z and its first derivatives; that is, we can write the functional in the form

$$e_r = \iint_I f(x, y, Z, Z_x, Z_y) dx dy,$$

where

$$Z_x = \frac{\partial Z}{\partial x} \quad \text{and} \quad Z_y = \frac{\partial Z}{\partial y}.$$

Using the method of variational calculus, an extremum of e_r satisfies the *Euler-Lagrange equation* (Courant & Hilbert [1962]),

$$f_Z - \frac{\partial f_{Z_x}}{\partial x} - \frac{\partial f_{Z_y}}{\partial y} = 0,$$

where

$$f_Z = \frac{\partial f}{\partial Z}, \quad f_{Z_x} = \frac{\partial f}{\partial Z_x}, \quad \text{and} \quad f_{Z_y} = \frac{\partial f}{\partial Z_y}.$$

In general, the depth values (and/or their derivatives) are not known on the boundaries. In the absence of prescribed boundary conditions, we must introduce the *natural boundary condition* on Z . This is given by

$$f_{Z_x} y_s - f_{Z_y} x_s = 0,$$

where $[x_s, y_s]^T$ is a unit vector tangent to the boundary. The problem as formulated is referred to as a *boundary value problem*.

We can derive similar results when the functional e_r depends also on the second-order derivatives of the function Z (as in e_{r_2}); that is,

$$e_r = \iint_I f(x, y, Z, Z_x, Z_y, Z_{xx}, Z_{xy}, Z_{yy}) dx dy.$$

In this case, the Euler-Lagrange equation is given by

$$f_z - \frac{\partial f_{z_x}}{\partial x} - \frac{\partial f_{z_y}}{\partial y} + \frac{\partial^2 f_{z_{xx}}}{\partial x^2} + \frac{\partial^2 f_{z_{yy}}}{\partial x \partial y} + \frac{\partial^2 f_{z_{yy}}}{\partial y^2} = 0.$$

In general, when the functional f contains derivatives of the function $Z(x, y)$ up to some order m , the Euler-Lagrange equation is given by

$$f_z - \left(\frac{\partial f_{z_x}}{\partial x} + \frac{\partial f_{z_y}}{\partial y} \right) + \left(\frac{\partial^2 f_{z_{xx}}}{\partial x^2} + \frac{\partial^2 f_{z_{yy}}}{\partial x \partial y} + \frac{\partial^2 f_{z_{yy}}}{\partial y^2} \right) \\ + (-1)^m \left(\frac{\partial^m f_{z_{x \dots x}}}{\partial x^m} + \dots + \frac{\partial^m f_{z_{y \dots y}}}{\partial y^m} \right) = 0.$$

A particular solution can be chosen from the manifold of all possible solutions of the above equation by imposing the proper boundary conditions. For a well-posed problem, we usually need to specify Z and all but the highest derivative occurring in the regularization functional on the boundary of region I .

Both the degree of smoothness imposed on the surface, and the accuracy of the estimate of the surface structure are related to the *order* of the functional e_r ; by this we mean the highest order of the derivatives present in the functional. In theory, a better estimate of the true solution may be obtained when the functional f includes the higher-order derivatives of the function Z , but this involves specifying more boundary conditions than may be feasible in practice. The more important issue, however, is that of computational complexity and numerical instability, especially in the presence of noise.

For the sake of argument, let us assume that f is a quadratic functional of order m . Then, the resulting Euler-Lagrange equation is a linear differential equation of order $2m$. Since we are interested in the solution on a grid, this can be translated into solving an equivalent set of linear difference equations after approximating the derivatives by the appropriate finite differences. The resulting system of equations are generally solved iteratively. In the new formulation, each difference equation involves the depth values of a grid point and some surrounding neighboring points. The size of the neighborhood, the *region of support*, will depend on the order of the differential equation and the particular finite difference approximation of the derivatives. The size of the region of support usually increases with m and, therefore, more computation may be required to update the depth values in an iterative scheme.

The iterative scheme requires updating the depth value at some grid point (i, j) , denoted by $Z_{i,j}$, from an equation of the form

$$Z_{i,j}^{k+1} = \sum_{p=1}^{p_{\max}} \sum_{q=1}^{q_{\max}} a_{i,j,p,q} Z_{p,q}^k.$$

The superscript k denotes the depth value at the k^{th} iteration. Also, p_{\max} and q_{\max} are the number of grid points in the x and y directions, respectively. The coefficients $a_{i,j,p,q}$ are mostly zero except for points in the region of support of the grid point (i, j) .

In general, the iterative methods for solving these equations have a slow convergence rate and can become unstable for large m . The issue of the slow convergence rate was addressed by Terzopoulos [1984a] in his extensive work on employing multi-grid relaxation methods to solve the surface interpolation problem. His method can be applied to other early vision problems that involve solving similar types of differential or difference equations (Terzopoulos [1984b]). The instability issue which is inherent to such numerical methods is a problem that may not be resolved using the existing methods for solving higher-order partial differential equations. Thus, even though higher-order regularization functionals may give a better estimate of the true solution, we are restricted by the computation complexity and instability issues that become inevitable. In addition, the behavior of the solution for large m in the presence of noise is very unsatisfactory. The solution becomes very oscillatory in response to the high frequency noise in the images. This is very similar to the problem we encounter when a high-order polynomial is fitted to noisy data.

8.3 Physical Interpretation of the Regularizing Functionals

It is important to be able to find a connection between a mathematically formulated problem and a physical phenomenon. The physical interpretation of the problem may allow us to determine if it is formulated properly and if a unique solution exists.

The regularization functionals given by e_{r_1} and e_{r_2} represent the potential energy of deformation of a membrane and a plate, respectively (see Courant & Hilbert [1962] for a detailed treatment of this topic). The minimization problem is then viewed as that of finding the stable state of the body; that is, the state of minimum energy.

The membrane can be viewed as a surface with potential energy proportional to change in its area; the proportionality factor depends on the thickness and the elastic property of the membrane. Suppose the region of interest, I , is the area covered by the membrane at rest and $Z(x, y)$ is the deformation in the z direction. If $|Z_x| \ll 1$ and $|Z_y| \ll 1$, then the expression for the area,

$$\iint_I (1 + Z_x^2 + Z_y^2)^{\frac{1}{2}} dx dy,$$

may be approximated by

$$\iint_I (1 + \frac{1}{2}(Z_x^2 + Z_y^2)) dx dy.$$

Therefore, the potential energy is proportional to

$$\iint_I (Z_x^2 + Z_y^2) dx dy.$$

This is the functional e_{r_1} given earlier.

Instead, suppose that the membrane is subject to external forces with density $p_i(x, y)$ in the interior, and external forces with linear density $p_b(s)$ on the boundary. Further, let the boundary, denoted by ∂ , be tied to its rest position by elastic forces characterized by a modulus of elasticity of linear density $\sigma(s)$. Then, the total potential energy of the system with deformation $Z(x, y)$ is given by

$$\frac{1}{2}\mu \iint_I (Z_x^2 + Z_y^2) dx dy + \iint_I p_i(x, y) Z dx dy + \int_{\partial} (p_b(s) Z + \frac{1}{2}\sigma(s) Z^2) ds,$$

where μ is the *Poisson's ratio*, a quantity that depends on the material of the plate.

The Euler-Lagrange equation in this case is

$$\Delta Z = p_i(x, y),$$

where ΔZ is the Laplacian of Z given by

$$\Delta Z = Z_{xx} + Z_{yy}.$$

The natural boundary condition can be written

$$(Z_x y_s - Z_y x_s) + \sigma(s) Z + p_b(s) = 0.$$

The unit vector $[x_s, y_s]^T$ is tangent to the boundary, therefore, the unit vector $[y_s, -x_s]^T$ is in the direction of the outward normal to the boundary. We can alternatively write the boundary condition as

$$\frac{\partial Z}{\partial n} + \sigma(s) Z + p_b(s) = 0,$$

where $\partial/\partial n$ is the derivative in the direction of the outward normal to the boundary.

When $\sigma = 0$, the problem does not usually possess a stable solution. This is best explained using a physical argument; the case $\sigma = 0$ corresponds to a freely moving membrane subject to arbitrary external forces. The membrane does not possess a stable state unless the external forces balance; that is, unless

$$\iint_I p_i(x, y) dx dy + \int_{\partial} p_b(s) ds = 0.$$

In the absence of external boundary forces this implies that for a stable state (in our terms, a unique solution) we should have

$$\iint_I p_i(x, y) dx dy = 0.$$

Even when the forces balance, the solution is non-unique unless the membrane is fixed in space. This can be seen easily if we add an arbitrary constant Z_0 to Z . The new expression for the total potential energy,

$$\frac{1}{2}\mu \iint_I (Z_x^2 + Z_y^2) dx dy + \iint_I p_i(Z + Z_0) dx dy + \int_{\partial} p_b(Z + Z_0) ds,$$

can be simplified to

$$e_a + \frac{1}{2}\mu \iint_I (Z_x^2 + Z_y^2) dx dy + \iint_I p_i Z dx dy + \int_{\partial} p_b Z ds,$$

where $e_a = 2AZ_0$ is a constant (A denotes the area of region I). Therefore, adding a constant to Z is the same as adding a constant to the total potential energy of the system. Consequently, the shape of the solution remains unchanged. To obtain a unique solution, it is common to fix a point on the surface, for example, by requiring that

$$\frac{1}{A} \iint_I Z dx dy = Z_0,$$

where Z_0 is some arbitrary constant. This means that the center of mass Z_c is at a rest position ($Z_c = Z_0$).

The case of the thin plate is as follows. A plate is an elastic 2-D body whose potential energy of deformation is given by the integral of some quadratic functional of the principal curvatures of the plate. If the principal curvature radii of the deformed plate are given by ρ_1 and ρ_2 , then the potential energy density is

$$k_1 \left(\frac{1}{\rho_1^2} + \frac{1}{\rho_2^2} \right) + 2k_2 \frac{1}{\rho_1 \rho_2},$$

where the constants k_1 and k_2 depend on the material of the plate. When Z , Z_x , Z_y , ..., are small, we can write

$$\left(\frac{1}{\rho_1} + \frac{1}{\rho_2} \right) = \Delta Z \quad \text{and} \quad \frac{1}{\rho_1 \rho_2} = Z_{xx} Z_{yy} - Z_{xy}^2.$$

Finally, the potential energy of deformation is proportional to

$$\iint_I ((\Delta Z)^2 - 2(1 - \mu)(Z_{xx} Z_{yy} - Z_{xy}^2)) dx dy.$$

The natural boundary conditions in the absence of boundary forces and moments are given by

$$\Delta Z - (1 - \mu)(Z_{xx}x_s^2 + 2Z_{xy}x_s y_s + Z_{yy}y_s^2) = 0,$$

$$\frac{\partial}{\partial n}(\Delta Z) + (1 - \mu)\frac{\partial}{\partial s}(Z_{xx}x_n x_s + Z_{xy}(x_n y_s + x_s y_n) + Z_{yy}y_n y_s) = 0,$$

where $\partial/\partial s$ and $\partial/\partial n$ are the derivatives in the direction of the tangent and outward normal to the boundary, respectively.

8.4 Recovering Motion and Depth: General Formulation

We can incorporate the smoothness assumption through the choice of an appropriate functional in terms of Z and its derivatives. Since we deal with image data, it is easier to write Z as a function of the image coordinates $Z = Z(x, y)$. Furthermore, since the brightness change equation is in terms of $d = 1/Z$, we let $d(x, y)$ represent the surface function. Here, we restrict ourselves to the two types of regularizing functionals that represent the potential energy of deformation in membranes and thin plates with Z replaced by $d = 1/Z$. For now, let $f = f(d, d_x, d_y, d_{xx}, d_{xy}, d_{yy})$ denote the regularization functional. Then we can formulate the problem as that of finding the motion parameters, ω and t , and the inverse depth function, $d(x, y)$, that minimize the functional

$$J = \iint_I (c + \mathbf{v} \cdot \omega + d \mathbf{s} \cdot \mathbf{t})^2 dx dy + \lambda \iint_I f(d, d_x, d_y, d_{xx}, d_{xy}, d_{yy}) dx dy$$

over the relevant region of the image. The coefficient λ is the regularization parameter that is chosen to scale the smoothness term with respect to the data term. In order to pose the problem properly, we need to impose the natural boundary conditions on d and its derivatives.

As we see next, this can be reduced to solving a set of simultaneous algebraic and partial differential equations through an iterative scheme. The motion parameters are computed from the algebraic equations and the partial differential equation is solved for the surface function.

8.4.1 Recovering Motion

For the extremum of J we must have

$$\frac{\partial J}{\partial \omega} = 0 \quad \text{and} \quad \frac{\partial J}{\partial t} = 0.$$

Performing the above differentiation we obtain

$$\iint_I \mathbf{v} (c + \mathbf{v} \cdot \omega + d (\mathbf{s} \cdot \mathbf{t})) dx dy = 0 \quad \text{and} \quad \iint_I d \mathbf{s} (c + \mathbf{v} \cdot \omega + d (\mathbf{s} \cdot \mathbf{t})) dx dy = 0.$$

We can write the above equations in the form

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \omega \\ t \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where

$$\mathbf{d}_1 = \iint_I c \mathbf{v} dx dy, \quad \mathbf{d}_2 = \iint_I d c \mathbf{s} dx dy,$$

$$\mathbf{M}_1 = \iint_I (\mathbf{v} \mathbf{v}^T) dx dy, \quad \mathbf{M}_2 = \iint_I d (\mathbf{v} \mathbf{s}^T) dx dy, \quad \text{and} \quad \mathbf{M}_4 = \iint_I d^2 (\mathbf{s} \mathbf{s}^T) dx dy.$$

Assuming that depth is known, this linear matrix equation can be solved for ω and t . The solution can be expressed analytically in several forms; for example, we can write

$$t = (M_4 - M_2^T M_1^{-1} M_2)^{-1} (M_2^T M_1^{-1} d_1 - d_2),$$

$$\omega = -M_1^{-1} (d_1 + M_2 t).$$

8.4.2 Recovering Depth

We can use the Euler-Lagrange equation in order to derive the partial differential equation that is solved for the inverse depth function d . If d represents the deformation of a membrane, then the regularizing functional is the potential energy of deformation due to small deflections of a membrane,

$$e_r = \iint_I (d_x^2 + d_y^2) dx dy.$$

The expression for J is given by

$$J = \iint_I (c + \mathbf{v} \cdot \omega + d \mathbf{s} \cdot \mathbf{t})^2 dx dy + \lambda \iint_I (d_x^2 + d_y^2) dx dy,$$

the resulting Euler-Lagrange equation is *Poisson's* equation,

$$\Delta d = \frac{1}{\lambda} (\mathbf{s} \cdot \mathbf{t}) (c + \mathbf{v} \cdot \omega + d \mathbf{s} \cdot \mathbf{t}),$$

and the natural boundary condition is given by

$$d_x y_s - d_y x_s = 0 \quad \text{or} \quad \frac{\partial d}{\partial \mathbf{n}} = 0.$$

If d represents the deformation in a thin plate, then the regularization functional is

$$e_r = \iint_I ((\Delta d)^2 - 2(1 - \mu)(d_{xx}d_{yy} - d_{xy}^2)) dx dy,$$

and J can be written as

$$J = \iint_I (c + \mathbf{v} \cdot \omega + d \mathbf{s} \cdot \mathbf{t})^2 dx dy + \lambda \iint_I ((\Delta d)^2 - 2(1 - \mu)(d_{xx}d_{yy} - d_{xy}^2)) dx dy.$$

In this case, the Euler-Lagrange equation reduces to

$$\Delta \Delta d + \frac{1}{\lambda} (\mathbf{s} \cdot \mathbf{t}) (c + \mathbf{v} \cdot \omega + d \mathbf{s} \cdot \mathbf{t}) = 0,$$

where

$$\Delta \Delta d = d_{xxxx} + 2d_{xxyy} + d_{yyyy}$$

is the biharmonic operator applied to d . The natural boundary conditions are

$$\Delta d - (1 - \mu)(d_{xx}x_s^2 + 2d_{xy}x_s y_s + d_{yy}y_s^2) = 0,$$

$$\frac{\partial}{\partial n}(\Delta d) + (1 - \mu)\frac{\partial}{\partial s}(d_{xx}x_n x_s + d_{xy}(x_n y_s + x_s y_n) + d_{yy}y_n y_s) = 0,$$

Note that the free parameter μ appears only in the boundary conditions and not in the Euler-Lagrange equation. If we set $\mu = 1$, that is, if we let

$$e_r = e_{r_1} = \iint_I (\Delta d)^2 dx dy,$$

then the boundary conditions simplify to

$$\Delta d = 0 \quad \text{and} \quad \frac{\partial}{\partial n}(\Delta d) = 0.$$

Consider the physical interpretation of the first term in the expression for J that represents the sum-squared-error in the brightness change constraint equation. This may be viewed as the potential energy due to the external forces that the interior of the membrane or thin plate is subject to. The contribution of this force to the total potential energy can be written in the form

$$\iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega})^2 dx dy + \iint_I [2(c + \mathbf{v} \cdot \boldsymbol{\omega})(\mathbf{s} \cdot \mathbf{t}) + (\mathbf{s} \cdot \mathbf{t})^2 d] d dx dy.$$

The first term is independent of d and may be ignored. The term

$$p_i = (\mathbf{s} \cdot \mathbf{t})[2(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t})d]$$

which is the error in the brightness change constraint equation scaled by $(\mathbf{s} \cdot \mathbf{t})$ can be viewed as the density of the external forces. Since the external forces generally do not balance (or the weighted error in the brightness change equation over the image region of interest do not sum up to zero), that is,

$$\iint_I p_i dx dy = \iint_I (\mathbf{s} \cdot \mathbf{t})[2(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t})d] dx dy \neq 0,$$

we conclude that the problem as formulated does not have a stable solution. As suggested earlier, we may impose the constraint that the center of mass is fixed at some position Z_0 , that is,

$$\frac{1}{A} \iint_I Z dx dy = \frac{1}{A} \iint_I \frac{1}{d} dx dy = Z_0.$$

Note that this assumption is quite appropriate since we can recover depth and the translational motion up to a scale factor. Hence, we can set the depth value at some point (in this case, the center of mass) to be equal to some value (Z_0). Alternatively, we may impose the constraint

$$\frac{1}{A} \iint_I d \, dx \, dy = d_0,$$

where d_0 is an arbitrary constant.

Given the estimate of the motion parameters, ω and \mathbf{t} , the boundary value problem (the Euler-Lagrange equation subject to the prescribed boundary conditions) can be solved for the inverse depth function $d(x, y)$.

8.4.2.1 Finite-Difference Approximation

Differential equations can be solved using either finite-element or finite-difference techniques. For the sake of completeness, we show the finite-difference approximation of the Poisson's and the biharmonic equations given in the previous section.

We consider Poisson's equation first, that is,

$$\Delta d = \frac{1}{\lambda} (\mathbf{s} \cdot \mathbf{t}) [(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t}) d].$$

Suppose we write the finite-difference approximation of the Laplacian as

$$\Delta d = \frac{1}{\delta^2} (d_{i,j+1} + d_{i+1,j} + d_{i,j-1} + d_{i-1,j} - 4d_{i,j}),$$

where δ is the grid dimension and $d_{i,j}$ denotes the inverse depth value at some grid point (i, j) . Then the discrete approximation to the Poisson's equation is given by

$$d_{i,j} = \bar{d}_{i,j} - \frac{\delta^2}{4\lambda} (\mathbf{s} \cdot \mathbf{t}) [(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t}) d_{i,j}],$$

where

$$\bar{d}_{i,j} = \frac{1}{4} (d_{i,j+1} + d_{i+1,j} + d_{i,j-1} + d_{i-1,j})$$

is the average inverse depth of the four immediate neighboring points. Poisson's equation can now be solved iteratively as follows

$$d_{i,j}^{k+1} = \bar{d}_{i,j}^k - \frac{\delta^2}{4\lambda} (\mathbf{s} \cdot \mathbf{t}) [(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t}) d_{i,j}^k].$$

On the boundary, we impose the natural boundary condition, $\partial d / \partial \mathbf{n} = 0$. Suppose we restrict the boundary to be polygonal with horizontal and vertical segments. Then the natural boundary condition on the vertical segments reduces to $d_x = 0$ and on the horizontal segments to $d_y = 0$. In the discrete domain, these can be written as $d_{i,j} = d_{i\pm 1,j}$

for the vertical boundary segments and $d_{i,j} = d_{i,j\pm 1}$ for the horizontal segments. This requires that we consider a band one grid wide beyond the region where the iterative method is applied to.

Now, consider the biharmonic equation. In this case, the inverse depth function is determined from

$$\Delta\Delta d + \frac{1}{\lambda}(\mathbf{s} \cdot \mathbf{t})[(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t})d] = 0.$$

One possible finite-difference approximation of the biharmonic operator is

$$\Delta\Delta d_{i,j} = \frac{20}{\delta^4}(d_{i,j} - \bar{d}_{i,j}),$$

where $\bar{d}_{i,j}$, the weighted average of the inverse depth values of the neighboring cells, is

$$\begin{aligned} \bar{d}_{i,j} = \frac{1}{20} & (8(d_{i+1,j} + d_{i-1,j} + d_{i,j+1} + d_{i,j-1}) \\ & - 2(d_{i+1,j+1} + d_{i-1,j-1} + d_{i-1,j+1} + d_{i+1,j-1}) \\ & - (d_{i+2,j} + d_{i-2,j} + d_{i,j+2} + d_{i,j-2})). \end{aligned}$$

Then the iterative scheme for solving the biharmonic equation becomes

$$d_{i,j}^{k+1} = \bar{d}_{i,j}^k - \frac{\delta^4}{20\lambda}(\mathbf{s} \cdot \mathbf{t})[(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t})d_{i,j}^k].$$

The natural boundary conditions are given by

$$\Delta d - (1 - \mu)(d_{xx}x_s^2 + 2d_{xy}x_s y_s + d_{yy}y_s^2) = 0,$$

$$\frac{\partial}{\partial n}(\Delta d) + (1 - \mu)\frac{\partial}{\partial s}(d_{xx}x_n x_s + d_{xy}(x_n y_s + x_s y_n) + d_{yy}y_n y_s) = 0.$$

If we set $\mu = 1$, these simplify to

$$\Delta d = 0 \quad \text{and} \quad \frac{\partial}{\partial n}\Delta d = 0.$$

The boundary condition $\Delta d = 0$ on the vertical and horizontal boundary segments reduces to $d_{xx} = 0$ and $d_{yy} = 0$, respectively. To obtain the finite difference approximation to these derivatives, we can simply write

$$\{d_{xx}\}_{i,j} = \frac{1}{\delta} \left(\frac{d_{i+1,j} - d_{i,j}}{\delta} - \frac{d_{i,j} - d_{i-1,j}}{\delta} \right)$$

or

$$\{d_{xx}\}_{i,j} = \frac{1}{\delta^2}(d_{i+1,j} - 2d_{i,j} + d_{i-1,j}).$$

Similarly, we can obtain

$$\{d_{yy}\}_{i,j} = \frac{1}{\delta^2}(d_{i,j+1} - 2d_{i,j} + d_{i,j-1}).$$

The boundary condition $\partial(\Delta d)/\partial n = 0$ on the vertical and horizontal segments reduce to $d_{xxx} = 0$ and $d_{yyy} = 0$, respectively. Repeating the procedure given above, we find that the finite difference approximation of these derivatives can be written in the form

$$\{d_{xxx}\}_{i,j}|_{\text{left}} = \frac{1}{\delta^3}(-d_{i-2,j} + 3d_{i-1,j} - 3d_{i,j} + d_{i+1,j}),$$

$$\{d_{xxx}\}_{i,j}|_{\text{right}} = \frac{1}{\delta^3}(d_{i+2,j} - 3d_{i+1,j} + 3d_{i,j} - d_{i-1,j}),$$

for the left and right vertical segments and

$$\{d_{yyy}\}_{i,j}|_{\text{top}} = \frac{1}{\delta^3}(d_{i,j+2} - 3d_{i,j+1} + 3d_{i,j} - d_{i,j-1}),$$

$$\{d_{yyy}\}_{i,j}|_{\text{bottom}} = \frac{1}{\delta^3}(-d_{i,j-2} + 3d_{i,j-1} - 3d_{i,j} + d_{i,j+1}),$$

for the top and bottom horizontal segments. These boundary conditions require that we consider a band that is two grid points wide beyond the region where the iterative scheme is applied.

8.4.3 Choice of the Regularization Parameter

The regularization parameter, λ , weighs the relative contribution of the total error in the brightness constraint equation over the image to that due to the smoothness term. It is usually made small when the data are accurate and large when they are noisy. Even with perfect data, it usually cannot be made arbitrarily small because of convergence and numerical instability considerations. To show this, let us assume that the data is perfect. Ideally, we want to have $\lambda = 0$; that is, no smoothing. Unfortunately, the problem becomes ill-conditioned—we had to impose the smoothness assumption to make it well-posed in the first place. Hence, we need to make λ as small as possible, yet, not so small that the problem is numerically ill-conditioned. This may require some heuristics. When data are noisy, we need to make λ relatively large to obtain a smooth solution.

Consider the finite-difference approximation of Poisson's equation. If $e_{i,j}^k$ denotes the error in the brightness equation at some grid point (i, j) after the k^{th} iteration,

$$e_{i,j}^k = [(c + \mathbf{v} \cdot \boldsymbol{\omega}) + (\mathbf{s} \cdot \mathbf{t})d_{i,j}^k],$$

then we can write

$$d_{i,j}^{k+1} = \bar{d}_{i,j}^k + \alpha e_{i,j}^k,$$

where

$$\alpha = \frac{\delta^2}{4\lambda}(\mathbf{s} \cdot \mathbf{t}).$$

This form of the equation shows how the iterative scheme works. At each iteration, we update the inverse depth value at some grid point according to the average inverse depth of the four neighboring points and the weighted error in the brightness change equation. If we make λ large (corresponding to a small α), then the brightness error is weighted less relative to the smoothness term (in this case, the smoothing corresponds to a local averaging of the four neighboring points). In the extreme case, when λ becomes very large (α tends to zero), the depth value is solely determined from the average of the neighboring points independent of the error in the brightness change equation. Conversely, for very small λ (very large α), the new depth value depends only on the error in the brightness change equation (the term involving the average inverse depth value of neighboring points becomes negligible). In this case, however, the problem becomes numerically unstable for the reason described earlier. For the biharmonic equation, a similar argument applies.

8.5 An Iterative Scheme

Given the equations for updating the motion parameters and the inverse depth function, an iterative scheme can be devised to recover depth and motion.

- (1) We start with an initial estimate of the depth function.
- (2) Given the updated depth function, the motion parameters are computed from the linear equations

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = - \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix},$$

where the matrices \mathbf{M}_i , $i = 2, 3, 4$ and the vector \mathbf{d}_2 , as given earlier, depend on the depth function.

- (3) Given the updated motion parameters, a new depth function is computed by solving the boundary value problem; either Poisson's equation,

$$\Delta d = \frac{1}{\lambda}(\mathbf{s} \cdot \mathbf{t})(c + \boldsymbol{\omega} \cdot \mathbf{t} + d \mathbf{s} \cdot \mathbf{t}),$$

with the natural boundary condition

$$\frac{\partial d}{\partial \mathbf{n}} = 0$$

or the biharmonic equation,

$$\Delta \Delta d + \frac{1}{\lambda}(\mathbf{s} \cdot \mathbf{t})(c + \boldsymbol{\omega} \cdot \mathbf{t} + d \mathbf{s} \cdot \mathbf{t}) = 0,$$

with the natural boundary conditions

$$\Delta d - (1 - \mu)(d_{xx}x_s^2 + 2d_{xy}x_sy_s + d_{yy}y_s^2) = 0,$$

$$\frac{\partial}{\partial n}(\Delta d) + (1 - \mu)\frac{\partial}{\partial s}(d_{xx}x_nx_s + d_{xy}(x_ny_s + x_sy_n) + d_{yy}y_ny_s) = 0.$$

- (4) We check the improvement in the solution to either stop if the solution has not improved or go to step (2).

Unfortunately, this iterative scheme is expensive computationally; solving the boundary value problem by itself generally requires a lot of computation even when multi-grid methods are used. In fact, the boundary value problem we have to solve (given the estimate of the motion parameters) is very similar to the problem of surface reconstruction from sparse depth data (Grimson [1981] or Terzopoulos [1984a]). Here, however, we have to solve the same problem over and over as the motion parameters are updated.

8.5.1 Comments on Experimental Results

We have implemented a coarse to fine strategy where the solution of the boundary value problem (solving for the inverse depth function) is first obtained on the coarsest grid (a 16×16 grid sampling) and is then propagated gradually to the finest grid (a 64×64 grid sampling). In our experiments, the number of iterations on each of the three sampling levels was usually fixed regardless of the size of the error (from 80 iterations on the coarsest level down to 20 iterations on the finest level). From the simulation results, we have learned that the iterative method almost never converges to an “acceptable” solution with arbitrary initial conditions (an acceptable solution for the surface would be one that somewhat resembles the true surface). In general, many solutions not similar to the true surface are obtained regularly depending on the initial condition. The algorithm converges to an acceptable solution only when a good estimate of either the motion parameters or the depth values is known (in our tests, this has been an estimate of motion within 20% of the solution). Even though it is hard to show uniqueness, it appears (in the cases we have tested) that the global minimum solution is close to the true solution (note that the two are not expected to be the same because of the smoothness assumption). Our conclusion based on the simulations performed is that even when the optimum solution is unique, it cannot be obtained through the iterative scheme we have suggested unless an accurate estimate of either motion or the surface structure is available.

8.5.2 Known Motion

In the limiting case when depth is known perfectly, the problem becomes that of surface reconstruction from depth data. Even though our primary goal is to recover motion and, therefore, this problem may not be very interesting to us, there are many situations where we need to determine the depth map of the environment after we have recovered

the motion by some other means. For example, consider the following two-camera set-up mounted on a vehicle. One camera is pointing towards the ground (not necessarily vertically, but perhaps at some angle), and the other is pointing straight ahead. Assuming that the first camera is viewing a planar surface, we can determine the motion of the vehicle using the method given for planar surfaces. Now, we need to recover the depth map of the scene in front of the vehicle from the images taken by the second camera.

When the motion is known, an estimate of the depth function can be obtained from the brightness change equation:

$$\tilde{d} = -\frac{c + \mathbf{v} \cdot \boldsymbol{\omega}}{\mathbf{s} \cdot \mathbf{t}},$$

or equivalently,

$$\tilde{Z} = -\frac{\mathbf{s} \cdot \mathbf{t}}{c + \mathbf{v} \cdot \boldsymbol{\omega}},$$

except where the denominator becomes zero. Here, we use \tilde{d} and \tilde{Z} to denote estimates of the true d and Z . Since the depth values are inaccurate due to noise in the images and quantization, the objective then becomes that of finding a smooth approximation to the depth function. This is similar to the problem of surface reconstruction from depth data (as addressed by Grimson [1981] or Terzopoulos [1984a]), or simply the problem of filtering noisy data.

Our objective is to determine a smooth surface (this is imposed through the regularizing functional) that is most consistent with the depth data. To be consistent, let us use d for the inverse depth function. Then, the surface reconstruction problem can be formulated as that of finding a smooth surface d most consistent with the data \tilde{d} . This is done by minimizing the functional

$$\iint_I (d - \tilde{d})^2 dx dy + \lambda^2 \iint_I f(d, d_x, d_y, \dots) dx dy,$$

where the second term is an appropriate regularization functional. The first term penalizes the discrepancy between the solution and the data and is weighted with respect to the smoothness term through the regularization parameter, λ . Substituting for \tilde{d} we obtain

$$\iint_I \left(d + \frac{c + \mathbf{v} \cdot \boldsymbol{\omega}}{\mathbf{s} \cdot \mathbf{t}}\right)^2 dx dy + \lambda \iint_I f(d, d_x, d_y, \dots) dx dy.$$

If we instead scale the first term by $(\mathbf{s} \cdot \mathbf{t})$, that is, if we choose it to be

$$\iint_I [(\mathbf{s} \cdot \mathbf{t})(d - \tilde{d})]^2 dx dy,$$

then we instead arrive at

$$\iint_I (c + \mathbf{v} \cdot \boldsymbol{\omega} + d(\mathbf{s} \cdot \mathbf{t}))^2 dx dy + \lambda \iint_I f(d, d_x, d_y, \dots) dx dy$$

which is the familiar form given earlier. A justification for scaling the error in the inverse depth function by $(s \cdot t)$ is that this roughly corresponds to giving more importance to image points with larger brightness derivatives. For a given noise level (either image noise or quantization) these points are expected to have a larger signal to noise ratio.

We conclude that the problem as formulated is, in fact, that of surface reconstruction from very dense and noisy depth data. In this case, the depth data are obtained directly from the time-varying imagery provided that a very good estimate of motion is available.

8.6 Summary

We have shown that the problem of recovering motion and depth from time-varying images can be formulated as a regularization problem by imposing smoothness on the solution of the depth function. The smoothness assumption is incorporated through the choice of an appropriate regularization functional that can be given a physical interpretation. This not only gives a physical insight to the formulation of the mathematical problem, but also allows us to formulate the problem properly. Two types of functionals were considered. These represent the potential energy of deformation of a membrane and a thin plate under external loading.

Using necessary conditions for optimality and variational principles the problem was reduced to solving a set of linear algebraic and partial differential equations iteratively. Given an estimate of the depth function, the motion parameters are determined from the algebraic equations. Using the updated estimate of the motion parameters, the depth function, in turn, is computed from the partial differential equation (either Poisson's or a biharmonic equation). The procedure is repeated until convergence is obtained. If a very good estimate of either motion or depth is not available, the iterative method requires a lot of computation. Furthermore, the algorithm may never converge to a reasonable solution, one that resembles the true solution. The optimum solution is generally somewhat different from the true solution because of the required smoothness. This is influenced by the choice of the regularization parameter λ . When data are perfect, we ideally want to make λ as small as possible so that the solution is most consistent with the image data. Unfortunately, we can not make it too small because of numerical considerations; that is, the problem becomes ill-conditioned when the smoothness term becomes negligible. Therefore, even with perfect data, the surface we obtain is usually smoother than the true surface. Interestingly, this is not a major issue with noisy data since we do not want the solution to be very consistent with the data in the first place.

As we explained, it seems unlikely that the method is very practical if a good estimate of depth or motion is not available since Poisson's equation or a biharmonic equation has to be solved several times. In many cases, it may be possible to obtain a good estimate of motion and the surface orientation by assuming that the surface is planar (solve the planar surface problem). We can then improve the solution through the method given in this chapter. Of course, this only works if the estimate of motion is within the convergence

radius of the iterative scheme. Even though it is hard to determine analytically what this is, the simulation results show that the algorithm converges to the solution when the initial guess is within 20% of the solution.

Summary and Concluding Remarks

This thesis addresses the problem of direct passive navigation. The primary goal is to (uniquely) recover the motion of an observer directly from the time-varying images of a stationary scene without the need to either compute the (local) optical flow or establish correspondence between dominant features in an image sequence. Only the direction of translation can be recovered, because of the well-known scale-factor ambiguity in motion vision; the motion field remains unchanged if we scale the depth values and the translational motion by the same (constant) factor. Once we know the motion, we can determine the scene structure; that is, the spatial arrangement of objects in the scene. The scene structure is usually defined in terms of the distances of points on the surfaces of the scene from the viewer.

We first derived the brightness change constraint equation. This equation shows how the motion of the viewer and the structure of the scene impose a constraint on the spatial/temporal variations of the image grey-levels at every image point. Consequently, it can be exploited to determine the structure of the scene and the motion of the viewer. The constraint equation was derived under the assumption that the brightness of an image point (or small patch) remains approximately constant as viewed by the observer in motion relative to the scene. For this to be true, the surface of the scene should have strong "texture" or surface markings and the lighting condition should remain relatively uniform between successive images. In order to estimate the spatial derivatives accurately, the image brightness should vary smoothly over the image or at least within the occluding boundaries. In practice, because they are band-limited, imaging devices cut off or attenuate the high frequency components of the image signals. As a result, images are analytical functions with bounded derivatives. Therefore, even in regions of sharp texture

changes on the object, image grey-level values vary relatively smoothly. To estimate the temporal derivative accurately we often need to assume small relative motion between image frames. This imposes a constraint on the required sampling rate.

We showed how to recover the motion parameters uniquely in special cases. When the motion is rotational, the three components of the rotation vector can be determined from three linear equations. When the depth values of at least 6 non-collinear image points are known, the solution of six linear equations gives the motion parameters. Similar equations have to be solved if the points in the scene are known to lie on a plane parallel to, but at some unknown distance from, the image. In general, we cannot determine three-dimensional motion uniquely from the brightness constraint equation alone; we need more constraints.

A reasonable approach to resolve this problem is to identify physical constraints that make the problem solvable. We considered several approaches in order to formulate a well-posed problem. We showed that when the surface of the scene can be parameterized (as is the case, for example, when the surface of the scene is planar or quadratic), the problem reduces to determining only a finite number of parameters. In the case of planes, we proposed a least-squares formulation; the problem was reduced to solving eight non-linear equations in terms of the motion parameters and the orientation of the plane. By exploiting the special structure of these equation, we suggested two iterative schemes for solving them. For quadratic surfaces, we need to solve 11 non-linear equations iteratively in terms of the motion parameters and the parameters of the tangent plane and the surface curvature. In either case, the unknown parameters can be estimated in closed form through a two-step procedure. We first need to compute certain intermediate parameters, the so-called essential parameters of planar or quadratic surfaces from certain moments of brightness derivatives over the whole image. In the case of planes, the essential parameters are the elements of a particular 3×3 matrix whose eigenvalue-eigenvector decomposition yields the motion parameters and the plane orientation. In the case of quadratic surfaces, the procedure involves a coordinate transformation to simplify the problem to solving a cubic equation. We explained why the analytical solution for planar surfaces gives the same estimate of the motion parameters as the iterative method and why the analytic method for quadratic surfaces, in general, does not.

An alternative approach to parameterizing the surface is regularization. Here, we impose smoothness on the surface of the scene without the need for any a priori knowledge of the type of the surface. The problem is formulated as that of minimizing a functional that penalizes the discrepancy between the observed data and that predicted by the solution over the whole image, as well as the magnitude of the functional used as measure of the departure from smoothness. The error due to the difference between the observed and predicted data is weighted against that due to smoothness assumption through the appropriate choice of a regularization parameter. Using necessary conditions for optimality and the principles of variational calculus, the problem is reduced to that of

solving a set of algebraic equations and a partial differential equation iteratively. Given an estimate of the depth function, the algebraic equations that are linear in the motion parameters are solved to obtain an estimate of these parameters. Given these, the partial differential equation is solved to update the estimate of the depth function. The procedure is repeated until convergence is obtained. The partial differential equation is typically either Poisson's equation or a biharmonic equation depending on the choice of the regularization functional, that is, the smoothness term.

The principle underlying the iterative schemes we have proposed is as follows. If we know the structure of the scene, we can recover motion rather easily from six linear equations. If we know the motion, then we can determine the structure of the scene. Through an iterative approach, we can gradually improve the estimates of motion and the structure of the scene. Depending on the method used, the scheme may or may not converge to a solution. For planar surfaces, only three linear equations have to be solved to update the structure of the scene. In the case of quadratic surfaces, this increases to six equations. In either case, the number of equations is the same as the number of parameters used to describe the scene. In theory, this number can be increased to estimate as many parameters as are necessary to describe the surface of the scene. In practice, the success of the iterative method will depend on the number of parameters, and quickly diminishes as we allow a larger number of degrees of freedom. The solution procedure given in the regularization method can be viewed as an extreme case. Here, we have to solve for every depth value and even the procedure for obtaining the depth values requires its own iterative process.

The simulation results confirm that the iterative method for planes has better convergence properties than the one for quadratic surfaces. In the cases we have tested, the algorithm for planes has always converged to the global minimum, whereas the one for quadratic surfaces sometimes converges to a local minimum. Further, the solution is more robust in the presence of noise in the data; that is, it changes continuously with noise. This is rather promising since, in many applications, the surface of the scene can be approximated as planar either globally or locally. One potential exception is in a near-ambiguous situation (for quadratic surfaces) where up to three solutions may exist. For example, suppose that we have obtained two possible solutions from the planar algorithm and three possible solutions from the quadratic algorithm. There is no guarantee that the two planar solutions correspond to any two of the three possible quadratic solutions nor is there any guarantee that one of the two solutions is the one we seek. Thus, even though we may implement the method for planes, it may be necessary to require some kind of confirmation, for example, by checking the solution of the scheme for quadratic surfaces.

One alternative that should be avoided, if possible, is to reduce the size of the image region. In most cases, the whole region of the image plane must be used to obtain robust results; in particular, the components of rotation about the optical axis and translation

in the direction of the optical axis are difficult to obtain if the image region is too small. In several of the examples we presented, the components of rotation and translation in the image plane could not be recovered accurately; however, the error in the estimate was consistent with the ambiguity in distinguishing a rotation about the positive x -axis from a translation in the negative y direction or a rotation about the positive y -axis from a translation in the positive x direction. This is an inherent ambiguity in the motion vision problem when local information (both in space and in time) is used. To resolve this problem, we not only need to consider a large field of view but we should use the data over several consecutive image frames. For example, we can formulate the problem as one of filtering and estimation. The time-varying images can be viewed as observation of a dynamic system with the scene structure and the motion parameters being the state variables. Alternatively, the motion parameters may be thought of as slowly-varying system parameters that are to be identified.

When the motion is restricted to the translation of the viewer, we showed that it is possible to determine the direction of translation from the directions of image brightness gradients. No assumption about the surface of the scene is needed. The constraint that is used here is that any point in the image corresponds to a point in the scene with a positive depth value (the point is in front of the camera). When an incorrect translation is assumed, many image points take on negative depth values. Usually, clusters of negative and positive depth values are formed around the line that connects the focus of expansion corresponding to the incorrect translation to the true focus of expansion. This can be used to locate the focus of expansion and consequently determine the direction of translation. In two examples, we showed that the focus of expansion can be located with good accuracy with as much as 60% noise in the data; however, the best results are usually obtained when the focus of expansion is within the field of view. It is possible to extend this method if the rotation parameters are not known. This involves assuming an arbitrary rotation so that we can discount the effect of rotation before we apply the method proposed for the case of a translating viewer. Ideally, the results are consistent (that is, the FOE can be located) only if we assume the correct rotation. Through a selected example, we showed that the solution may not be unique in a local neighborhood of the solution because of the inherent ambiguities in the presence of noise in the data.

An important question in motion vision is what circumstances can give rise to an ambiguity in the interpretation of 3-D motion aside from the scale-factor ambiguity. Even though some results have been given in the literature, these have been restricted to special cases such as the case of planes or quadratic surfaces. We have shown that motion vision problem can have at most three solutions. Only planar surfaces and certain special quadratic surfaces, namely those with a negative or zero Gaussian curvature, can give rise to an ambiguity under very special circumstances. We have also derived the relationship among the multiple solutions (when they exist). These results are significant not only for motion vision applications but in other applications where the relative position and

orientation of two cameras have to be determined from the images taken from the two stations, for example, as in photogrammetry, stereo, or camera calibration. In each of these problems, we may need to determine the relative position and orientation of two camera stations from images of certain feature points. Under rare occurrences, if the points happen to lie on an "ambiguous surface" and the unknown displacement and rotation that align the two cameras happen to be consistent with the corresponding "ambiguous motion", we may obtain only one of up to three possible solutions from an iterative method. We can then determine the other solutions, when they exist, from analytical expressions. Actually, the motion vision problem is slightly different since we deal with an infinitesimal base-line in comparison to finite base-line in other applications; however, the ambiguities are similar in either case. When an ambiguity in the solution exists, it is usually possible to identify the correct solution by imposing the constraint that the correct solution should give rise to positive depth values over the relevant region of the image.

One important issue that we have not addressed in this thesis revolves around the issue of scale. We need to filter and sample the image appropriately in order to obtain accurate estimates of the motion parameters from the information in the brightness derivatives. In order to estimate the spatial derivatives accurately, the image brightness should vary smoothly over the image or within the occluding boundaries. In most cases, the image should be low-pass filtered before we estimate the spatial derivatives; however, excessive filtering can leave us with too little information to work with. It is also very important to filter the image before it is sampled. One way to do this is to average several consecutive frames before we sample the image. To estimate the temporal derivatives accurately we often need to assume small relative motion between image frames. This imposes a constraint on the required sampling rate. At the same time, the Nyquist conditions of sampling theory should be satisfied. To summarize, we need to know the optimum sampling rate in recording images to obtain a good estimate of the temporal derivatives and the optimum filtering rate for a good estimate of spatial derivatives.

Image segmentation is another important issue. In many situations, the scene consists of both stationary and moving objects. Before we can make any attempt to recover motion, it is necessary to segment the image into regions of "similar" velocity, that is, regions that include objects with identical velocities; this is not an easy task when our primary goal is to recover the motion. Ideally, we would like to build the segmentation process into the system for recovering motion. This may require a procedure similar to the one adopted by Adiv [1984] using optical flow data and a measure of confidence at each image point to perform image segmentation and motion estimation simultaneously. Here, we use the brightness gradients instead. Initially, we can segment the image into several regions. For each region, we estimate the motion parameters, for example, by assuming that the surface is planar in that region. For each region, we test if the estimated motion parameters are consistent with the data. Based on this, we either divide the

segments into smaller ones or patch together “similar” neighboring segments (segments with similar velocities). This process is repeated until different segments are identified and their velocities are estimated. The initial segmentation can be done coarsely, for example, based on setting simple grey-level thresholds. Alternatively, we can build the segmentation process into the method given for locating the focus of expansion using brightness gradient vectors. In this case, regions with similar velocities give rise to the same focus of expansion. When the image includes several objects with different velocities, there would be several foci of expansion which can be located from the intersection points of the FOE constraint lines. In this case, we obtain several distinct intersection points corresponding to the different loci of expansion.

In conclusion, it is our belief that much can be learned about three-dimensional motion simply by studying the variations in image brightness patterns. With this point of view, this thesis should serve as a first step in that direction. It is conceivable that we need more information than that provided by the time-varying image, especially since local information, both spatial and temporal, can be highly ambiguous. This is not surprising since most sophisticated biological vision systems do not usually perform adequately with only local information. The additional information can be incorporated in the form of constraints on the nature of the solution. In some cases, simple assumptions are sufficient to obtain an accurate estimate of three-dimensional motion. In other situations, we may need more than a few mathematical simplifications.

Appendix

A: Non-Robustness of Certain Analytical Methods

A common approach to obtaining a closed-form solution to a set of non-linear equations in terms of some unknown parameters is (1) to identify a set of intermediate parameters that are some functions (usually non-linear) of the original unknown parameters, (2) to estimate these parameters from the available data and, finally, (3) to estimate the original unknown parameters from the intermediate parameters in closed form. The justification for doing this is that it is sometimes possible to determine the original parameters from the intermediate parameters analytically, whereas the original parameters cannot be determined directly from the data in closed form. The intermediate parameters are usually defined in terms of the original unknowns in a way that makes it easy to compute them from the available data; for example, they are defined so that the constraint equation(s) among the data and the intermediate parameters become(s) linear in the intermediate parameters. The essential parameters we defined for planar surfaces (in Chapter 4) or higher-order surfaces (in Chapter 6) are typical intermediate parameters. In most cases, the number of intermediate parameters exceeds the number of original unknowns. To compute the original unknowns from the intermediate parameters, then either some of the constraints are selectively ignored or they are used to resolve any ambiguity in the solution of the original unknown parameters, if it exists. Such methods are sometimes labeled incorrectly as least-squares methods simply because a least-squares formulation is used to estimate the intermediate parameters.

Examples of methods based on this approach are Longuet-Higgins & Prazdny [1980], Longuet-Higgins [1981], Waxman & Ullman [1983], Tsai & Huang [1984], Negahdaripour

& Horn [1986] (which is given in Chapter 4), and the formulation given in Chapter 6 to derive the uniqueness results. We will now explain why the methods based on this approach are generally non-robust and their solution is non-unique. Therefore, they are not appropriate for implementation unless certain constraints are explicitly enforced before the intermediate parameters are computed (as in Negahdaripour & Horn [1986]). We use the method given in Chapter 6 to demonstrate the types of problems that we may encounter when such methods are used. We also explain why the method given in Chapter 4 is a correct implementation in the least-squares sense; that is, the estimate obtained from the closed-form solution is the same as the one obtained from a least-squares formulation in terms of the original parameters which requires solving some non-linear equations iteratively.

In Chapter 6, we presented some results on the unique interpretation of the image motion field arising from the motion of an observer with respect to its environment. In doing so, 12 essential parameters that are the coefficients of the motion field equations up to second order were used to determine the motion parameters and the surface parameters up to the second-order terms. The remaining essential parameters were then used to determine the higher-order terms of the surface and to resolve the ambiguities in the solution of the motion parameters. The essential parameters are computed from the image data through a least-squares formulation where the sum-squared error in the brightness change equation,

$$e^2 = \iint_I (c + \mathbf{e} \cdot \mathbf{m})^2 dx dy,$$

is minimized (c is the brightness temporal derivative, \mathbf{e} is some vector of moments of the brightness gradient and \mathbf{m} is the vector of the essential parameters). The solution is obtained from

$$\mathbf{m} = - \left(\iint_I \mathbf{e} \mathbf{e}^T dx dy \right)^{-1} \left(\iint_I c \mathbf{e} dx dy \right).$$

The above equation shows that the solution of \mathbf{m} cannot be determined uniquely if the inverse matrix does not exist. In general, the solution is non-robust even when the matrix is invertible for the following reasons. For a small field of view, we have $l = |[x, y]^T| < 1$ for the majority of the data points. Therefore, the higher-order terms of \mathbf{e} , which depend on the higher powers of the image point coordinates, will tend to zero. Consequently, we cannot accurately estimate those essential parameters that are the coefficients of the higher-order terms of the flow field; these are functions of the higher order terms of the surface. Therefore, we are forced to consider a large portion of the image. When the field of view is large, the solution is sensitive to error in the estimate of the brightness gradient since the elements of vector \mathbf{e} that are the higher-order moments of the brightness derivative will be inaccurate (these are the brightness gradient scaled by a term of order l^n where n is the order of the surface). In general, the estimates of the intermediate parameters are not accurate due to noise in the data and quantization error.

As mentioned earlier, the number of the essential parameters is generally greater than the number of independent motion and surface parameters; there are $6 + n(n + 3)/2$ unknown motion and surface parameters, however, the number of essential parameters is $(n + 1)(n + 3)$. When the surface is planar ($n = 1$), we have 8 essential parameters and 8 independent unknowns. For $n > 1$, we always have $(n + 1)(n + 3) > 6 + n(n + 3)/2$. In other words, there are only $6 + n(n + 3)/2$ degrees of freedom in the $(n + 1)(n + 3)$ essential parameters for non-planar surfaces. With perfect data, the least-squares formulation gives the correct value for the essential parameters. Therefore, these will obey the constraints imposed by the $(n + 1)(n + 3)$ equations. When the data are noisy, the essential parameters are generally independent with $(n + 1)(n + 3)$ degrees of freedom since the least-squares procedure gives the optimum parameters that minimize the error function without enforcing the constraint(s) among these parameters. Consequently, the unknown motion and surface parameters will not satisfy the constraint equations imposed by the essential parameters. Furthermore, the solution will depend on the particular method used to compute the motion and surface parameters. This point is best demonstrated by the simple yet very informative example taken from Scott [1986].

Suppose we are given n data points that are expected to lie on a pair of straight lines in the image—we are to determine the equations of the two lines. Writing the equations of the two lines as

$$ax + by + 1 = 0 \quad \text{and} \quad a'x + b'y + 1 = 0,$$

the goal is to obtain an optimum estimate of the four independent parameters a , b , a' , and b' . While we can consider several iterative procedures we are interested in a closed-form solution.

With perfect data, the points lie on at least one of the two lines and their coordinates satisfy the equation

$$(ax + by + 1)(a'x + b'y + 1) = 0.$$

After multiplying out and simplifying we arrive at

$$1 + (a + a')x + (b + b')y + (aa')x^2 + (ab' + ba')xy + (bb')y^2 = 0,$$

which can be written

$$1 + \mathbf{e} \cdot \mathbf{m} = 0,$$

where

$$\mathbf{e} = (x, y, x^2, xy, y^2)^T \quad \text{and} \quad \mathbf{m} = ((a + a'), (b + b'), aa', (ab' + ba'), bb')^T.$$

The above equation has the same form as the brightness change constraint equation. The parameters we are after, the elements of \mathbf{m} , correspond to the essential parameters of an n^{th} order surface (in this case, we have five of them) and \mathbf{e} plays the same role as the vector of moments of image brightness gradient. We can estimate \mathbf{m} by employing

a least-squares formulation like the one used to estimate the essential parameters. The least-squares solution for \mathbf{m} is given by

$$\mathbf{m} = -\left(\sum_{i=1}^n \mathbf{e}\mathbf{e}^T\right)^{-1} \left(\sum_{i=1}^n \mathbf{e}\right).$$

We can then proceed to compute a and a' from

$$m_1 = a + a' \quad \text{and} \quad m_3 = aa',$$

and b and b' from

$$m_2 = b + b' \quad \text{and} \quad m_5 = bb'.$$

This requires solving two quadratic equations; therefore, two solutions are possible for each of a , a' , b , and b' . Finally, we can use the equation

$$m_4 = ab' + ba'$$

to select the correct solution; that is, we use it to pair up a with b , and a' with b' .

Now, let us consider the problems we face when the data are noisy. In general, the estimate of the five "essential parameters" (the elements of the vector \mathbf{m}) will not be consistent with them being composed of four independent parameters (four degrees of freedom) in the right combinations. In fact, these parameters will be the coefficients of a conic section (see the quadratic constraint equation). Therefore, we may find a circle, an ellipse, a parabola, or a hyperbola instead of two lines. We might be more unfortunate (or fortunate depending on the point of view) if the two quadratic equations we have to solve to obtain a , a' , b , and b' have no real solutions. Furthermore, we obtain a different solution (when the solution exists) if we choose another set of four equations (out of the five equations) to estimate the four unknown parameters, a , a' , b , and b' . In order to obtain a unique solution that is optimum in the least-squares sense, we have to incorporate the constraint among the essential parameters in the least-squares formulation used to estimate these parameters.

Consider, now, the closed-form solution given in Chapter 4 for the special case of planar surfaces. In this formulation, the brightness constraint equation is written

$$E_t + \mathbf{r}^T \mathbf{P} \mathbf{s} = 0.$$

As before, \mathbf{r} is a point in the image, E_t is the time derivative of the brightness at point \mathbf{r} , \mathbf{s} is a vector of moments of the spatial derivative of brightness at \mathbf{r} , and \mathbf{P} is a 3×3 matrix whose elements are the nine essential parameters of a planar surface which is defined to be

$$\mathbf{P} = -\Omega + \mathbf{n}\mathbf{t}^T.$$

However, there are only eight degrees of freedom in \mathbf{P} since the six unknowns in \mathbf{n} and \mathbf{t} can be recovered only up to a scale factor. We have $\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = \mathbf{n}\mathbf{t}^T + \mathbf{t}\mathbf{n}^T$ because $\mathbf{\Omega}$ is a skew-symmetric matrix. Since \mathbf{P}^* is the sum of two dyadic products, it has at most rank two. Therefore, the constraint among the essential parameters can be written

$$\det(\mathbf{P}^*) = \det(\mathbf{P} + \mathbf{P}^T) = 0.$$

Without imposing the above constraint, we may attempt to determine the essential parameters by minimizing the integral of the square of the error in the brightness change constraint equation over the whole image; that is, by minimizing

$$\iint_I (E_t + \mathbf{r}^T \mathbf{P} \mathbf{s})^2 dx dy.$$

When noisy data, the solution of the essential parameters will not be consistent with the fact that there are only eight degrees of freedom in these parameters. Furthermore, we now have nine equations in eight unknowns; we obtain nine estimates for each solution (remember the two-fold ambiguity of planar surfaces) depending on which eight equations we use to obtain the solution.

To compute the nine essential parameters consistently with the existence of only eight degrees of freedom, we have to incorporate the constraint $\det(\mathbf{P}^*) = 0$ in the least-squares formulation. For example, by introducing the lagrange multiplier k we can reduce the problem to that of an unconstrained optimization. This involves choosing the elements of \mathbf{P} and k such that we minimize the error function

$$\iint_I (E_t + \mathbf{r}^T \mathbf{P} \mathbf{s})^2 dx dy + k \left| \mathbf{P} + \mathbf{P}^T \right|^2.$$

Unfortunately, it is not easy to determine the essential parameters from this formulation since the error function is now non-quadratic because of the added new term; however, we showed a way around this problem by noting that the middle eigenvalue of \mathbf{P}^* is zero. Using this fact, we showed that we can arbitrarily set one of the diagonal elements of \mathbf{P} equal to zero (we chose the last element). Consequently, we can determine the remaining eight essential parameters using either the brightness change constraint equation at eight image points or from a least-squares formulation. In either case, the number of sought-after essential parameters is the same as the number of degrees of freedom after we impose the appropriate constraint. Therefore, we are guaranteed to obtain only one estimate for the motion parameters since we do not have to selectively discard a constraint after the essential parameters are computed. Furthermore, the estimate is optimum in the least-squares sense.

We are not as lucky in the case of quadratic surfaces or other higher-order surfaces. Consider the coefficients of the flow field up to the second-order terms. In this case, we have 12 essential parameters with only 11 unknown motion and surface parameters.

To compute the essential parameters “consistently”, we need to identify the appropriate constraint among the 12 essential parameters. Unfortunately, this cannot be done as easily as in the case of planar surfaces. Furthermore, there is no guarantee that we can determine the essential parameters easily after we have incorporated this constraint in the formulation. Suppose we compute the 12 essential parameters without enforcing the constraint among them. Then the 12 essential parameters will not satisfy the constraint that they are composed of only 11 independent parameters in the right combinations. With noisy data, the solution of the motion and surface parameters can change drastically depending on which 11 essential parameters are used. For example, the method presented to show uniqueness results can give a solution very different from the one obtained if we choose any other set of 11 equations. In general, 12 different solutions may be obtained. The situation is worse if there is a potential two-fold or three-fold ambiguity in the solution. In fact, the method we have proposed is prone to instability since we need to determine the transformation angle from one of the essential parameters that is the coefficient of one of the second-order terms of the flow field. In general, we cannot determine the second-order coefficients accurately (Le Guilloux [1986]). Therefore, we are faced with two problems; the inaccuracy in the estimate of the essential parameters obtained from the least-squares formulation, and additional errors introduced by using the wrong transformation angle to compute the essential parameters in the new coordinate system.

In conclusion, we again emphasize that one should be well aware of the danger of using the kind of analytical methods described here. Since these methods usually do not work well with synthetic data (as shown in several examples in Chapter 6), it is hard to believe that they will ever work well on real images. Unfortunately, these methods still have their appeal for some theoreticians. It seems more reasonable to implement an iterative method regardless of the computational cost than to use an analytical method that, in most cases, will not fail to disappoint us.

B: Surface Properties relevant to Ambiguities in Motion Vision

The procedure for presenting uniqueness results requires the solution of a set of non-linear equations. It is easier to solve these equations in a transformed image coordinate frame. To compute the transformation angle, we need to solve the cubic equation,

$$(t_2 \tan \theta - t_1)(d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22}) = 0,$$

where t_i denotes the component of translation parallel to the image plane and d_{ij} are the coefficients of the quadratic terms of the surface function,

$$d = 1 + d_1 x + d_2 y + \frac{1}{2} d_{11} x^2 + d_{12} xy + \frac{1}{2} d_{22} y^2 + O(e).$$

Only one of the solutions of the cubic equation, $\tan \theta = t_1/t_2$, corresponds to the proper image coordinate transformation. The other two solutions (the so-called spurious solutions), i.e., the solutions of

$$(d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22}) = 0$$

are given by

$$\tan \theta = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}.$$

These solutions, if they exist, are directly related to the local properties of the surface, namely, the Gaussian curvature and the directions of asymptotic lines of the surface at the fixation point O . We are only interested in the real solutions of the above quadratic equation; these may not exist. Here, we will investigate this in more details.

Let \mathbf{H} be the Hessian of the surface at the point of fixation O , that is,

$$\mathbf{H} = \begin{pmatrix} \left(\frac{\partial^2 Z}{\partial X^2}\right)_o & \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_o \\ \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_o & \left(\frac{\partial^2 Z}{\partial Y^2}\right)_o \end{pmatrix}.$$

The directions of maximum and minimum curvature of the surface (at any point) which are orthogonal to each other are referred to as the directions of *principal curvature*. The product of the maximum and minimum curvatures is known as the *Gaussian curvature*, and $\frac{1}{2}$ their sum is known as the *mean curvature*. These are two properties of the surface that are invariant with respect to the orientation of the coordinate system. The sign of the Gaussian curvature of the surface at O is that of $\det(\mathbf{H})$, where

$$\det(\mathbf{H}) = \left(\frac{\partial^2 Z}{\partial X^2}\right)_o \left(\frac{\partial^2 Z}{\partial Y^2}\right)_o - \left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_o^2.$$

Depending on the sign of the Gaussian curvature, the fixation point is labeled as either an *elliptic paraboloid*, a *parabolic cylinder*, or a *hyperbolic paraboloid (saddle)* point (Spivak [1975]).

In Appendix C, we have shown that

$$d_o = \frac{1}{Z_o}, \quad d_1 = -\frac{1}{Z_o} \left(\frac{\partial Z}{\partial X}\right)_o, \quad d_2 = -\frac{1}{Z_o} \left(\frac{\partial Z}{\partial Y}\right)_o,$$

$$d_{11} = -\left(\frac{\partial^2 Z}{\partial X^2}\right)_o, \quad d_{12} = -\left(\frac{\partial^2 Z}{\partial X \partial Y}\right)_o, \quad \text{and} \quad d_{22} = -\left(\frac{\partial^2 Z}{\partial Y^2}\right)_o.$$

Therefore, $\det(\mathbf{H})$ is also given by

$$\det(\mathbf{H}) = d_{11}d_{22} - d_{12}^2.$$

When $\det(\mathbf{H}) < 0$, that is, the surface has a negative Gaussian curvature at O , the point is labeled as a hyperbolic paraboloid (or saddle) point. Figure B-1a shows an example of a surface with a negative Gaussian curvature at the origin. In this case, the principal curvatures have opposite signs. Since we have

$$\det(\mathbf{H}) = d_{11}d_{22} - d_{12}^2 < 0,$$

the two solutions of

$$d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22} = 0$$

given by

$$\tan \theta = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}$$

are real. The two directions given by the transformation angle θ are those of the asymptotic lines of the surface. Surfaces with a negative Gaussian curvature possess two asymptotic lines that are symmetric with respect to the two directions of principal curvature.

When $\det(\mathbf{H}) = 0$ but $\mathbf{H} \neq \mathbf{0}$ (that is, \mathbf{H} is singular) the Gaussian curvature at O is zero. The point O is labeled as a parabolic cylinder (see Figure B-1b). In this case, one of the principal curvatures is zero. Since we have

$$\det(\mathbf{H}) = d_{12}^2 - d_{11}d_{22} = 0,$$

the two solutions of

$$d_{11} \tan^2 \theta + 2d_{12} \tan \theta + d_{22} = 0,$$

given by

$$\tan \theta = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}}$$

degenerate into one solution, $\tan \theta = -d_{12}/d_{11} = -d_{22}/d_{12}$. The direction given by θ corresponds to the only asymptotic direction of the surface. In this case, the only asymptotic direction coincides with the direction of zero curvature of the surface. In the special case when $\mathbf{H} = \mathbf{0}$ at the origin, both principal curvatures are zero and the point O degenerates into a planar point (see Figure B-1c).

When $\det(\mathbf{H}) > 0$, that is, the surface has a positive Gaussian curvature at O , the point is labeled as an elliptic paraboloid point. Figure B-1d shows examples of surfaces with a positive Gaussian curvature at the origin. In this case, both principal curvatures are either positive or negative and the solutions given by

$$\tan \theta = \frac{-d_{12} \pm \sqrt{d_{12}^2 - d_{11}d_{22}}}{d_{11}},$$

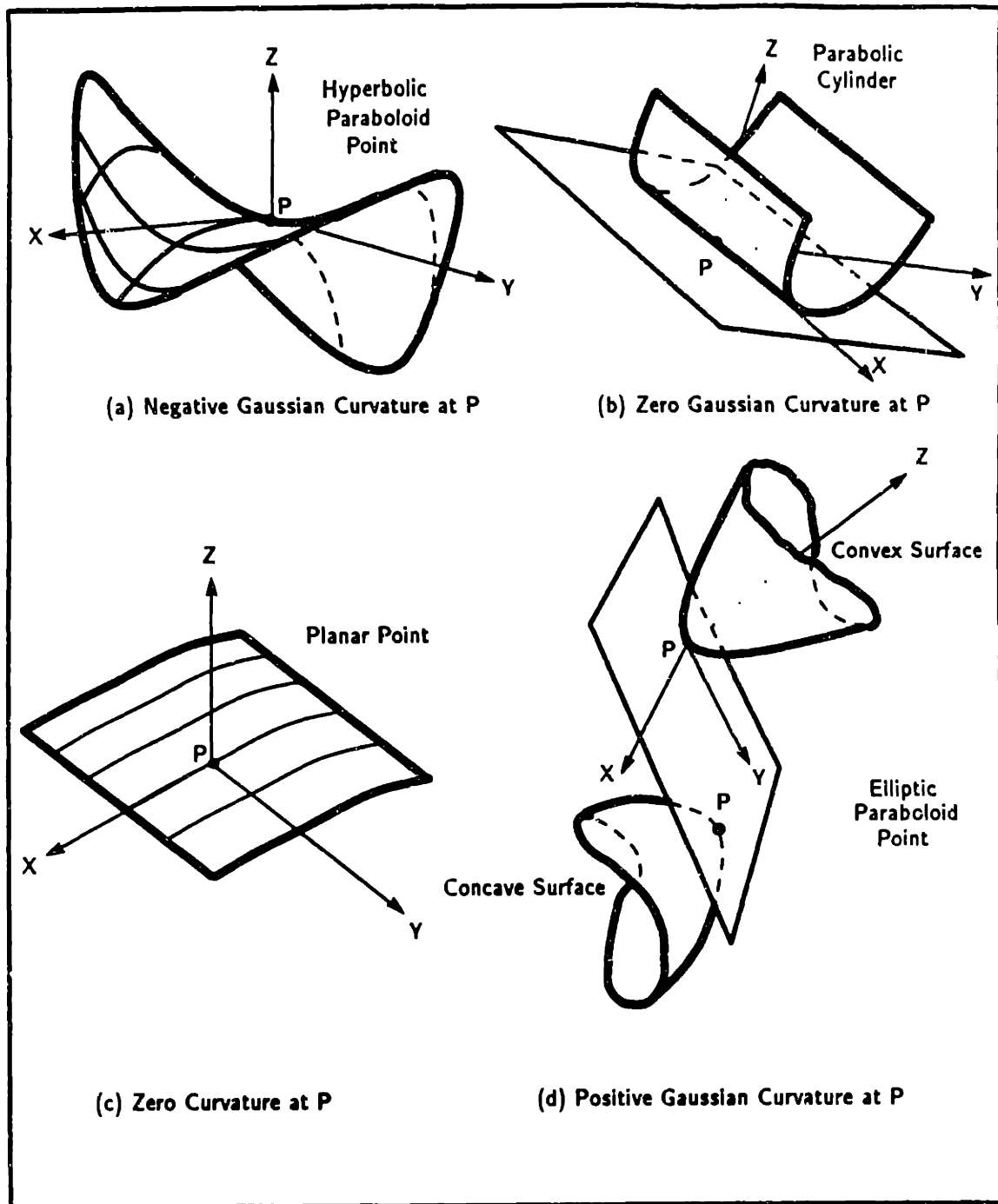


Figure B-1. Surfaces with negative, zero, and positive Gaussian curvature.

become complex. Further, the surface does not possess any asymptotic lines.

Figure B-2 shows a surface that has points with positive, zero, and negative Gaussian curvature.

ZZ: A Point with Zero Curvature
 ZP: A Point with Zero Gaussian Curvature and Positive Mean Curvature
 ZN: A Point with Zero Gaussian Curvature and Negative Mean Curvature
 PN: Point with Positive Gaussian Curvature and Negative Mean Curvature
 N: A Point with Negative Gaussian Curvature

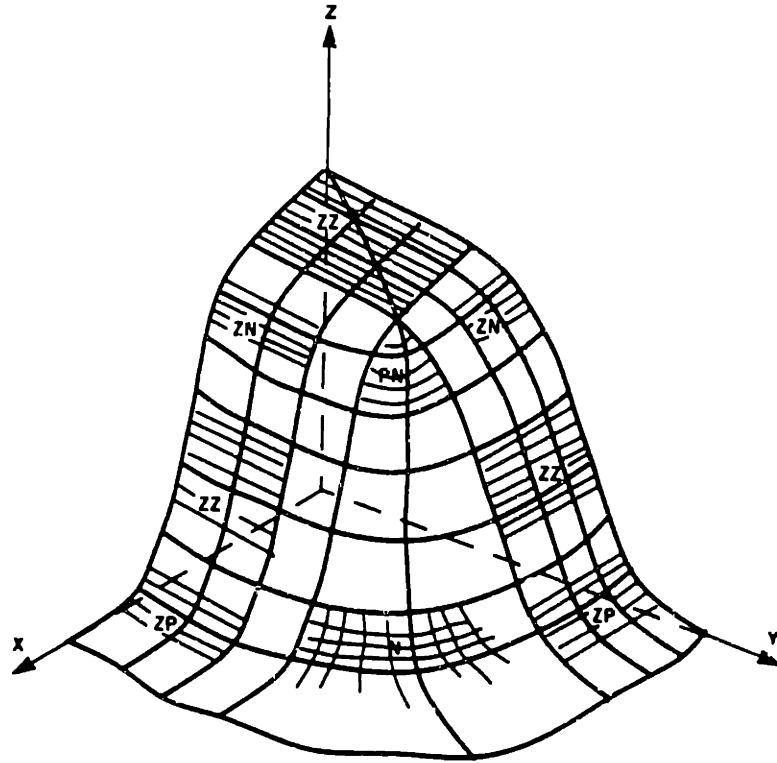


Figure B-2. A surface consisting of points with negative, zero, and positive Gaussian curvature.

C: Relating Image and Surface Taylor Coefficients

We defined $d = 1/Z$. Therefore

$$\frac{\partial d}{\partial x_i} = -\frac{1}{Z^2} \frac{\partial Z}{\partial x_i}, \quad \frac{\partial^2 d}{\partial x_i \partial x_j} = -\frac{1}{Z^2} \frac{\partial^2 Z}{\partial x_i \partial x_j} + \frac{2}{Z^3} \left(\frac{\partial Z}{\partial x_i} \right) \left(\frac{\partial Z}{\partial x_j} \right).$$

Evaluating these expressions at the origin, we obtain

$$\left(\frac{\partial d}{\partial x_i} \right)_0 = -\frac{1}{Z_0^2} \left(\frac{\partial Z}{\partial x_i} \right)_0, \quad \left(\frac{\partial^2 d}{\partial x_i \partial x_j} \right)_0 = -\frac{1}{Z_0^2} \left(\frac{\partial^2 Z}{\partial x_i \partial x_j} \right)_0 + \frac{2}{Z_0^3} \left(\frac{\partial Z}{\partial x_i} \right)_0 \left(\frac{\partial Z}{\partial x_j} \right)_0.$$

We now use the perspective projection equation $X_i = x_i Z$ to arrive at

$$\frac{\partial Z}{\partial x_i} = \frac{\partial Z}{\partial X_m} \frac{\partial X_m}{\partial x_i} = \frac{\partial Z}{\partial X_m} (\delta_{im} Z + x_m \frac{\partial Z}{\partial x_i}).$$

Therefore, we have

$$\frac{\partial Z}{\partial x_i} = Z \frac{\partial Z}{\partial X_i} \left(1 - x_m \frac{\partial Z}{\partial X_m} \right)^{-1}.$$

Now,

$$\frac{\partial^2 Z}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_j} \left(\frac{\partial Z}{\partial X_m} \right) (\delta_{im} Z + x_m \frac{\partial Z}{\partial x_i}) + \frac{\partial Z}{\partial X_m} \left(\delta_{im} \frac{\partial Z}{\partial x_j} + \delta_{jm} \frac{\partial Z}{\partial x_i} + x_m \frac{\partial}{\partial x_j} \left(\frac{\partial Z}{\partial x_i} \right) \right),$$

or

$$\frac{\partial^2 Z}{\partial x_i \partial x_j} = \frac{\partial}{\partial X_n} \left(\frac{\partial Z}{\partial X_m} \right) \frac{\partial X_n}{\partial x_j} (\delta_{im} Z + x_m \frac{\partial Z}{\partial x_i}) + \frac{\partial Z}{\partial X_m} (\delta_{im} \frac{\partial Z}{\partial x_j} + \delta_{jm} \frac{\partial Z}{\partial x_i} + x_m \frac{\partial^2 Z}{\partial x_i \partial x_j}).$$

This simplifies to

$$\frac{\partial^2 Z}{\partial x_i \partial x_j} = \frac{\partial^2 Z}{\partial X_m \partial X_n} (\delta_{im} Z + x_m \frac{\partial Z}{\partial x_i}) (\delta_{jn} Z + x_n \frac{\partial Z}{\partial x_j}) + \frac{\partial Z}{\partial X_m} (\delta_{im} \frac{\partial Z}{\partial x_j} + \delta_{jm} \frac{\partial Z}{\partial x_i} + x_m \frac{\partial^2 Z}{\partial x_i \partial x_j}).$$

After substituting for $\partial Z / \partial x_i$, we arrive at

$$\frac{\partial^2 Z}{\partial x_i \partial x_j} = \frac{\partial^2 Z}{\partial X_m \partial X_n} \tilde{Z}_{im} \tilde{Z}_{jn} + \frac{\partial Z}{\partial X_m} + \left[Z \left(1 - x_k \frac{\partial Z}{\partial X_k} \right)^{-1} (\delta_{im} \frac{\partial Z}{\partial X_j} + \delta_{jm} \frac{\partial Z}{\partial X_i}) + x_m \frac{\partial^2 Z}{\partial x_i \partial x_j} \right],$$

where

$$\tilde{Z}_{im} = \delta_{im} Z + x_m Z \frac{\partial Z}{\partial X_i} \left(1 - x_k \frac{\partial Z}{\partial X_k} \right)^{-1}.$$

Solving for $\partial^2 Z / \partial x_i \partial x_j$, we obtain

$$\frac{\partial^2 Z}{\partial x_i \partial x_j} = \left(1 - x_r \frac{\partial Z}{\partial X_r} \right)^{-1} \tilde{Z}_{im} \tilde{Z}_{jn} \frac{\partial^2 Z}{\partial X_m \partial X_n} + Z \left(1 - x_k \frac{\partial Z}{\partial X_k} \right)^{-2} (\delta_{im} \frac{\partial Z}{\partial X_j} + \delta_{jm} \frac{\partial Z}{\partial X_i}) \frac{\partial Z}{\partial X_m}.$$

Evaluating $(\partial Z / \partial x_i)_0$ and $(\partial^2 Z / \partial x_i \partial x_j)_0$ at the origin results in

$$\left(\frac{\partial Z}{\partial x_i} \right)_0 = Z_0 \frac{\partial Z}{\partial X_i}, \quad \left(\frac{\partial^2 Z}{\partial x_i \partial x_j} \right)_0 = Z_0^2 \left(\frac{\partial^2 Z}{\partial X_i \partial X_j} \right)_0 + 2Z_0 \left(\frac{\partial Z}{\partial X_i} \right)_0 \left(\frac{\partial Z}{\partial X_j} \right)_0.$$

Finally, substituting for $(\partial Z / \partial x_i)_0$ and $(\partial^2 Z / \partial x_i \partial x_j)_0$ into the equations for $(\partial d / \partial x_i)_0$ and $(\partial^2 d / \partial x_i \partial x_j)_0$ we arrive at

$$\left(\frac{\partial d}{\partial x_i} \right)_0 = -\frac{1}{Z_0} \left(\frac{\partial Z}{\partial X_i} \right)_0, \quad \left(\frac{\partial^2 d}{\partial x_i \partial x_j} \right)_0 = -\left(\frac{\partial^2 Z}{\partial X_i \partial X_j} \right)_0.$$

D: Distance from a Point to a Conic Section

We want to determine the shortest distance from a given point, (x_0, y_0) , to the conic section,

$$d_0 + d_1 x + d_2 y + \frac{1}{2} d_{11} x^2 + d_{12} xy + \frac{1}{2} d_{22} y^2 = 0.$$

The solution to this problem is probably given in several references (for example, an astronomy textbook); however, the solution can be derived as easily as it can be looked up.

For simplicity, let us assume that the point is the origin, that is, $(x_0, y_0) = 0$. There is no loss of generality since we can always move the origin to the given point to accommodate this. This introduces a simple transformation on the coefficients d_0, d_1, \dots, d_{22} . Now the problem can be formulated as a constrained optimization problem:

Minimize $l^2 = x^2 + y^2$ subject to the constraint

$$d_0 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2 = 0.$$

We can make this an unconstrained optimization problem by introducing a Lagrange multiplier. Thus, we solve the following problem:

Determine (x, y) and λ that minimize

$$L = \frac{1}{2}(x^2 + y^2) + \lambda(d_0 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2).$$

Necessary conditions for optimality are

$$\frac{\partial L}{\partial x} = 0, \quad \frac{\partial L}{\partial y} = 0, \quad \text{and} \quad \frac{\partial L}{\partial \lambda} = 0.$$

Performing the above differentiation we obtain

$$\begin{aligned} x + \lambda(d_1 + d_{11}x + d_{12}y) &= 0, \\ y + \lambda(d_2 + d_{12}x + d_{22}y) &= 0, \\ d_0 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2 &= 0. \end{aligned}$$

We can eliminate λ from the first two equations. Then, the solution is obtained by solving simultaneously the two quadratic equations

$$\begin{aligned} d_2x - d_1y + d_{12}x^2 + (d_{22} - d_{11})xy - d_{12}y^2 &= 0, \\ d_0 + d_1x + d_2y + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2 &= 0. \end{aligned}$$

These can be reduced to a fourth-order polynomial (quartic equation) that has at most two real solutions because of the geometry of the problem (analytical expressions for the solutions of a quartic can be found in any mathematics handbook, for example, Korn & Korn [1968]). When the conic section is either a circle or an ellipse, the two solutions

correspond to the two points closest to and furthest away from the origin. For parabolas, there is only one solution corresponding to the closest point to the origin. Finally, for hyperbolas, there are two solutions, one global minimum and one local minimum, corresponding to the two points closest to the origin each located on one branch of the hyperbola (the local minimum may be equal to the global one if the origin lies on the line of symmetry which is the locus of points that are equidistant from the two branches).

Special Case: Let us consider the special case corresponding to the three-fold ambiguity of quadratic surfaces. In this case, we have

- (1) $d_0 = 1, d_1 = d_2 = 0,$
- (2) $d_{12}^2 - d_{11}d_{22} > 0,$ and
- (3) $d_{11} + d_{22} = 2.$

Then, the quadratic equations reduce to

$$\begin{aligned} d_{12}x^2 + (d_{22} - d_{11})xy - d_{12}y^2 &= 0, \\ 1 + \frac{1}{2}d_{11}x^2 + d_{12}xy + \frac{1}{2}d_{22}y^2 &= 0. \end{aligned}$$

If $d_{12} \neq 0$, we can always write $y = kx$ for some constant k . Substituting into the first equation we obtain a quadratic equation in k ,

$$d_{12}k^2 + (d_{22} - d_{11})k - d_{12} = 0.$$

The solutions are given by

$$k = \left(\frac{d_{22} - d_{11}}{2d_{12}} \right) \pm \sqrt{\left(\frac{d_{22} - d_{11}}{2d_{12}} \right)^2 + 1}.$$

Finally, substituting $y = kx$ into the second equation we obtain

$$(x = \pm \sqrt{\frac{-2}{d_{22}k^2 + 2d_{12}k + d_{11}}}, y = kx),$$

where k is given by the earlier equation. Two solutions are obtained for the same k that are symmetric with respect to the origin (the other solution for k gives two complex solutions for x).

When $d_{12} = 0$, we have

$$d_{12}^2 - d_{11}d_{22} = -d_{11}d_{22} > 0 \quad \text{or} \quad d_{11}d_{22} < 0.$$

Since $(d_{11} + d_{22}) = 2$, we must have

$$\text{either } d_{11} > 2 \text{ and } d_{22} < 0 \quad \text{or} \quad d_{11} < 0 \text{ and } d_{22} > 0.$$

The two quadratic equations reduce to

$$(d_{22} - d_{11})xy = 0,$$

$$1 + \frac{1}{2}d_{11}x^2 + \frac{1}{2}d_{22}y^2 = 0.$$

The solution is either

$$(x = 0, y = \pm\sqrt{-\frac{2}{d_{22}}}) \quad \text{if } d_{22} < 0$$

or

$$(y = 0, x = \pm\sqrt{-\frac{2}{d_{11}}}) \quad \text{if } d_{11} < 0.$$

Note that only one of the two conditions holds because of the constraints on d_{11} and d_{22} .

E: Point with Least Sum-Squared Distance From n Lines

Consider n lines where the i^{th} line is defined by a point $\tilde{r}_i = (x_i, y_i)$ on the line and its slope m_i . Then the equation of the line is written as

$$m_i(x - x_i) + (y - y_i) = 0.$$

We seek a point (x_o, y_o) with the least sum-squared distance from these lines. The distance of a point (x_o, y_o) from a line

$$m_i(x - x_i) + (y - y_i) = 0$$

is given by

$$l_i = \left| \frac{m_i(x_o - x_i) + (y_o - y_i)}{\sqrt{1 + m_i^2}} \right|.$$

Therefore, the point with the least sum-squared distance is the one that minimizes

$$J = \sum_{i=1}^n l_i^2 = \sum_{i=1}^n \frac{(m_i(x_o - x_i) + (y_o - y_i))^2}{1 + m_i^2}.$$

The solution is obtained from the linear equations,

$$\begin{pmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{12} & \alpha_{22} \end{pmatrix} \begin{pmatrix} x_o \\ y_o \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix},$$

where

$$\alpha_{11} = \sum_{i=1}^n \frac{m_i^2}{k_i}, \quad \alpha_{12} = \sum_{i=1}^n \frac{m_i}{k_i}, \quad \alpha_{22} = \sum_{i=1}^n \frac{1}{k_i},$$

$$\beta_1 = \sum_{i=1}^n \frac{m_i^2 x_i + m_i y_i}{k_i}, \quad \beta_2 = \sum_{i=1}^n \frac{m_i x_i + m_i^2 y_i}{k_i},$$

and

$$k_i = 1 + m_i^2.$$

That is,

$$x_o = \frac{\beta_1 \alpha_{22} - \beta_2 \alpha_{12}}{\alpha_{11} \alpha_{22} - \alpha_{12}^2} \quad \text{and} \quad y_o = \frac{\beta_1 \alpha_{12} - \beta_2 \alpha_{11}}{\alpha_{11} \alpha_{22} - \alpha_{12}^2}.$$

We obtain simpler expressions for the coefficients α_{ij} and β_i ($i, j = 1, 2$) if l_i^2 is scaled by k_i in the expression given for J .

F: Some Properties of Dyadic Products

The dyadic product, $(\mathbf{a}\mathbf{b}^T)$, of the vectors \mathbf{a} and \mathbf{b} can be thought of as the product of a matrix whose single row consists of the components of the vector \mathbf{a} and a matrix whose single column consists of the components of the vector \mathbf{b} . Alternatively, it can be defined by the identities

$$(\mathbf{a}\mathbf{b}^T)\mathbf{c} = \mathbf{a}(\mathbf{b}^T\mathbf{c}) = \mathbf{a}(\mathbf{b} \cdot \mathbf{c}) \quad \text{and} \quad \mathbf{c}^T(\mathbf{a}\mathbf{b}^T) = (\mathbf{c}^T\mathbf{a})\mathbf{b} = (\mathbf{c} \cdot \mathbf{a})\mathbf{b}.$$

Lemma 1: $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)\mathbf{c} = \mathbf{c} \times (\mathbf{a} \times \mathbf{b})$.

Proof: $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)\mathbf{c} = \mathbf{a}(\mathbf{b} \cdot \mathbf{c}) - \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) = \mathbf{c} \times (\mathbf{a} \times \mathbf{b})$. ■

Lemma 2: If $\mathbf{a}\mathbf{b}^T = \mathbf{c}\mathbf{d}^T$, then either

- (1) $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, and $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, or
- (2) $\mathbf{a} \parallel \mathbf{c}$ and $\mathbf{b} \parallel \mathbf{d}$.

Proof: $\mathbf{c}^T(\mathbf{a}\mathbf{b}^T)\mathbf{d} = \mathbf{c}^T(\mathbf{c}\mathbf{d}^T)\mathbf{d}$, or $(\mathbf{c} \cdot \mathbf{a})(\mathbf{b} \cdot \mathbf{d}) = (\mathbf{c} \cdot \mathbf{c})(\mathbf{d} \cdot \mathbf{d})$. Now if $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, then $|\mathbf{c}|^2 |\mathbf{d}|^2 = 0$, which implies that either $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$. Conversely, if $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, then $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$.

For the rest of the proof we assume that $|\mathbf{a}|$, $|\mathbf{b}|$, $|\mathbf{c}|$, and $|\mathbf{d}|$ are non-zero. Now $(\mathbf{a}\mathbf{b}^T)\mathbf{b} = (\mathbf{c}\mathbf{d}^T)\mathbf{b}$ so that $\mathbf{a}(\mathbf{b} \cdot \mathbf{b}) = \mathbf{c}(\mathbf{d} \cdot \mathbf{b})$. It follows that $\mathbf{a} \parallel \mathbf{c}$. Similarly, $\mathbf{a}^T(\mathbf{a}\mathbf{b}^T) = \mathbf{a}^T(\mathbf{c}\mathbf{d}^T)$ so that $(\mathbf{a} \cdot \mathbf{a})\mathbf{b}^T = (\mathbf{a} \cdot \mathbf{c})\mathbf{d}^T$. Therefore, $\mathbf{b} \parallel \mathbf{d}$. ■

Lemma 3: If \mathbf{M} and \mathbf{N} are (3×3) skew-symmetric matrices and $\mathbf{M}^2 = \mathbf{N}^2$, then $\mathbf{M} = \pm\mathbf{N}$.

Proof: The result is immediate if \mathbf{M} and \mathbf{N} are zero. So from now on we assume that they are non-zero. If $\mathbf{M}^2 = \mathbf{N}^2$, then $\mathbf{M}^2\mathbf{x} = \mathbf{N}^2\mathbf{x}$ for all vectors \mathbf{x} . Using the isomorphism between (3×3) skew-symmetrical matrices and vectors we have

$$\mathbf{m} \times (\mathbf{m} \times \mathbf{x}) = \mathbf{n} \times (\mathbf{n} \times \mathbf{x}),$$

where the vectors \mathbf{m} and \mathbf{n} correspond to the matrices \mathbf{M} and \mathbf{N} respectively. So

$$(\mathbf{m} \cdot \mathbf{x})\mathbf{m} - (\mathbf{m} \cdot \mathbf{m})\mathbf{x} = (\mathbf{n} \cdot \mathbf{x})\mathbf{n} - (\mathbf{n} \cdot \mathbf{n})\mathbf{x},$$

or

$$[\mathbf{m}\mathbf{m}^T - (\mathbf{m} \cdot \mathbf{m})\mathbf{I}]\mathbf{x} = [\mathbf{n}\mathbf{n}^T - (\mathbf{n} \cdot \mathbf{n})\mathbf{I}]\mathbf{x}.$$

If this is to be true for all vectors \mathbf{x} we must have $[\mathbf{m}\mathbf{m}^T - (\mathbf{m} \cdot \mathbf{m})\mathbf{I}] = [\mathbf{n}\mathbf{n}^T - (\mathbf{n} \cdot \mathbf{n})\mathbf{I}]$. Taking the trace of both sides we obtain $\mathbf{m} \cdot \mathbf{m} = \mathbf{n} \cdot \mathbf{n}$. Consequently, $\mathbf{m}\mathbf{m}^T = \mathbf{n}\mathbf{n}^T$. Using Lemma 2, we see that $\mathbf{m} \parallel \mathbf{n}$. Taken together with $|\mathbf{m}| = |\mathbf{n}|$ this means that $\mathbf{m} = \pm \mathbf{n}$ and so $\mathbf{M} = \pm \mathbf{N}$. ■

Lemma 4: If $\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T = \mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T$, then either

- (1) $|\mathbf{a}| = 0$ or $|\mathbf{b}| = 0$, and $|\mathbf{c}| = 0$ or $|\mathbf{d}| = 0$, or
- (2) $\mathbf{a} \parallel \mathbf{c}$ and $\mathbf{b} \parallel \mathbf{d}$, or
- (3) $\mathbf{a} \parallel \mathbf{d}$ and $\mathbf{b} \parallel \mathbf{c}$.

Proof: We first show that either $\mathbf{a}\mathbf{b}^T = \mathbf{c}\mathbf{d}^T$ or $\mathbf{a}\mathbf{b}^T = \mathbf{d}\mathbf{c}^T$. The proof then follows using lemma 2. We also need to show that $\mathbf{a} \cdot \mathbf{b} = \mathbf{c} \cdot \mathbf{d}$. This follows from the fact that $\text{Trace}(\mathbf{a}\mathbf{b}^T) = \mathbf{a} \cdot \mathbf{b}$ and $\text{Trace}(\mathbf{c}\mathbf{d}^T) = \mathbf{c} \cdot \mathbf{d}$. Now $(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T)^2 = (\mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T)^2$, or

$$\begin{aligned} (\mathbf{a} \cdot \mathbf{b})\mathbf{a}\mathbf{b}^T + (\mathbf{b} \cdot \mathbf{b})\mathbf{a}\mathbf{a}^T + (\mathbf{a} \cdot \mathbf{a})\mathbf{b}\mathbf{b}^T + (\mathbf{a} \cdot \mathbf{b})\mathbf{b}\mathbf{a}^T \\ = (\mathbf{c} \cdot \mathbf{d})\mathbf{c}\mathbf{d}^T + (\mathbf{d} \cdot \mathbf{d})\mathbf{c}\mathbf{c}^T + (\mathbf{c} \cdot \mathbf{c})\mathbf{d}\mathbf{d}^T + (\mathbf{c} \cdot \mathbf{d})\mathbf{d}\mathbf{c}^T. \end{aligned}$$

Further, since $\mathbf{a} \cdot \mathbf{b} = \mathbf{c} \cdot \mathbf{d}$, we have $(\mathbf{a} \cdot \mathbf{b})(\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T) = (\mathbf{c} \cdot \mathbf{d})(\mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T)$. Subtracting twice this amount from the previous equality we arrive at

$$\begin{aligned} -(\mathbf{a} \cdot \mathbf{b})\mathbf{a}\mathbf{b}^T + (\mathbf{b} \cdot \mathbf{b})\mathbf{a}\mathbf{a}^T + (\mathbf{a} \cdot \mathbf{a})\mathbf{b}\mathbf{b}^T - (\mathbf{a} \cdot \mathbf{b})\mathbf{b}\mathbf{a}^T \\ = -(\mathbf{c} \cdot \mathbf{d})\mathbf{c}\mathbf{d}^T + (\mathbf{d} \cdot \mathbf{d})\mathbf{c}\mathbf{c}^T + (\mathbf{c} \cdot \mathbf{c})\mathbf{d}\mathbf{d}^T - (\mathbf{c} \cdot \mathbf{d})\mathbf{d}\mathbf{c}^T, \end{aligned}$$

which reduces to $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T)^2 = (\mathbf{c}\mathbf{d}^T - \mathbf{d}\mathbf{c}^T)^2$. Using the result of lemma 3, we obtain $(\mathbf{a}\mathbf{b}^T - \mathbf{b}\mathbf{a}^T) = \pm(\mathbf{c}\mathbf{d}^T - \mathbf{d}\mathbf{c}^T)$. Adding $\mathbf{a}\mathbf{b}^T + \mathbf{b}\mathbf{a}^T = \mathbf{c}\mathbf{d}^T + \mathbf{d}\mathbf{c}^T$ to this equality we find that either $2\mathbf{a}\mathbf{b}^T = 2\mathbf{c}\mathbf{d}^T$, or $2\mathbf{a}\mathbf{b}^T = 2\mathbf{d}\mathbf{c}^T$. The conclusion follows using Lemma 2. ■

Bibliography

- Adiv, G. (1985) "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume PAMI-7, No 4, July.
- Adiv, G. (1985) "Inherent Ambiguities in Recovering 3-D Motion and Structure From a Noisy Flow Field," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, San Francisco, June.
- Aloimonos, J., & C. Brown, (1984) "Direct Processing of Curvilinear Sensor Motion from a Sequence of Perspective Images," *Proceedings of Workshop on Computer Vision: Representation and Control*, Annapolis, Maryland.
- Aloimonos, J., & A. Basu (1986a) "Determining the Translation of a Rigidly Moving Surface, without Correspondence," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, June.
- Aloimonos, J., & A. Basu (1986b) "Shape and Motion from Contour without Point-to-Point Correspondence: General Principles," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, June.
- Anandan, P. (1984) "Computing Dense Displacement Fields with Confidence Measures in Scenes Containing Occlusion," COINS TR-84-32, University of Massachusetts, Amherst, MA, December.
- Ballard, D.H., & O.A. Kimball (1983) "Rigid Body Motion from Depth and Optical Flow," *Computer Vision, Graphics, and Image Processing*, Vol. 22, No 1, April.

- Barnard, S.T., & W.B. Thompson (1980) "Disparity Analysis of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No 4, July.
- Bruss, A.R., & B.K.P. Horn (1983) "Passive Navigation," *Computer Vision, Graphics, and Image Processing*, Vol. 21, No 1, January.
- Buxton, B.F., H. Buxton, D.W. Murray, & N.S. Williams (1984) "3-D Solution to the Aperture Problem," *Proceedings of the sixth European Conference in Artificial Intelligence*, Pisa, Italy, September.
- Courant, R., & Hilbert, D. (1962) *Methods of Mathematical Physics*, Vol. II, John Wiley & Sons, New York.
- Cornelius, N., & T. Kanade (1983) "Adapting Optical Flow to Measure Object Motion in Reflectance and X-Ray Image Sequences," *ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion: Representation and Perception*, Toronto, Canada, April.
- Davis, L.S., Z. Wu, & S. Hanfang (1983) "Contour-Based Motion Estimation," *Computer Vision, Graphics, and Image Processing*, Vol. 23, No 3, September.
- Dreschler, L.S., & H.-H. Nagel (1982) "Volumetric Model and 3-D Trajectory of a Moving Car Derived from Monocular TV-Frame Sequences of a Street Scene," *Computer Vision, Graphics, and Image Processing* Vol. 20, No 3, November.
- Fang, J.Q., & T.S. Huang (1984) "Solving Three-Dimensional Small-Rotation Motion Equations: Uniqueness, Algorithms, and Numerical Results," *Computer Vision, Graphics, and Image Processing*, Vol. 26, No 2, May.
- Fennema, C.L., & W.B. Thompson (1979) "Velocity Determination in Scenes Containing Several Moving Objects," *Computer Graphics and Image Processing*, Vol. 9, No 4, April.
- Gibson, J.J. (1950) *The Perception of the Visual World*, Houghton Mifflin, Boston.
- Gibson, J.J., P. Olum, & F. Rosenblatt (1957) "Parallax and Perspective During Aircraft Landing," *American Journal of Psychology*, Vol. 68, No 2.
- Gibson, J.J. (1957) "Optical Motions and Transformations as Stimuli for Visual Perception," *Psychology Review*, Vol. 64, No 5.
- Gibson, J.J., & E.J. Gibson (1957) "Continuous Perspective Transformations and the Perception of Rigid Motions," *Journal of Experimental Psychology*, Vol. 54, No 2.
- Gordon, D.A. (1965) "Static and Dynamic Fields in Human Space Perception," *Journal of the Optical Society of America*, Vol. 55, 1965.
- Grimson, W.E.L. (1981) *From Images to Surfaces: A Computational Study of the Human Early Visual System*, MIT Press, Cambridge, MA.
- Hadamard, J. (1923) *Lectures on the Cauchy Problem in Linear Partial Differential equations*, Yale University Press, CT.

- Hay, J.C. (1966) "Optical Motion and Space Perception: An Extension of Gibson's Analysis," *Psychology Review*, Vol. 73, No 6.
- Helmholtz, H. van. (1925) "Helmholtz's Traetsie on Physiological Optics," *Optical Society of America*, (ed. J.P. Southall).
- Hildreth, E.C. (1983) *The Measurement of Visual Motion*, MIT Press, Cambridge, MA.
- Horn, B.K.P. (1974) "Determining Lightness from an Image," *Computer Graphics and Image Processing*, Vol. 3, No 1, December.
- Horn, B.K.P., & B.G. Schunck (1981) "Determining Optical Flow," *Artificial Intelligence*, Vol. 17.
- Horn, B.K.P., & M.J. Brooks (1985) "The Variational Approach to Shape From Shading," *Computer Vision, Graphics, and Pattern Recognition*, Vol 33, No 2, February.
- Horn, B.K.P. (1986a) *Robot Vision*, The MIT Press, and McGraw-Hill Book Company.
- Horn, B.K.P., & E.J. Weldon (1986) "Robust Direct Methods for Recovering Motion," submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.
- Horn, B.K.P. (1986b) "Rareness of Ambiguities in Motion Vision," in preparation.
- Ikeuchi, K., & B.K.P. Horn (1981) "Numerical Methods for Shape from Shading," *Artificial Intelligence*, Vol. 17.
- Jain, R. (1983) "Direct Computation of the Focus of Expansion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No 1, January.
- Jerian, C., & R. Jain (1984) "Determining Motion Parameters for Scenes with Translation and Rotation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No 4, July.
- Kanatani, K. (1985a) "Tracing Planar Surface Motion from a Projection without Knowing the Correspondence," *Computer Vision, Graphics, and Image Processing*, Vol. 29, No 1, January.
- Kanatani, K. (1985b) "Detecting the Motion of a Planar Surface by Line and Surface Integrals," *Computer Vision, Graphics, and Image Processing*, Vol 29, No 1, January.
- Koenderink, J.J., & A.J. van Doorn (1975) "Invariant Properties of the Motion Parallax Field due to the Movement of Rigid Bodies Relative to the Observer," *Optica Acta*, Vol. 22, No 9.
- Koenderink, J.J., & A.J. van Doorn (1976) "Local Structure of Movement Parallax of the Plane," *Journal of the Optical Society of America*, Vol. 66, No 7, July.
- Koenderink, J.J., & A.J. van Doorn (1981) "Exterosppecific Component of the Motion Parallax Field," *Journal of the Optical Society of America*, Vol. 71, No 8, August.

- Korn & Korn (1968) *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill Book Co.
- Lawton, D.T. (1983) "Processing Translational Motion Sequences," *Computer Vision, Graphics, and Image Processing*, Vol. 22, No 1, April.
- Le Guilloux, Y. (1986) "Structure from Motion, Acceleration and Taylor Series," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, June.
- Longuet-Higgins, H.C., & K. Prazdny (1980) "The Interpretation of a Moving Retinal Image," *Proceedings of the Royal Society of London, Series B*, Vol. 208.
- Longuet-Higgins, H.C. (1981) "A Computer Algorithm for Reconstructing a Scene from Two Projections," *Nature*, Vol. 293.
- Longuet-Higgins, H.C. (1984) "The Visual Ambiguity of a Moving Plane," *Proceedings of the Royal Society of London, Series B*, Vol. 223.
- Maybank, S.J. (1984) "The Angular Velocity Associated with the Optical Flow Field due to a Single Moving Rigid Plane," *Proceedings of the Sixth European Conference in Artificial Intelligence*, Pisa, Italy, September.
- Meiri, A.Z. (1980) "On Monocular Perception of 3-D Moving Objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No 6.
- Mitiche, A. (1984) "Computation of Optical Flow and Rigid Motion," *Proceedings of Workshop on Computer Vision: Representation and Control*, Annapolis, Maryland.
- Murray, D.W., & B.F. Buxton (1984) "Reconstructing the Optical Flow Field from Edge Motion: An Examination of Two Different approaches," *Proceedings of First Conference on AI Applications*, Denver.
- Nagel, H.-H. (1983a) "Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences," *Computer Vision, Graphics, and Image Processing*, Vol. 21, No 1, January.
- Nagel, H.H. (1983b) "On the Estimation of Dense Displacement Vector Fields from Image Sequences," *ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion: Representation and Perception*, Toronto, Canada, April.
- Negahdaripour, S., & B.K.P. Horn (1985a) "Direct Passive Navigation," AI Memo 821, MIT Artificial Intelligence Laboratory, Cambridge, MA, February.
- Negahdaripour, S., & B.K.P. Horn (1985b) "Determining 3-D Motion of Planar Objects from Image Brightness Patterns," *Proceedings of the Ninth International Joint Conference in Artificial Intelligence*, Los Angeles, CA, August.

- Negahdaripour, S., & B.K.P. Horn (1986a) "Direct Passive Navigation: Analytical Solution for Planes," *Proceedings of IEEE International Conference on Robotics and Automation*, San Francisco, April (also AI Memo 863, MIT Artificial Intelligence Lab, Cambridge, MA, August, 1985).
- Negahdaripour, S., & B.K.P. Horn (1986b) "Direct Passive Navigation," to appear in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November.
- Negahdaripour, S., & A. Yuille (1986) "Direct Passive Navigation: Analytical Solution for Quadratic Surfaces," unpublished work.
- Paquin, R., & E. Dubois (1983) "A Spatio-Temporal Gradient Method for Estimating the Displacement Field in Time-Varying Imagery," *Computer Vision, Graphics, and Image Processing*, Vol. 21, No 2, February.
- Prager, J.M., & M. Arbib (1983) "Computing the Optical Flow: The MATCH Algorithm and Prediction," *Computer Vision, Graphics, and Image Processing*, Vol. 24, No 3, December.
- Poggio, T., & V. Torre (1984) "Ill-Posed Problems and Regularization Analysis in Early Vision," AI Memo 773, MIT AI Lab, Cambridge, MA, April.
- Poggio, T., H. Voorhees, & A. Yuille (1985b) "Edge Detection," AI Memo 833, MIT AI Lab, Cambridge, MA.
- Prazdny, K. (1979) "Motion and Structure from Optical Flow," *Proceedings of the sixth International Joint Conference on Artificial Intelligence*, Tokyo, Japan, August.
- Prazdny, K. (1981) "Determining the Instantaneous Direction of Motion from Optical Flow Generated by a Curvilinearly Moving Observer," *Computer Graphics and Image Processing*, Vol. 17, No 3, November.
- Roach, J.W., & J.K. Aggarwal (1980) "Determining the Movement of Objects from a Sequence of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2, No 6, November.
- Rougée, A., B.C. Levy, & A.S. Willsky (1986) "Optical Flow Estimation inside a Bounded Domain," LIDS-P-1589, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, August.
- Scott, G.L. (1986) "Local and Global Interpretation of Moving Images," Ph.D. Thesis, University of Sussex, Sussex, England, June.
- Spivak, M. (1975) *A Comprehensive Introduction to Differential Geometry, I, II, III, IV, V*, Publish or Perish, Inc., Wilmington, Delaware.
- Terzopoulos, D. (1984a) "Multiresolution Computation of Visible-Surface Representations," Ph.D. Thesis, MIT, Cambridge, MA, January.

- Terzopoulos, D. (1984b) "Multigrid Relaxation Methods and Analysis of Lightness, Shading, and Flow," AI Memo 803, MIT AI Lab, Cambridge, MA, October.
- Tsai, R.Y., T.S. Huang, & W.L. Zhu (1982) "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch, II: Singular Value Decomposition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No 4, August.
- Tsai, R. (1983) "Estimating 3-D Motion Parameters and Object Surface Structures from the Image Motion of Curved Edges," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, Washington, D.C., June.
- Tsai, R.Y., T.S. Huang (1984) "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, No 1, January.
- Ullman, S. (1979) *The Interpretation of Visual Motion*, MIT Press, Cambridge, MA.
- Ullman, S. (1983) "Computational Studies in the Interpretation of Structure from Motion: Summary and Extension," AI Memo 706, MIT AI Lab, Cambridge, MA, March.
- Ullman, S. (1984) "Maximizing Rigidity: The Incremental Recovery of 3-D Structure from Rigid and Non-Rigid Motion," *Perception*, Vol. 13.
- Wallach, H., & D.N. O'Connell (1953) "The Kinetic Depth Effect," *Journal of Experimental Psychology*, Vol. 45, No 4, April.
- Waxman, A.M., & S. Ullman (1983) "Surface Structure and 3-D Motion from Image Flow: A Kinematic Analysis," CAR-TR-24, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, October.
- Waxman, A.M., B. Kamgar-Parsi, & M. Subbarao (1986) "Closed-Form Solutions to Image Flow Equations for 3-D Structure and Motion," CAR-TR-190, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, February.
- Waxman, A.M., & K. Wohn (1984) "Contour Evolution, Neighborhood Deformation and Global Image Flow: Planar Surfaces in Motion," CAR-TR-58, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, April.
- Webb, J.A., & J.K. Aggarwal (1982) "Structure from Motion of Rigid and Jointed Objects," *Artificial Intelligence*, Vol. 19, No 1, September.
- Wohn, K. (1984) "A Contour-Based Approach to Image Flow," Ph. D. Thesis, Center for Automation Research, University of Maryland, September.
- Yashida, M. (1983) "Determining Velocity Maps by Spatio-Temporal Neighborhoods from Image Sequences," *Computer Vision, Graphics, and Image Processing*, Vol. 21, No 2, February.

- Yasumoto, Y., Medioni, (1985) "Experiments in Estimation of 3-D Motion Parameters from a sequence of Image Frames," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, San Francisco, June.
- Yen, B.L., & T.S. Huang (1983) "Determining 3-D Motion/Structure of a Rigid Body Over 3 Frames Using Straight Line Correspondence," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, Washington, D.C., June.