Technical Report 1162

Three-Dimensional Motion Estimation Using Shading Information in Multiple Frames

Jean-Pierre Schott

MIT Artificial Intelligence Laboratory

This blank page was inserted to preserve pagination.

Three–Dimensional Motion Estimation Using Shading Information in Multiple Frames

by

Jean-Pierre Schott

Ing. Dipl., Ecole Supérieure d'Electricite (1981)
MSEE & EE, Massachusetts Institute of Technology (1982)

Submitted in Partial Fulfillment
of the Requirements for the
Degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology
September 1989

Signature of Author _____
Department of Electrical Engineering and Computer Science
August 11, 1989

Certified by _____
Berthold K.P. Horn
Thesis Supervisor

Accepted by _____
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Three–Dimensional Motion Estimation Using Shading Information in Multiple Frames

by

Jean-Pierre Schott

Submitted to the Department of Electrical Engineering and Computer Science
on August 11, 1989 in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Abstract

Traditionally motion and shading have been treated as two disjoint problems. On the one hand, researchers studying motion or structure from motion often assume uniform lighting conditions over the whole surface and good contrast at high spatial frequencies to minimize the effects of variations of the image irradiance of the patch as the surface moves. On the other hand, researchers primarily concerned with the shape from shading problem only consider static brightness data in order to recover the shape without considering the change of brightness induced by motion.

A new formulation for recovering the structure and motion parameters of a moving patch is presented. It is based on using the spatiotemporal derivatives of irradiance that are computed from a time–varying irradiance sequence and combined into a differential constraint equation. The new approach determines the rigid body motion and the structure of the patch directly from the irradiance sequence using *both* motion and shading information.

A new constraint equation, the full irradiance constraint equation (FICE), is derived. It links the spatiotemporal gradients of irradiance to the motion and structure parameters *and* the temporal variations of the surface shading. This equation separates the contribution to the irradiance spatiotemporal gradients of the gradients due to texture from those due to shading and allows the FICE to be used for textured and textureless surface. The new approach combining motion and shading information, leads directly to two different contributions: it can compensate for the effects of shading variations in recovering the shape and motion; and it can exploit the shading/illumination effects to recover motion and shape when they cannot be recovered without it. The FICE formulation is extended to multiple frames, and several methods are presented for efficiently computing the structure and motion parameters directly from a sequence of data.

Overall, the examples demonstrate the superiority of the FICE algorithms to the classical CE algorithms in two distinct areas: the accuracy of the results is higher for textured surfaces and a solution can be determined in the case of textureless surfaces.

Thesis Supervisor: Prof. Berthold K.P. Horn
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

Throughout the years of my graduate work at MIT, I had the privilege and the pleasure of working with and learning from many individuals. I wish to extend my appreciation to everyone who made this thesis possible and made my stay at MIT a memorable experience. In particular, I would like to extend my special gratitude to several individuals.

I would first like to express my sincerest thanks to Professor Berthold K.P. Horn, my thesis supervisor, who gave me the total freedom to study my area of interest. Working with him was a real pleasure, both academically and personally.

I am grateful to my thesis reader Dr. Arun Netravali, who motivated my interest in computer vision and supervised the beginning of this thesis, and to Professor Schreiber who provided financial support and academic advice during part of this research.

My thanks go to Pat O'Keefe who carefully reviewed my draft and provided many detailed comments. His friendship, our regular dinners and our discussions helped me through many difficulties.

Several friends also deserve special thanks: Michele Covell for a friendship which resisted many storms; Lori Lamel for the uncountable messages of support and encouragement; and Jerry Roylance for many talks and helpful gestures.

The MIT Artificial Intelligence Laboratory was a unique experience. Its mix of wildly different people and ideas, its atmosphere and its "crazyness" contributed to making it the most interesting place at MIT.

My thanks to Ellen Hildreth who reviewed my thesis and made comments that improved the present technical report.

And last, but not least, I thank my family for their neverending and unquestioning love. Distance never eroded their enthusiasm and their belief that, one day, I would be done.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology, supported by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014–85–K–0124.

[redacted]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, a lot of attention has been focussed on the problem of recovering the three–dimensional structure and motion of objects from the corresponding two–dimensional planar projections of these objects on the image plane. Advances in robot and mobile robot technology have led to an increased demand for reliable algorithms that can determine structure and motion of the surrounding environment from time-varying sequences. The problem arises in navigation, e.g. (Bruss and Horn 1983, Negahdaripour and Horn 1987), where the motion of an observer in a fixed environment is computed, object tracking, e.g. (Roach and Aggarwal 1980, Schalkoff and McVey 1982), where one or more objects are moving against a slowly changing background, object recognition and any other area where a machine interacts with its environment with the aid of a vision system.

Humans can rapidly integrate a large number of visual cues like stereo, shading and motion information and easily perform the task of recovering the three–dimensional world from a series of essentially two–dimensional projections of the world on their retinas. Most current machine vision algorithms do not provide data fusion nor do they use multiple sources of information to compute the three–dimensional representation, i.e. the relative or absolute depths at each point of an object and its motion. Currently prevailing algorithms either use stationary cues to recover the structure, as in the *shape from* x problems, where x can be shading, texture, silhouettes etc. or use motion cues to recover structure and motion. Each formulation has its own set of assumptions and limitations and little has been done in the way of combining information of different areas to increase the quality of the solution.

Traditionally motion and shading have been treated as two disjoint problems. On the one hand, researchers studying motion or structure from motion often assume uniform lighting conditions over the whole surface and good contrast at high spatial frequencies to minimize the effects of variations of the image irradiance (or more subjectively the brightness) of the patch as the surface moves. On the other hand, researchers primarily concerned with the shape from shading problem only consider static brightness data in order to recover the shape without considering the change of brightness induced by motion.

This thesis will combine the use of motion and shading information to estimate the three–dimensional motion and structure of an object from its time–varying sequence of image brightness. The new approach, combining motion and shading information, leads directly to two different contributions: it can compensate for the effects of shading variations in recovering the shape and motion; and it can exploit the shading/illumination effects to recover motion and shape when they cannot be recovered without it.

1.1 Problem Description

The most general task would be to recover the three–dimensional motion and structure of arbitrary, multiple objects from the time–varying sequence of 2–D projections of these objects onto the image plane. Such a general formulation is not yet within reach and needs to be specialized further by the type of motion and by the type and number of objects. One important special case is rigid body motion. It provides a compact way of expressing motion compared to the dense, 2–D velocity field computed in optical flow methods, and is quite realistic for a large variety of objects that are rigid or can be segmented into rigid subparts.

The extent of the task will be further restricted to a single moving rigid object to avoid dealing with the problem of scene segmentation which is beyond the scope of this thesis. Such segmentation can be facilitated by knowledge of the structure or depth map, or an estimate of it, since in many cases different objects can be separated on the basis of depth discontinuities. The availability of stereo data from which we can compute a depth map, or direct range information from sonar or radar scans, provides a possible solution to the segmentation problem.

Finally, the last restriction that will be imposed is on the type of object. Unless we are given the structure information in terms of a depth map, in order to recover rigid motion, we need

to assume some a priori structure, e.g. planar or curved patches, and determine the structure parameters, like surface normals and curvatures.

After taking into account the previous restrictions, the problem that we want to solve can be formulated in the following way:

Problem: *Given a time–varying sequence of image irradiance patterns obtained by imaging a single rigid object under inertial motion, and given an a priori structure and surface reflectance model, we want to recover the structure parameters and the instantaneous velocity of the object using shading and motion induced variations in the sequence of images.*

Since we are concerned with instantaneous velocity, a differential approach to the problem can be used and constraint equations that link the variations of irradiance, represented by its spatiotemporal gradients, to the temporal variation of the shading on the surface, can be derived. The problem can then be solved by a least–squares minimization of all the available data. The constraint equations can be extended temporally to several frames at a time, under mild additional assumptions, to increase the accuracy of the solution and decrease the noise sensitivity by augmenting the amount of data used in the minimization.

1.1.1 Shading Constraints

Shading is a physical phenomenon produced by the interaction of a light source with the surface of an object. The two–dimensional irradiance distribution, observed on the image plane, is related to the geometry and properties of the three–dimensional scene by the process of image formation (see section 2.1). Shading depends on many parameters of which the structure of the object and its position in space with respect to the light source and camera are also parameters relevant to the structure and motion recovery task, and need to be estimated.

In the shape from shading problem the goal is to recover the surface normal at every point given a surface reflectance and a light source model. Depending upon the formulation and assumptions, the light source direction and its strength are either given or estimated. Work on illuminant direction determination can be found in Pentland (1984), Brown and Ballard (Brown et al. 1982) and Lee and Rosenfeld (1985), while Brooks and Horn (1985) simultaneously estimate the smoothest normal field from a Lambertian surface and the source direction. In many cases the strong assumption that the surface is textureless, i.e. has constant reflectance

properties across it, is required to solve the shape from shading problem.

In the formulation of this thesis, an a priori surface model is required for which we can predict the spatial (across the surface) and temporal shading variations, given a reflectance model and a source model and position. In a multisensor situation the shape could be directly available in the form of range data, but in our case, image irradiance is the only input and a geometric model of the surface is assumed. In this context, the easiest surface models are parametrized patches that can be characterized by a small set of parameters, such as a unique normal for a planar patch or the normal and curvatures at a point of a quadratic patch. However, unlike the usual formulation of the shading problem, no restrictions are placed on the texture.

The shading equation relates the irradiance at a given point of the image to the normal at the corresponding point on the surface and to the position of the light source. The brightness constraint equation, more correctly called the full irradiance constraint equation (FICE), introduced here expresses the relationship between the spatial irradiance gradient of a given image point, the motion of the corresponding point on the surface, the temporal variation of the shading and the structure of the object. Given a point on the image plane, the shading constraint allows us to separate the temporal variations of irradiance due to the change in orientation of the surface, in motion with respect to the light source, from the spatiotemporal variations of irradiance due to the motion of the 3–D point being imaged (see Section 2.3).

The framework developed in the thesis is valid for any parameterized surface patch although the algebraic complexity increases much faster than the surface degree. For clarity in explanation, most of development in the next chapters will be for planar surfaces when a specific surface model is required. Extensions to the quadratic patches case will be treated in chapter 5.

1.1.2 Multiple Frames Processing

One of the main problems in dealing with real data is the inherent noisy nature of the data. Careful acquisition of the data with good video cameras reduces the noise levels, but sensor noise, spatial non–uniformity of the optics and quantization noise due to the conversion of the continuous analog pixel intensity to an 8 bit quantity, always contribute to the overall noise in the sequence.

The least–squares approach adopted in the algorithms allows us to minimize the effect of

the noise and obtain robust solutions. The technique is particularly robust in cases where the amount of data is very large in comparison to the number of parameters to estimate. The noise averaging property of the least–squares method is proportional to the square root of the amount of redundant data used. Given the spatial extent of the image, or equivalently the field of view, and a spatial sampling, the amount of data available per frame is fixed and the only way to increase it is to use multiple consecutive temporal frames. In the context of this thesis, "multiple" means at least three, since all motion algorithms require a minimum of two frames already.

Additional, mild assumptions on the motion are required in the multiple–frame formulation. In order to keep the number of unknown parameters constant, and benefit from the additional data, the instantaneous translation and rotation vectors need to be constant in the time interval spanned by the frames used in the minimization. Although the assumption seems very restrictive, typical motions can generally be treated as constant over a brief period of time. If the sequence is captured with a regular video camera, each field has a duration of 1/60 s, and the motion constancy approximation is well justified, for normal motion speed, over three to seven frames. The previous assumption can be relaxed and algorithms can be designed to accommodate slowly varying motion. In essence, constancy is assumed when computing the instantaneous motion vectors of the central frame, but the motion vectors are allowed to change slightly when the temporal window is moved one frame and the motion parameters for the next frame are computed. In practice, the motion vectors are made up of a base term, or initial velocity term, and an update term that is computed in subsequent frames.

1.2 Previous Approaches

Two major classes of methods, two–stage and direct, are used in computing motion and structure. In the two–stage methods, the structure and motion parameters are not computed directly from the brightness data but from an intermediate quantity that is derived from the data. Typical preprocessing operations are the computation of the optical flow or the determination of the correspondence of discrete features. In contrast, direct methods compute the motion vectors and recover the structure directly from the brightness field, bypassing the preprocessing stage. Two–stage approaches are further split into discrete and differential methods. Discrete meth-

ods deal with sparse data, or features, and generally compute finite motions while differential methods can determine the instantaneous velocities. The direct approaches use a differential formulation exclusively.

Although a lot of work in the field has been aimed at obtaining more robust solutions and reduced sensitivity to noisy data, it comes as a surprise that very little has been done to develop algorithms that use, globally, the temporally highly correlated data provided by time-varying sequences. Similar to the work in optical flow computation, algorithms have been developed to use multiple frames only in a weak way. The final estimate of the flow, or motion and structure parameters, is used as initial estimates in the processing of the next two frames of data. Many of these algorithms perform very well if a good enough initial estimate is available, and one can achieve a large reduction in the computation cost of subsequent estimates by using the previous frames estimates. Even though they are apparently using multiples frames, these latter algorithms are truly two–frame algorithms, and the quality of the successive estimates is highly dependent on the very first step and the ability to find an initial guess that is good enough for the algorithm. The Locally Constant Angular Momentum model (LCAM) of Huang (Huang et al. 1986) is such a pseudo multiframe algorithm, and is successively applied to pairs of frames in a sequence. True multiframe methods have been proposed and will be discussed in greater detail in section 1.2.4.

Finally, we will review some previous attempts in including shading in the motion and structure problem. The first two papers (Webb and Aggarwal 1983, Aloimonos 1987) are truly alternate solutions to the shape from shading problems, since they only recover the structure of the object, but use motion information, while Nagel (1989) addresses the problem of recovering both structure and rigid motion using shading information but fails to present a realistic shading model and does not go beyond the presentation of a model leaving open the question of the solution.

1.2.1 Discrete Motion Estimation

In discrete motion estimation the problem is twofold. First, we have to determine good features and compute an accurate disparity field, i.e. solve the correspondence problem, then we have to extract the motion and structure parameters from the disparity field. Usually, features

are points, corners or lines. In the second stage, the problem is reformulated with respect to intermediate variables, often called *essential parameters*, that are solved first. The final operation is to extract the motion and structure parameters from these variables.

Different schemes that use n points in p frames have been proposed, for example Tsai and Huang (1981) use 8 points in 2 images while Roach and Aggarwal (1980) use 5 points in 3 views, although, theoretically, 5 points in 2 frames are necessary and sufficient to recover the structure and motion parameters. Most of the proposed schemes lead to a system of nonlinear equations, although some algorithms for planar patches (Tsai and Huang 1984) only require the solution of a set of linear equations. The discrete approach has the major disadvantage of requiring the solution to the correspondence problem, that is the identification of corresponding image features between successive frames, a task that can be extremely difficult. By definition, the method relies on the spatial disparity of similar quantities and can run into trouble when feature points are too close together. Finally, although such schemes can be coded very efficiently and are computationally attractive, the method is very sensitive to noise and feature occlusion.

Ullman (1983) introduced a multiframe algorithm for wire frame objects that is relatively insensitive to errors in the discrete measurements. His algorithm maintains a model of the viewed object and updates that model each time a new frame is obtained. The model is updated in a way that minimizes the amount of nonrigid motion necessary to account for the changes observed in the new frame. This position–based model was later extended to a velocity–based model by Grzywacz and Hildreth (1987).

The problem of recovering motion and shape from widely separated frames, using known correspondence of several points in several frames, is similar to the classic photogrammetric problem of relative orientation that can be solved iteratively from the parallax equations, e.g. (Moffit and Mikhail 1980). Relative orientation is also a key issue in binocular stereo vision where we need to determine the relative orientation of one camera with respect to the other. Recently, Horn (1987) presented a new scheme that does not require an initial guess of the baseline and the rotation, for the recovery of relative orientation and discussed the existence of multiple solutions and their interpretation.

1.2.2 Rigid Body Motion From Optical Flow

In some of the two–stage differential approaches (Longuet-Higgins and Prazdny 1980, Waxman and Ullman 1983), the optical flow and its derivatives are used to determine the motion and the structure of the scene. First applied to planar patches, e.g. (Waxman and Wohn 1984), the method has been extended to curved surfaces, e.g. (Wohn and Waxman 1985) for quadratic patches, by approximating the patch in the neighborhood of the line of sight by its truncated (usually at the second order) Taylor series expansion. These methods, although very elegant mathematically, suffer from pronounced sensitivity to noise since they use higher derivatives of the optical flow or assume, unrealistically, that the optical flow and its spatial derivatives are known exactly.

Differential methods are very often based on a least-squares formulation that uses a dense field, like the optical flow, in conjunction with a constraint equation and, sometimes, a model of the structure of the patch. For example, Adiv (1985), Ballard and Kimball (1983) and Bruss and Horn (1983), do not assume any structure and use the optical flow field at every point of the image in order to minimize the sensitivity to noisy data while Netravali and Salz (1985) performs successive linearization around the current estimate without assuming any given structure.

1.2.3 Direct Estimation of Motion and Structure Parameters

More recently, several methods (Negahdaripour and Horn 1987, Horn and Weldon 1988) that directly use the image brightness and its gradients, *without* computing the optical flow have been proposed. These differential methods compute the motion parameters directly from the observed data without requiring the intermediate step of optical flow computation and rely on Horn and Schunck's constraint equation (Horn and Schunck 1981) rewritten in terms of the instantaneous rotation $\boldsymbol{\omega}$, instantaneous translation $\mathbf{t}$ and the reciprocal of the depth Z at each point. The brightness constraint equation links the spatiotemporal irradiance gradients to the rigid body motion parameters and to the object structure by the depth value at each point. The authors present iterative and closed–form solutions for planar surfaces (Negahdaripour and Horn 1987) and an iterative solution to the quadratic patch case (Negahdaripour 1986).

Direct methods have also been used to estimate special motion of arbitrary surfaces. Horn and Weldon (1988) developed robust algorithms to recover pure rotation or pure translation of

a rigid body and demonstrated it on real data.

1.2.4 Multiple–Frame Algorithms

In addition to the apparent multiframe methods described in the introduction of this section, two multiframe, discrete or feature based approaches have been proposed, a geometrical approach and a stochastic estimation approach using Kalman filtering. In addition to these discrete methods that all rely on correspondence of features, Bolles and others developed a totally new framework that relies on very dense time sampling of image sequences.

The geometrical approach of Shariat and Price (1986) is reminiscent of the previously mentioned discrete approaches where there is a choice of how many points in how many frames to consider in order to have a fully determined problem. In their work, they describe a 5 frames, 1 feature algorithm although they also mention possible use of 4 frames and 2 features or 3 frames and 3 features. The model consists of a rigid object, with constant motion, observed from a temporally–uniform sampled sequence. The algorithm is based on the idea that, if the translation of the rigid body is compensated for, every feature on the object will trace a circle in space. They infer the necessary equations from geometrical relations. The convergence of the algorithm is highly dependent on the choice of the initial guess and no proof of convergence or uniqueness of the solutions is given.

In their study Broida and Chelappa (1985, 1986) cast the problem of recovering motion parameters into a classical parameter estimation problem in the presence of noise and use a Kalman filtering and a maximum likelihood (ML) approach. In their model, they assume the structure of the object and the image coordinates of the object match points and only consider transparent objects to insure that every match point is always visible throughout the sequence. They present both a recursive solution (Broida and Chellappa 1986) that uses an extended Kalman filtering approach and a batch solution (Broida and Chellappa 1985) that computes an ML estimate of the parameters. Additionally, it is assumed that the noise level associated with the match points is known in the recursive approach, and that the initial angular orientation is available in the ML formulation. These techniques perform very well on a wire frame synthetic model and achieve results very close to the theoretical limit as given by the Cramer-Rao bound, but have the major drawback of assuming that the *correspondence problem* has already been

solved.

Bolles and other (Bolles and Baker 1985, Bolles et al. 1987) opened up a new dimension in structure recovery by making time the preferred direction in their epipolar plane image analysis. Although their goal is not to recover motion but to recover the structure of the scene from motion sequences *knowing* the motion, their innovative use of the highly correlated temporal information might spawn novel ideas in the field of motion and structure recovery and is worth outlining here. In their method, they collect high frame rate video sequences, stack them into a spatiotemporal parallelepiped, slice the data along the temporal dimension to localize features in the temporal slice, deduce the 3–D location of the features and reconstruct the structure of the scene. In their initial paper (Bolles and Baker 1985), they assumed a viewing direction perpendicular to the direction of motion so that features were linear, but later (Bolles et al. 1987) relaxed the assumption and only required a known arbitrary motion.

1.2.5 Combining Shape, Shading and Motion

1.2.5.1 Using Motion Information in Recovering Shape From Shading

Webb and Aggarwal (1983) incorporate motion information in their solution to the shape from shading problem. Their analysis relies heavily on the assumption of rigid body motion, planar surface and orthographic projection, but does not require explicit knowledge of the rigid motion. The algorithm applies to a Lambertian surface with arbitrary spatial variations in albedo and a single distant light source of known direction, and starts by considering a single point correspondence between two adjacent frames. The next step matches respective neighboring regions of the matched point, by assuming that the texture of the two patches are related by an affine transformation (small rigid body motion assumption) and that the image intensities can only change by a multiplicative factor. If $\mathbf{n}^T = (-p, -q, 1)$ represents the normal at a match point, these two constraints can be expressed by two conic equations in p and q. Once the normal at the matched point is computed, the constraints are propagated to the neighboring points within the match region.

In contrast to the previous approach, Aloimonos and Bandopadhay (1987) use the full optical flow between two consecutive frames and recover both the shape of the object and the light source direction. Their formulation assumes that the images are viewed under orthographic

projection, that the surface has a Lambertian reflectance function with a constant known albedo and that the retinal correspondence, i.e. the optical flow, is known. The source direction can only be computed assuming a planar surface, while the computation of the shape only requires the surface to be smooth, with the added strong restriction that the shape is known exactly at the occluding boundary and that the normal at that boundary is parallel to the image plane.

1.2.5.2 Motion Recovery Using Shading

In a recent paper, Nagel (1989) introduces a new constraint equation that incorporates a shading term and discusses other extensions to the classical constraint equation. The stated goal of the paper is to derive a constraint equation using perspective projection, differential geometry and radiometric information. Unfortunately, the results fall short of the goals. His derivations, using differential geometry, tend to be more complicated than the traditional ones without increasing the intuitive feel for the results. It is a little surprising that he presents, in great detail, the derivation of the discrete motion equations, that require the solution of the correspondence problem, and then proceeds to differentiate the equations, obtaining the classical optical flow equations, and only uses the latter in the subsequent derivations.

More serious problems with his approach are the use of Schunck's divergence equation (Schunck 1985) and his choice of a shading model. In the former instance, he fails to recognize that Schunck's equation is only applicable to texture density and is wrong in the context of an image brightness constraint equation. Contrary to his claim, his shading model has nothing to do with a Lambertian reflectance model of the surface but only relies on the well known optical imaging $\cos^4 \alpha$ law where α represents the angle of the incident rays entering the lens assembly. His model is inaccurate, since, in practice, any well–built system can be calibrated to eliminate the $\cos^4$ dependence and this phenomenon is negligible compared to the vignetting effect[1] that occurs in real lenses.

All the problems outlined in this summary, compounded by the fact that Nagel does not even attempt to solve his model, clearly reduce the usefulness of his contribution.

[1] Vignetting is the reduction of the light gathering process caused by the partial occlusion of inclined incident rays by components of the lens assembly system.

1.3 Machine Vision and Image Processing

Strong competition exists between the machine vision and the image processing communities. Each camp advocates that their algorithms are more robust or realistic in solving the *same* problem i.e. recovering the motion between two frames of a video sequence. Much of the problem is the failure to recognize that, until recently, the two communities were solving different problems.

Machine vision algorithms are trying to recover the *true* three–dimensional motion of the objects and use this information in path planning, collision avoidance or robot navigation, while the image processing algorithms, especially in the image coding field, only try to compute a two–dimensional field that reduces the entropy of the residual error signals between the original images and the coded one. The computed field may or may not have any relationship with the optical flow depending on the methods used, but is a valid field as long as a coding gain is obtained, compared to the non *motion–compensated* coding schemes.

More recently, there has been a demand for the computation of *true* motion fields for television imagery and their use in motion–compensated interpolation, where the goal is to reconstruct as faithfully as possible the missing frames from temporally and spatially sampled frames. The newest algorithms are very close to the traditional machine vision optical flow algorithms. The new requirements have, in essence, started to bridge the gap between the image processing and machine vision communities. The former group has not yet started to use and compute rigid body motion since typical broadcast television imagery is not composed of rigid bodies. However, such an analysis is relevant when it comes to estimate and compensate for the motion of the camera with respect to the scene. The difficulty in that situation is that the scene is not static and it might be difficult, or impossible, to separate the rigid motion of the camera from the individual motions of the elements of the scene being imaged.

1.4 Goal of the Thesis

In this thesis, we investigate the problem of combining shading and motion information to recover the structure and motion of a parameterized rigid patch with respect to a fixed observer, from a sequence of image irradiance The goal is to recover the three–dimensional motion by

observing the image irradiance, its temporal and spatial derivatives and by assuming, or estimating, a shading model for the parametrized surface, given a set of lighting conditions. More specifically, the goal can be refined to the following three components:

1. Establish a theoretical and computational framework to incorporate the a priori knowledge of shading conditions and use it, in conjunction with the motion cues, to refine the structure and motion solutions and extend them to previously difficult cases under the traditional assumption of brightness constancy.

2. Evaluate qualitatively and quantitatively the importance and relevance of shading information with respect to the added complexity of the implementation and to the accuracy of the solution.

3. Extend the analysis to longer sequences to use efficiently the additional data and improve the noise performance of the solutions for real data.

Rigid body motion of objects with respect to a fixed observer is considered, as opposed to the problem of passive navigation, which considers rigid body motion of the camera with respect to a fixed background. In the former case, an object based reference frame is usually chosen and the motion of the camera, or observer, evaluated with respect to the stationary scene and *light sources*. In the latter case, a camera based reference frame is usually considered and the motion of the rigid bodies is expressed with respect to the camera. The latter distinction is very important because, in the case of passive navigation, with objects that have a reflectance function independent of the viewing direction, there is no motion induced shading variations.

1.5 Summary of Results and Contributions

A new formulation for recovering the structure and motion parameters of a moving patch is presented. It is based on the spatiotemporal derivatives of irradiance that are computed from a time–varying irradiance sequence and combined into a differential constraint equation. The new approach determines the rigid body motion and the structure of the patch directly from the irradiance sequence using *both* motion and shading information.

First, a new constraint equation (CE), the full irradiance constraint equation (FICE), is derived. It links the spatiotemporal gradients of irradiance to the motion and structure parameters *and* the temporal variations of the surface shading. This equation separates the contribution to the irradiance spatiotemporal gradients of the gradients due to texture from those due to shading and allows the FICE to be used for textured and textureless surface.

The theory of the new CE is first developed for a generic surface and an arbitrary shading model; the minimization equations that define the motion and structure parameters, are derived in the general case. The models are subsequently specialized and various implementation are given for planar and quadratic patches; various shading models and numerical examples, that use synthetic and real data, are presented for different combinations of surface, light source geometry and surface reflectance.

The FICE formulation is extended to multiple frames, and several methods are presented for efficiently computing the structure and motion parameters directly from a sequence of data. Several examples that focus on the behavior of the multiframe algorithms in the presence of noise are described. In particular, it is shown that the accuracy of the results is greatly improved in the presence of noise with the multiframe implementations.

Overall, the examples demonstrate the superiority of the FICE algorithms to the classical CE algorithms in two distinct areas: the accuracy of the results is higher for textured surfaces and a solution can be determined in the case of textureless surfaces.

In this thesis, at least three major contributions to the problem of recovering the structure and motion parameters were made: a constraint equation that allows the concurrent use of motion and shading information is formulated; a study of the importance of shading in rigid motion estimation is presented and situations in which the classical CE is too inaccurate to be used is determined; the FICE is expanded to multiple frames and a global formulation is provided. In addition to these contributions, insight is gained in the solution of nonlinear system of equations and in the challenge offered by real data.

Prior to this work, there was no attempt to bridge the fields of motion estimation and shape from shading and a single algorithm or class of algorithms could not recover the structure and motion parameters from a moving patch with an albedo that can either be constant (textureless surface) or arbitrary but smooth. In addition, there was no real understanding of the domain of

validity of the classical CE that was used indiscriminately in all situations. Our work presents evidence that, although the classical CE is a very useful and excellent approximation in many cases, in other cases it is significantly in error and greatly improved results can be computed with the novel formulation.

1.6 Thesis Overview

Because the use of shading information and multiple frames is decoupled, the theory of shading information within a constraint equation and the of multiple frames is presented separately. These algorithms are independent and can be used separately or together. Algorithms using shading information can be used with a minimum of two frames while algorithms using multiple frames can use the conventional constraint equation as described in Negahdaripour (Negahdaripour and Horn 1987).

In Chapter 2 an overview of the image formation process and representation of motion fields is presented. The Full Irradiance Constraint Equation (FICE) and the incremental FICE are derived and an extension of these equations to second order derivatives presented. Presently available numerical methods to solve systems of nonlinear equations are surveyed and the problem of efficient and robust computation of spatiotemporal irradiance derivatives treated. In Chapter 3, different shading models, the regular constraint equation, as derived by Negahdaripour (Negahdaripour and Horn 1987), and the FICE model are examined and discussed. Specific planar patch implementations and examples are featured in Chapter 4, while extensions to quadratic patches and corresponding examples are presented in Chapter 5. In Chapter 6, three multiple–frame constraint equations are derived and examples are shown. Finally, a summary of the results and the conclusions are presented in Chapter 7.

Chapter 2

Problem Formulation for Two Frames With Shading

Image formation produces a sequence of irradiance patterns from snapshots of a real three–dimensional object illuminated by light sources and moving in space. Motion fields mathematically represent the motion of the image of the object and shading characterizes the "appearance" of the object interacting with its lighting environment. This chapter analyses the image formation process, the representation of the motion of an object and the shading variations induced by motion, and presents algorithms that compute the structure and motion parameters from the irradiance sequence.

The two–frame case is described in a generic way that does not rely on a specific object shape or shading model and introduces all the tools and mathematical techniques needed for the study. Algorithms are built incrementally from a simple, purely two–frame algorithm that uses motion cues and shading information to recover the motion and structure parameters, to a more complex formulation that recovers incremental motion from a stack of frames. The overall purpose of this chapter is to provide the theoretical framework and to set up a computational framework that is shape and shading model independent.

2.1 Image Formation

An analysis of the principles of image formation is important to the understanding of the methods for recovering structure and motion from brightness images. The analysis reveals two main aspects of the image formation process, the geometric process by which a point from the three–dimensional surface projects onto a point in the image plane, and the photometric process that determines the irradiance of the points in the image plane.

2.1.1 Geometric Model

The analysis of a real lens–based camera is complicated. For regular focal length and normal to wide angle lenses, the actual lens can be approximated by a pin–hole camera placed at a distance equal to the effective focal length F of the lens from the image plane. Since light travels in a straight line from a point in the object surface to a point in the image plane, the transformation from the three–dimensional world to the projected two–dimensional image is a perspective transformation defined by

$$\frac{\mathbf{r}}{F} = \frac{\mathbf{R}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \tag{2.1}$$

where $\mathbf{R} = (X, Y, Z)^T$ represents the coordinates of the 3–D point in the world frame (camera based coordinate system), $\mathbf{r} = (x, y, F)^T$ the coordinates of the projected point and $\hat{\mathbf{z}}$ is the unit normal on the optical axis (see Figure 2.1). The 3–D vector $\mathbf{r}$ represents the projected point in the image plane and the 2–D vector $\mathbf{x} = (x, y)^T$ denotes the coordinates of the image point in the image plane, i.e. $\mathbf{x}$ is the restriction of $\mathbf{r}$ to the two–dimensional subspace of the image plane.

Although only normal or wide angle lenses are taken into consideration, and therefore only the perspective projection model is used, it should be noted that zoom and telephoto lenses can under certain circumstances be nicely and more easily approximated by orthographic projection.

An immediate consequence of the perspective projection equation (2.1) is that an image point with coordinates $\mathbf{r}$ can be computed uniquely from the coordinates $\mathbf{R}$ of the corresponding object point, *but* a unique object point cannot be determined from a given image point unless the distance along the ray is known (structure), since *any* point on the ray can be backprojected to the original point. The problem of recovering $\mathbf{R}$ from $\mathbf{r}$ is also known as the *inverse optics*

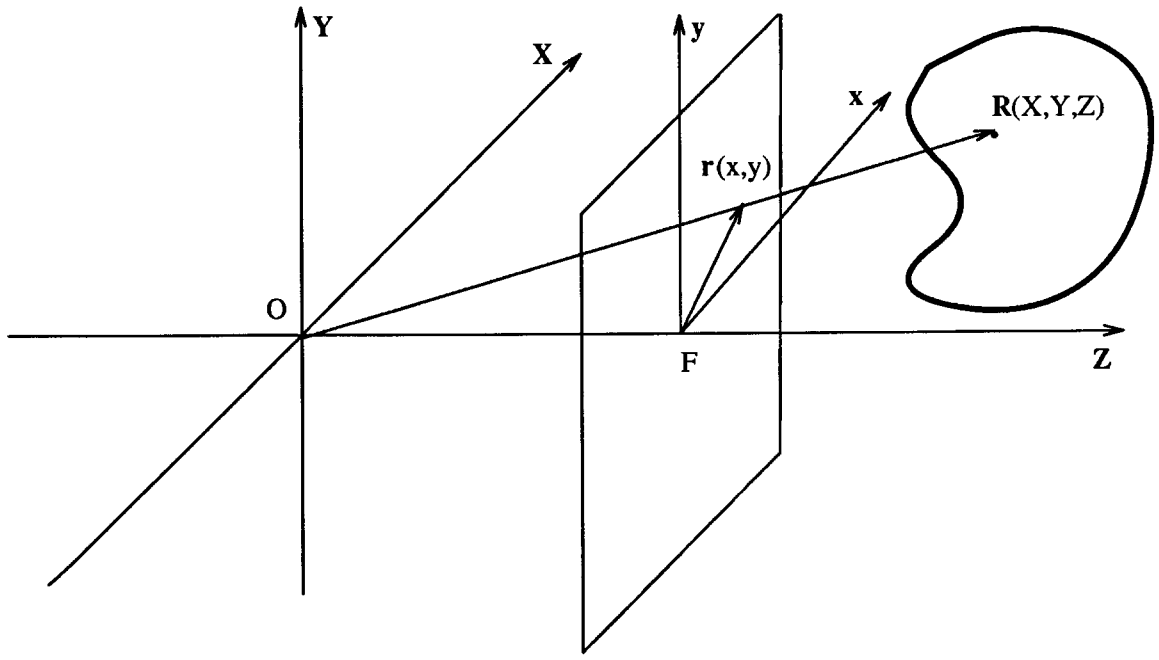

Figure 2.1: Perspective transformation

problem. It is an ill–posed problem that cannot be solved unless additional assumptions or constraints are introduced.

Pin–hole cameras have problems; if the size of the pin–hole is infinitely small, diffraction occurs, if the size is too big our analysis no longer holds and a point on the object is imaged as a small circle. Lenses overcome these problems to a large extent; they gather a finite amount of light while keeping the object surface in sharp focus.

2.1.2 Photometric Model

We will briefly show how to compute the image irradiance E, also informally called image brightness, and introduce the concepts of scene radiance and surface reflectance.

2.1.2.1 Computing Image Irradiance

The image irradiance, or apparent brightness of the surface, depends on the microstructure of the surface, the distribution of the incident light and the orientation of the surface with respect to the viewer and the light sources. It is formally defined as the power per unit area of radiant energy falling on a surface. The scene radiance L, or informally scene brightness, is defined

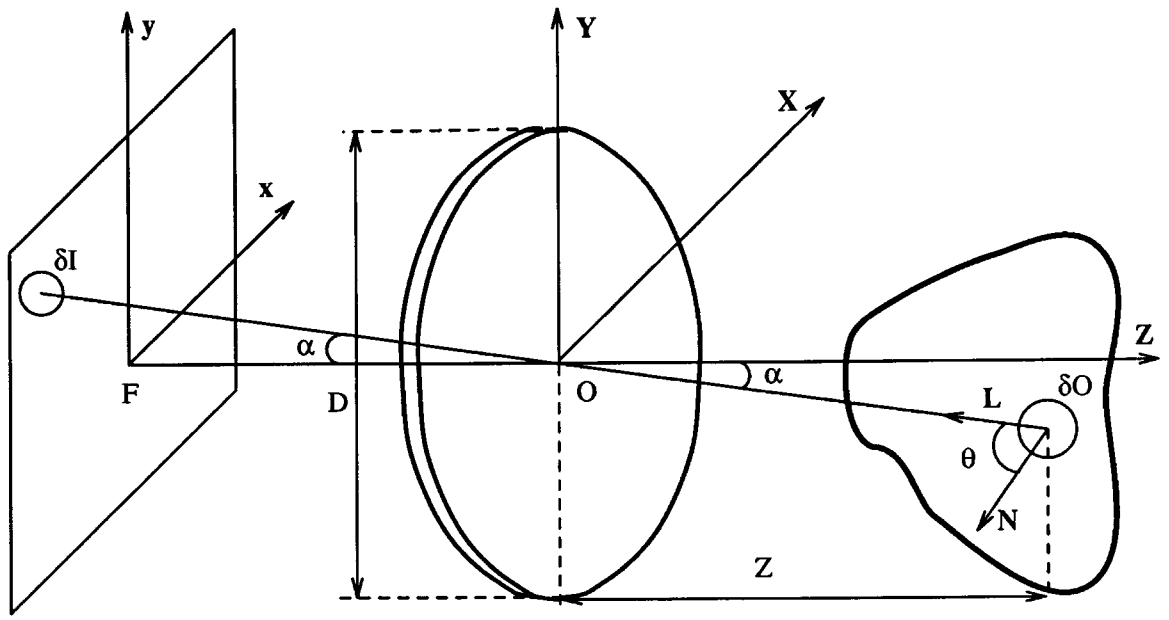

Figure 2.2: Image formation for a lens based optical system

as the power per unit foreshortened area emitted in a unit solid angle, and is *not* an intrinsic property of the surface. It depends on the reflectance property of the surface and can change with motion and illumination.

Let us consider an object patch δO that maps into the image patch δI through the lens (see Figure 2.2). Since rays passing through the center of the lens are not deflected, the solid angles $\Omega_{\delta I}$ and $\Omega_{\delta O}$ subtended at the center of the lens by δI and δO are equal, i.e.

$$\Omega_{\delta I} = \frac{\delta I \cos\alpha}{(F/\cos\alpha)^2} = \Omega_{\delta O} = \frac{\delta O \cos\theta}{\left((\mathbf{R}\cdot\hat{\mathbf{z}})/\cos\alpha\right)^2}$$

where α is the angle between the incident rays from the light source and the optical axis, and θ is the angle between the surface normal and the rays at δO. Taking the ratio of the solid angles, we obtain

$$\frac{\delta O}{\delta I} = \left(\frac{\cos\alpha}{\cos\theta}\right)\left(\frac{F}{\mathbf{R}\cdot\hat{\mathbf{z}}}\right)^2. \tag{2.2}$$

Let L be the radiance of a point of the patch δO on the object surface. It is the result of the light from the sources of illumination being reflected from the surface of the object and depends upon the location and nature of the sources of illumination, and upon the orientation and reflectance of the surface. The amount of light δL emanating from the surface patch δO,

and concentrated on the image patch δI, is given by, e.g. (Horn 1986) or (Horn and Sjoberg 1979),

$$\delta L = L\delta O \cos\theta\pi \left(\frac{d}{\mathbf{R}\cdot\hat{\mathbf{z}}}\right)^2 \cos^3\alpha, \tag{2.3}$$

where d is the diameter of the lens. If we neglect the losses in the lens, the image irradiance E, defined as the ratio of δL and δI, can be expressed as

$$E = L\frac{\pi}{4}\left(\frac{1}{F/d}\right)^2 \cos^4\alpha \tag{2.4}$$

using (2.2) and (2.3). Equation 2.4 is one expression of the well–known $\cos^4$ optics law. Very often the quantity F/d is used as a direct specification of a lens, and is called the effective F–number. In practice, systems are often calibrated to eliminate the $\cos^4$ dependence.

2.1.2.2 Scene Radiance and Surface Reflectance

Radiance is a directional quantity that depends on the direction of the illumination, characterized by the polar angle θ_i between the normal to the surface $\hat{\mathbf{n}}$ and the direction of the rays $\hat{\mathbf{L}}$, and by the azimuthal angle ϕ_i between the perpendicular projection of $\hat{\mathbf{L}}$ and any chosen line in a plane perpendicular to $\hat{\mathbf{n}}$. Nicodemus (Nicodemus et al. 1977) introduced the Bidirectional Reflectance Distribution Function (BRDF) that relates the incident irradiance δE in the direction (θ_i, ϕ_i), to the reflected radiance δL in the direction (θ_e, ϕ_e) (see figure 2.3 for the notations). The BRDF is defined as

$$BRDF(\theta_i, \phi_i; \theta_e, \phi_e) = \frac{\delta L(\theta_e, \phi_e)}{\delta E(\theta_i, \phi_i)} \tag{2.5}$$

and it determines how bright a surface illuminated in one direction appears when viewed from another direction. If the illumination source is distributed over a range of directions (θ_i, ϕ_i) (extended source), the total reflected radiance is obtained by integrating equation 2.5 over the solid angle of incidence.

The BRDF is a complete description of the relationship between scene radiance and image irradiance but is too complex to use since, in practice, it needs to be determined experimentally for each type of surface. Fortunately, there are some types of theoretical surface reflectance that can be easily modeled and that approximate some actual surfaces nicely. We will present two of these surfaces, the ideal Lambertian diffuser and the ideal specular reflector.

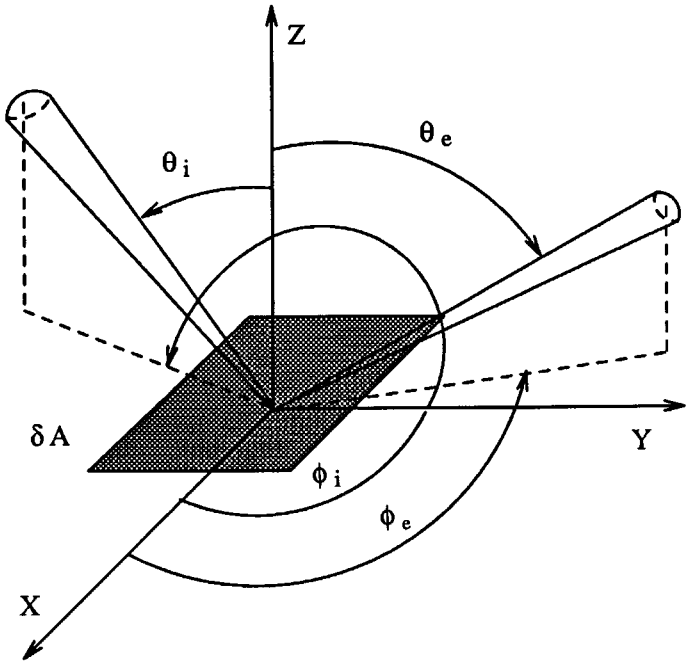

Figure 2.3: Local geometry of incident and reflected rays for the definition of the bidirectional reflectance distribution function (from figure 6 of Horn and Sjoberg (1979)).

An ideal Lambertian surface is a perfect diffuser that appears equally bright from all viewing directions and reflects all incident light, absorbing none. Although no real surface exactly behaves like this, tightly pressed powders of highly transparent material like barium sulfate and magnesium carbonate come close. Matte white paint, opal glass, rough papers and snow are somewhat worse approximations. The BRDF of an ideal Lambertian surface has a constant value $1/\pi$ and the radiance of a surface illuminated by a single point source is given by

$$L(\theta_i, \phi_i) = \frac{1}{\pi} \cos\theta_i E = \frac{1}{\pi}(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}})E.$$

The ideal specular reflector, very well approximated by polished metals, reflects all the light arriving from (θ_i, ϕ_i) into the direction $(\theta_i, \phi_i + \pi)$. Its BRDF can be expressed as

$$BRDF_{specular}(\theta_i, \phi_i; \theta_e, \phi_e) = \frac{\delta(\theta_e - \theta_i)\delta(\phi_e - \phi_i - \pi)}{\sin\theta_i \cos\theta_i}$$

where $\delta()$ represents the Dirac function. The scene radiance relates to the image irradiance simply by, $L(\theta_e, \phi_e) = E(\theta_e, \phi_e - \pi)$.

2.2 Images, Motion and Motion Fields

This section discusses the representation of three–dimensional motion, two–dimensional motion fields and their relationship. The special case of rigid body motion is developed in some detail and the relationship between motion and observed image sequences is discussed.

2.2.1 Representation of Motion

2.2.1.1 Image Motion Field

We saw in section 2.1 that an image is formed by the projection of a three–dimensional scene onto the target of the camera. In general, each point in the image corresponds to a unique point on the surface of the object. Transparent or translucent surfaces violate that rule and will be excluded. The relative motion of an object and the camera results in the motion on the image plane of the projections of the object points. The motion field is defined as the two–dimensional vector field of instantaneous velocities of points in the image plane as a function of position and time and is denoted by $\mathbf{v}(\mathbf{x}, t)$. These velocities are related to the three–dimensional velocities of the corresponding scene points through the perspective projection of the camera.

The motion will generally induce a spatiotemporal variation in the image plane irradiance, and it is the analysis of these variations that allows us to estimate the motion which has occurred. It should be noted that the motion is not always observable from the image irradiance even though a nonzero motion field might exist everywhere. The case of a rotating textureless sphere illuminated by a fixed light source is such an example. The image of the sphere is nonuniform, due to shading effects, but the image irradiance *does not* change with time.

A related concept, which is widely used, is that of the *optical flow*, defined as the apparent motion of the image brightness pattern (Horn and Schunck 1981). Optical flow is defined only in terms of the image and the apparent motion may be due to true motion, illumination effects, or both. The optical flow is not uniquely determined by its definition and does not correspond to any physical reality.

When dealing with temporally sampled images a quantity of interest, closely related to the velocity field, is the displacement field. It establishes the correspondence between the points in the image at time t_0 and the points in previous and subsequent frames at time $t_0 - kT$, where

$k \in \{..., -2, -1, 1, 2, ...\}$ and T is the temporal sampling rate. In the context of this thesis, *motion field* will be used to refer to either a velocity or a displacement field.

2.2.1.2 Velocity and Displacement Fields

The projection of each object point traces out a trajectory in the image plane during the time it is visible in the image. Trajectories can be specified by the function $\mathbf{c}(t; \mathbf{r}_0)$ which gives spatial location at time t of the point which was at the spatial location $\mathbf{r}_0$ at time t_0. From this definition, it is clear that $\mathbf{c}(t_0; \mathbf{r}_0) = \mathbf{r}_0$. Let us denote by $t_i(\mathbf{r}, t_0)$ the starting time and by $t_j(\mathbf{r}, t_0)$ the ending time of the trajectory of the point $\mathbf{r}$. The velocity field gives the rate of change of position at a given time for each pixel on the frame i.e.

$$\mathbf{v}(\mathbf{r}, t) = \frac{d}{dt}\mathbf{c}(t; \mathbf{r}_0)\bigg|_{t=t_0} = \mathbf{v}\big(\mathbf{c}(\mathbf{r}; t_0), t_0\big)$$

and, as expected the displacement field is related to the velocity field by integration. If $\mathbf{d}(t_1; \mathbf{r}, t_0)$ represents the displacement of $\mathbf{r}$ from the time t_0 to t_1 ($t_1 > t_0$)

$$\mathbf{d}(t_1, \mathbf{r}, t_0) = \int_{t_0}^{t_1} \mathbf{v}\big(\mathbf{c}(\mathbf{r}; t_0), t\big)\, dt.$$

The velocity field corresponds to the projection of the motion of the points in space and can be expressed directly in terms of these points. Let $\mathbf{R}$ be a point in space with coordinates (X, Y, Z) in a reference frame attached to the camera. We saw in section 2.1 that the image of the corresponding point in the image frame is $\mathbf{r} = (x, y, F)^T = F\mathbf{R}/(\mathbf{R} \cdot \hat{\mathbf{z}})$ where F is the effective focal length of the lens and $\hat{\mathbf{z}}$ is the optical axis. The point $\mathbf{R}$ is in relative motion with respect to the camera with the velocity $\mathbf{v} = \dot{\mathbf{R}}$, where dotted quantities represent time derivatives. The corresponding motion of the projection of $\mathbf{R}$ on the image plane is $\dot{\mathbf{r}} = (\dot{x}, \dot{y}, 0)^T$ (or $\dot{\mathbf{x}} = (\dot{x}, \dot{y})^T$), if the analysis is restricted to the image plane subspace, and can be expressed as

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

This derivative can be rewritten in terms of $\mathbf{r}$ and rearranged:

$$\dot{\mathbf{r}} = \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}}\big(\hat{\mathbf{z}} \times (\dot{\mathbf{R}} \times \mathbf{r})\big). \tag{2.6}$$

2.2.2 Parametric Representation of Motion Fields

The previous description of the motion field is general and does not place any constraint on the local form of these motion fields. In fact, the degrees of freedom of the fields can be reduced by assuming a certain parametric representation associated with a restricted class of motion. One important subclass of motions is rigid body motion.

If the point $\mathbf{R}$ is assumed to lie on a surface patch which can be considered as a rigid body, then the instantaneous motion of all the points of the patch can be described by

$$\dot{\mathbf{R}}(t) = \dot{\mathbf{R}} = \boldsymbol{\omega} \times \mathbf{R} + \mathbf{t} = \boldsymbol{\Omega}\mathbf{R} + \mathbf{t}, \tag{2.7}$$

where

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

is the (3×3) skew–symmetric matrix isomorphic to the instantaneous rotation vector $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)^T$, and $\mathbf{t}$ represents the instantaneous translation vector. Substituting equation 2.7 of the rigid body motion into equation 2.6 gives the expression of the velocity field of the images of points on the patch of rigid body in question, in terms of the three–dimensional motion parameters

$$\dot{\mathbf{r}} = \hat{\mathbf{z}} \times \left(\mathbf{r} \times \left(\frac{\mathbf{r}}{F} \times \boldsymbol{\omega} - \frac{\mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}} \right) \right). \tag{2.8}$$

The vectorial equation 2.8 represents the traditional optical flow equation and can be written in the more familiar component form:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} -\frac{xy}{F} & \left(F + \frac{x^2}{F}\right) & -y \\ -\left(F + \frac{y^2}{F}\right) & \frac{xy}{F} & x \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} + \frac{1}{\mathbf{R} \cdot \hat{\mathbf{z}}} \begin{pmatrix} Ft_x - xt_z \\ Ft_y - yt_z \end{pmatrix}. \tag{2.9}$$

Appendix A presents a comparison between the equations of this section, which use the instantaneous motion, and the optical flow equations which can be derived directly from the finite motion equation.

Motion fields are most easily visualized by means of needle diagrams, an array of vectors indicating the magnitude and direction of the velocity on a selected grid of points (Figure 2.4 is

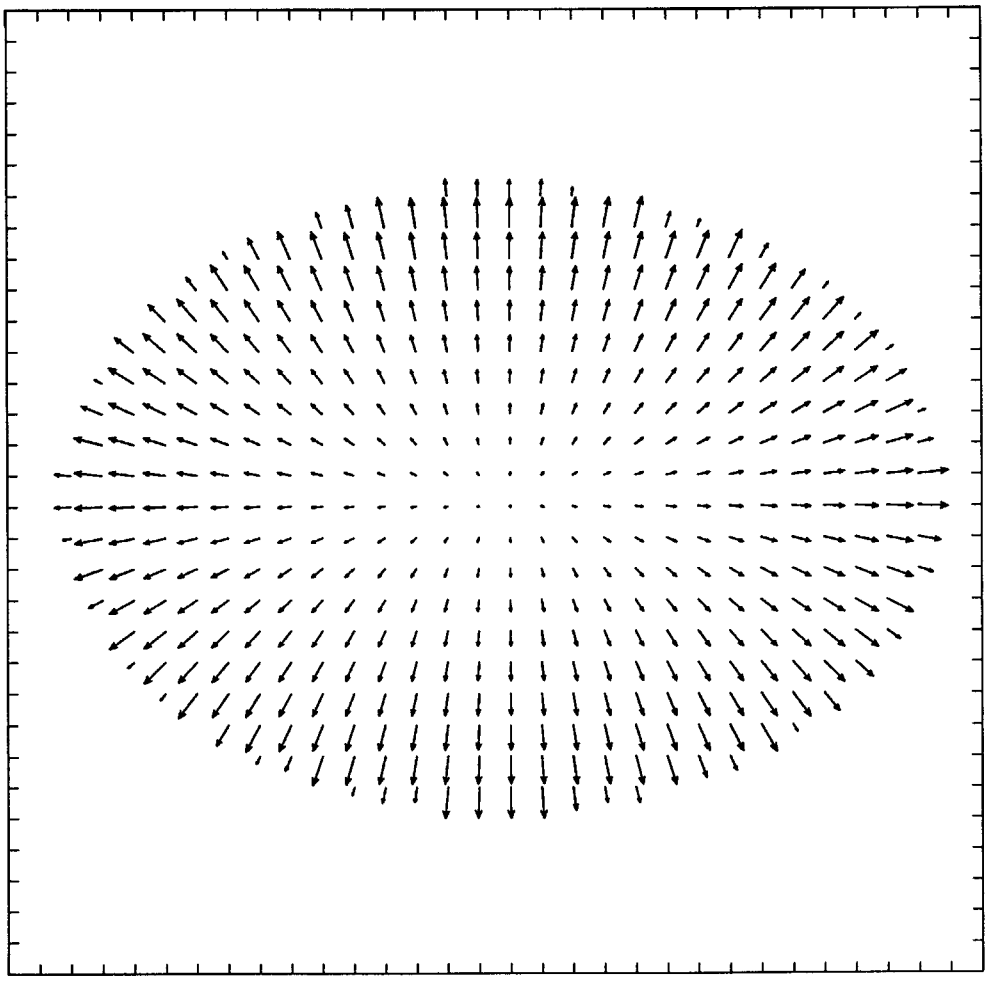

Figure 2.4: Needle diagram of a optical flow field representing a zoom along the optical axis

an example of the flow field induced by a pure translation along the optical axis) or by a color image where the hue indicates the direction and the saturation the magnitude of the field.

2.2.3 Time–Varying Images and Motion Fields

Having characterized motion by means of motion trajectories and motion fields, we can now relate the observed time–varying images to these motion fields. As seen in section 2.1, image irradiance $E(\mathbf{x}, t)$ is measured at each point of the image plane and the irradiance of the projected image point depends, in general, on the position of the point in space, on the local orientation of the patch, on the reflectance of the surface and on the light sources. The fundamental assumption in classical motion estimation is that the image irradiance varies *slowly*

along the motion trajectory. In practice, shading variations have been ignored or considered negligible *vis a vis* the spatial changes due to the texture of the surface. In most cases, no quantitative or convincing qualitative explanations have been formulated to justify such an assumption. In a recent paper, Horn (1988) provides some justification for the brightness change constraint equation by showing that, in the case of a sinusoidal grating, it is possible to neglect the change due to viewing direction or illumination as long as there is good contrast at high spatial frequencies. Section 3.3 analyses the validity of such an assumption as a function of the texture on the surface.

2.2.3.1 Classical Motion Constraint Equation

Assuming that the irradiance of a patch remains approximately constant as the surface moves in the environment, that is $dE/dt = 0$, the frame difference signal can be related to the displacement and to the spatiotemporal gradients of the image intensity. Limb and Murphy (1975) and, to a greater extent, Cafforio and Rocca (1983) were the first to use implicitly what Horn and Schunck called the *constraint equation* (CE) that can be expressed in the form (Horn and Schunck 1981)

$$E_{\mathbf{x}} \cdot \dot{\mathbf{x}} + E_t = 0. \tag{2.10}$$

In this equation $E_{\mathbf{x}} = \dfrac{\partial E}{\partial \mathbf{x}} = \left(\dfrac{\partial E}{\partial x}, \dfrac{\partial E}{\partial y}\right)^T$ represents the spatial gradient of the irradiance E with respect to $\mathbf{x} = (x, y)^T$, the image coordinates of the perspective projection of point $\mathbf{R}$ onto the image plane at effective focal distance F, and $E_t = \dfrac{\partial E}{\partial t}$ is the temporal gradient of the irradiance.

Although many, more or less correct, derivations of the CE have been presented, the above formulation, to be meaningful, corresponds to tracking a specified point $\mathbf{R} = (X, Y, Z)^T$ on the 3–D surface, or equivalently, if the point remains visible throughout the motion, its projection $\mathbf{r} = (x, y, F)^T$, or $\mathbf{x} = (x, y)^T$ on the image plane and to consider its temporal variations. Let us consider the projected trajectory specified by the function $\mathbf{c}(t; \mathbf{r}_0)$. The value of the image irradiance $E(\mathbf{x}, t)$ along the trajectory $\mathbf{c}(t; \mathbf{r}_0)$, $E(\mathbf{c}(t; \mathbf{r}_0), t)$ can be considered as a one-dimensional time signal denoted $s(t; \mathbf{r}_0)$. Then, the assumption of *no change* in intensity along the motion trajectory from the initial time t_i to the final time t_f is

$$s(t; \mathbf{r}_0) = E(\mathbf{r}_0, t_0), \qquad t_i(\mathbf{r}_0, t_0) \leq t \leq t_f(\mathbf{r}_0, t_0) \tag{2.11}$$

and the constraint equation is obtained by taking the derivative of (2.11) with respect to t, applying the chain rule and evaluating at $t = t_0$.

The constraint equation can also be interpreted in terms of a directional derivative of the brightness function $E(\mathbf{x}, t)$. If $\nabla_{\mathbf{a}} E$ denotes the directional derivative of E with respect to the arbitrary vector $\mathbf{a} \in R^3$, where $R^3 = R^2_{spatial} \times R_{time}$, then

$$\nabla_{\mathbf{a}} E = \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \nabla E,$$

where ∇E is the spatiotemporal gradient of $E(x, y, t)$, i.e. $\nabla E = (E_x, E_y, E_t)^T$.

If we consider the vector $\mathbf{a} = (\dot{\mathbf{x}}, 1)^T$ tangent to the motion trajectory, the directional derivative along this vector is given by

$$\nabla_{\mathbf{a}} E = \frac{1}{\|\mathbf{a}\|}(\dot{\mathbf{x}} \cdot E_{\mathbf{x}} + E_t) = \frac{1}{\|\mathbf{a}\|}(\dot{\mathbf{r}} \cdot E_{\mathbf{r}} + E_t)$$

which is equal to zero under the assumption of constant irradiance of the moving patch.

2.2.3.2 Relationship Between Displacement Field and Image Sequence

When dealing with a time sampled image, it is more natural to describe the motion in terms of a displacement field $\mathbf{d} = \mathbf{d}(t - T; \mathbf{x}, t)$. Under the assumption of irradiance constancy, the Displaced Frame Difference (DFD) denoted by $\mathbf{D}_E(\mathbf{x}, t, \mathbf{d})$ will be zero i.e.

$$\mathbf{D}_E(\mathbf{x}, t, \mathbf{d}) = E(\mathbf{x}, t) - E(\mathbf{x} - \mathbf{d}(t - T; \mathbf{x}, t), t - T) = 0. \tag{2.12}$$

A first–order Taylor series of the term $E(\mathbf{x}-\mathbf{d}, t-T)$ in (2.12), produces an equivalent expression to the motion constraint equation (2.10) in terms of the displacement field and the DFD:

$$\mathbf{D}_E(\mathbf{x}, t, \mathbf{d}) + \mathbf{d} \cdot \nabla_{\mathbf{x}} E(\mathbf{x}, t - T) = 0. \tag{2.13}$$

If we assume that the velocity is constant in the time interval $[t - T, t]$, $\mathbf{d} = \mathbf{v}T = \dot{\mathbf{x}}T$ and the previous equation can be written as

$$\frac{\mathbf{D}_E(\mathbf{x}, t, \mathbf{d})}{T} + \dot{\mathbf{x}} \cdot \nabla_{\mathbf{x}} E(\mathbf{x}, t - T) = 0. \tag{2.14}$$

Equation 2.14 is equivalent to the motion constraint equation (2.10), as T approaches 0, since $\frac{\partial E}{\partial t} = \lim_{T \to 0} \frac{\mathbf{D}_E(\mathbf{x}, t, \mathbf{d})}{T}$. In practice, only equation 2.13 should be used when dealing with

sampled images. The infinitesimal motions need to be approximated by finite displacements and the DFD value is a much better approximation to the temporal derivative than the usual first difference (see section 2.5).

A more general form of equation 2.13 is obtained by taking a Taylor series of the irradiance function $E(\mathbf{x}-\mathbf{d}, t-T)$ about the point $(\mathbf{x}-\mathbf{d}_0, t-T)$

$$\mathbf{D}_E(\mathbf{x}, t, \mathbf{d}_0) + (\mathbf{d}-\mathbf{d}_0)\cdot\nabla_{\mathbf{x}} E(\mathbf{x}-\mathbf{d}_0, t-T) = 0. \tag{2.15}$$

Equation 2.15 is identical to (2.13) for $\mathbf{d}_0 = 0$ but may be a better approximation if $\mathbf{d}_0$ is close to the true displacement because all the quantities are very small and the equation is an excellent approximation to the differential motion constraint equation. These equations will be reformulated directly in terms of the rigid body motion parameters in section 2.3.

2.2.3.3 Constraint Equation for Sampled Images

The constraint equation (2.10), derived in the previous section, only applies to the image as a function of continuous space and time. In the following development, the effects of quantization on the amplitude of the image irradiance value will be neglected and only the three–dimensional spatiotemporal sampling of the image sequence will be considered. Let Λ_u be a sampling lattice associated with the intensity image $u(\mathbf{x}, t)$ in the 3–D space–time coordinate system. Let $\left(T_u^h, T_u^v, T_u^t\right)^T$ be the sampling periods and let $\mathbf{x}_i$ be a 2–D image point from the sampling lattice. The sampled intensity image is $u_s(\mathbf{x}_i, t) = u(\mathbf{x}, t)$ where $(\mathbf{x}_i, t) = \left(kT_u^h, lT_u^v, mT_u^t\right)^T \in \Lambda_u$.

If the spectrum of the continuous image is limited to a unit cell of the reciprocal lattice Λ_u^*, the irradiance signal can be uniquely reconstructed from its samples by means of an interpolation formula of the form

$$u(\mathbf{x}, t) = \sum_{\mathbf{x}_i, \tau} u_s(\mathbf{x}_i, \tau)\Phi_i(\mathbf{x}-\mathbf{x}_i, t-\tau)$$

where

$$\Phi_i(\mathbf{x}, t) = K\int_W e^{(\boldsymbol{\omega}_{\mathbf{x}}\cdot\mathbf{x}+\omega_t t)} d\boldsymbol{\omega},$$

and W represents the baseband in the image plane. However, if the original image is not suitably bandlimited with respect to the sampling lattice, error–free reconstruction is not possible and aliasing will occur. Spatial aliasing can be easily avoided by low–pass filtering the analog video signal *before* spatially sampling it. The main problem, in a regular video camera, is the temporal

sampling. Let us consider a 3–D signal $u(\mathbf{x},t)$ obtained from a 2–D signal $u_0(\mathbf{x})$ by uniform translational velocity $\mathbf{v}$, $u(\mathbf{x},t) = u_0(\mathbf{x} - \mathbf{v}(t - t_0))$. The 3–D Fourier transform of $u(\mathbf{x},t)$ is related to the 2–D Fourier transform of $u_0(\mathbf{x})$by the relation

$$\mathcal{F}[u(\omega_x,\omega_y,\omega_t)] = \mathcal{F}[u_0(\omega_x,\omega_y)]e^{-(\omega_x v_x+\omega_y v_y)t_0}\delta(\omega_x v_x + \omega_y v_y + \omega_t).$$

If the signal $u_0(\mathbf{x})$ is bandlimited to $(|\Omega_x|,|\Omega_y|)$ then the signal $u(\mathbf{x},t)$ is bandlimited to $|\Omega_x,\Omega_y,\Omega_t|$ where

$$\Omega_t = \Omega_x|v_x| + \Omega_y|v_y|, \tag{2.16}$$

and temporal aliasing is avoided if

$$|\Omega_t| < \Omega_x|v_x| + \Omega_y|v_y|. \tag{2.17}$$

Equation 2.16 shows that in the simple case of an object under uniform translation, the temporal bandwidth Ω_t is a linear function of the magnitude of the translational velocity $\mathbf{v}$ and therefore, for velocity magnitude $\|\mathbf{v}\|$ greater than one, the signal should be sampled fast enough temporally to avoid temporal aliasing. The temporal anti–aliasing condition described by equation 2.17 is never met, or even approached, in present day video cameras and temporal aliasing will always be present.

In practice the continuous signal $u(\mathbf{x},t)$ is not reconstructed exactly from the sampled signal $u_s(\mathbf{x}_i,t)$ but $u(\mathbf{x},t) = \sum u_s(\mathbf{x}_i,\tau)\Psi_i(\mathbf{x} - \mathbf{x}_i, t - \tau) + \epsilon(\mathbf{x},t) = \widetilde{u}(\mathbf{x},t) + \epsilon(\mathbf{x},t)$ where $\widetilde{u}$ is the interpolated value of the image irradiance, Ψ is an interpolation kernel and $\epsilon(\mathbf{x},t)$ represents the interpolation error functional which may be partially due to aliasing or partially due to a suboptimal interpolation kernel. The choice of Ψ has a substantial effect on the accuracy of the solution. It will be discussed, in section 2.5 in the context of surface fitting and irradiance derivative interpolation and computation. For a sampled image, the motion constraint equation (2.10), can be expressed in terms of the interpolated continuous irradiance function and takes the form

$$\widetilde{u}_t + \mathbf{v}\cdot\nabla_{\mathbf{x}}\widetilde{u} + \epsilon_t + \mathbf{v}\cdot\nabla_{\mathbf{x}}\epsilon = 0,$$

but will usually be approximated by

$$\widetilde{u}_t + \mathbf{v}\cdot\nabla_{\mathbf{x}}\widetilde{u} = 0.$$

2.3 Full Irradiance Constraint Equation

In the previous section, the motion constraint equation was derived under the assumption of no change in intensity along the motion trajectory. In this section, a motion constraint equation called the *full irradiance change constraint equation* (FICE), that takes into account the change in intensity along the motion trajectory is presented. Mathematically, the FICE is the equation that defines the total derivative of the irradiance function $E(\mathbf{r},t)$, i.e.

$$\frac{dE(\mathbf{r},t)}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} \tag{2.18}$$

Equation 2.18 will be derived geometrically next, to gain insights in its physical meaning.

Let us consider a segment of rigid body (Σ, t_0) at time t_0 and the displaced segment (Σ, t_1) at time $t_1 = t_0 + dt$ (see Figure 2.5). Let $M(\mathbf{R}_0, t_0)$ be a generic point on (Σ, t_0) that projects to $\mathbf{r}_0(\mathbf{R}_0, t_0)$ on the image plane (Π). Let $P'(\mathbf{S}_1, t_1)$ be the corresponding point on (Σ, t_1) that also projects onto $\mathbf{r}_0$, and $P(\mathbf{S}_0, t_0)$ the point on (Σ, t_0) such that $P' = \mathcal{B}(P)$ where $\mathcal{B}$ represents the rigid body motion of the surface Σ. By construction, $P(\mathbf{S}_0, t_0)$ and $P'(\mathbf{S}_1, t_1)$ represent the same point on the surface and have the same coordinates (α, β), called surface coordinates, with respect to a coordinate system attached to the surface, the same albedo $\rho = \rho_\lambda(\alpha,\beta)$[1], but have a different surface radiance due to the different orientation of the patches (Σ, t_0) and (Σ, t_1) relative to the light source. If we denote by $\overline{P(\mathbf{R},t)}$ the projection of the point P of space coordinates $\mathbf{R}$ at time t, and $E\left(\overline{P(\mathbf{R},t)},t\right)$ the corresponding image irradiance, we have the relation

$$E\left(\overline{P'(\mathbf{S}_1,t_1)},t_1\right) - E\left(\overline{P(\mathbf{R}_0,t_0)},t_0\right) = \underbrace{E\left(\overline{P'(\mathbf{S}_1,t_1)},t_1\right) - E\left(\overline{M(\mathbf{S}_0,t_0)},t_0\right)}_{\Delta E(\mathbf{S}_1,\mathbf{S}_0)} + \underbrace{E\left(\overline{M(\mathbf{R}_0,t_0)},t_0\right) - E\left(\overline{P(\mathbf{S}_0,t_0)},t_0\right)}_{\Delta E(\mathbf{R}_0,\mathbf{S}_0)}. \tag{2.19}$$

The left hand side of equation 2.19 represents the temporal variations of the irradiance of the point S, of surface coordinates (α,β), due to the change of orientation of the surface Σ, i.e. $E(\mathbf{r}_0,t_0) - E(\mathbf{r}_1,t_1)$. The first part of the right hand side of the previous equation, $\Delta E(\mathbf{S}_1,\mathbf{S}_0)$, represents the change of irradiance due to the motion of the observed point P. Since the

[1] $\rho_\lambda(\alpha,\beta)$ denotes the surface albedo at wavelength λ and surface coordinates (α,β). In practice, the continuous wavelength data are not available and only their integrals with a small number of filter functions corresponding to different channels are accessible.

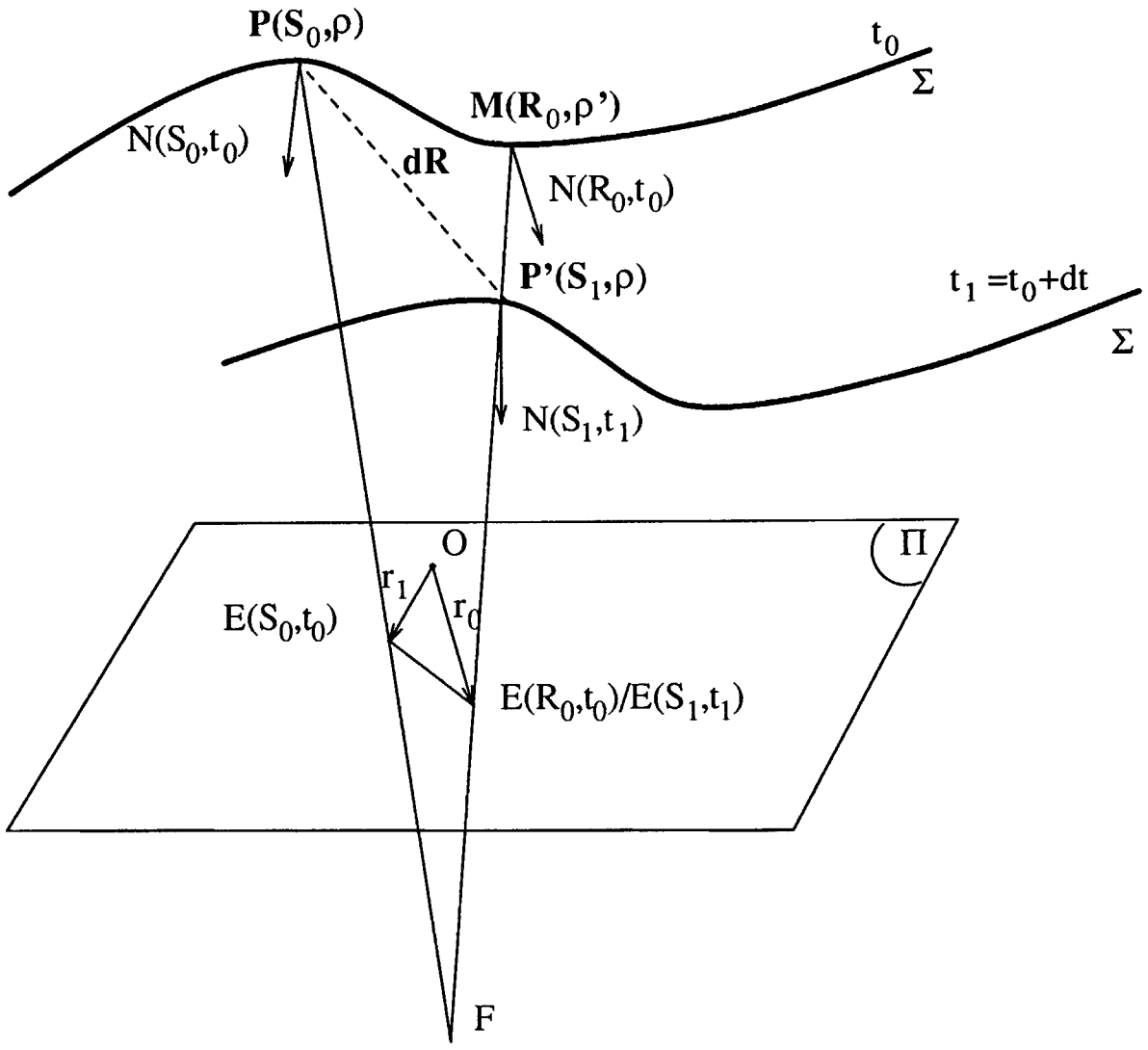

Figure 2.5: Relationship between the 3–D surface, the texture on the surface and the projection of the surface onto the image plane for the full irradiance constraint equation.

points P' and M project onto the same image point $\mathbf{r}_0$, $\Delta E(\mathbf{S}_1, \mathbf{S}_0) = E(\mathbf{r}_0, t_1) - E(\mathbf{r}_0, t_0)$ and $\lim_{dt \to 0} \Delta E(\mathbf{S}_1, \mathbf{S}_1) = \dfrac{\partial E(\mathbf{r}_0, t)}{\partial t}$. $\Delta E(\mathbf{R}_0, \mathbf{S}_0)$ represents the difference of irradiance on the points P and M on the surface (Σ, t_0) and $\lim_{dt \to 0} \Big(\Delta E(\mathbf{R}_0, \mathbf{S}_0) = E(\mathbf{r}_1, t_0) - E(\mathbf{r}_0, t_0)\Big) = d\mathbf{r} \cdot E_{\mathbf{r}_0}(\mathbf{r}_0, t_0)$. Taking the limit, as dt goes to 0, of (2.19) yields the FICE differential equation (2.18), that can also be written as

$$E_{\mathbf{r}} \cdot \dot{\mathbf{r}} + E_t = \dot{E}(\mathbf{r}, t) = \dot{E}, \tag{2.20}$$

where $\dot{E}$ represent the temporal variations of the shading model.

Using the expression of $\dot{\mathbf{r}}$ in terms of the instantaneous rigid body motion parameters $(\boldsymbol{\omega}, \mathbf{t})$ (see equation 2.8), we can rewrite the FICE (2.20) in the form

$$E_t - \mathbf{v} \cdot \boldsymbol{\omega} - \frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{R} \cdot \hat{\mathbf{z}}} = \dot{E} \tag{2.21}$$

where

$$\mathbf{s} = (E_{\mathbf{r}} \times \hat{\mathbf{z}}) \times \mathbf{r} = \begin{pmatrix} -FE_x \\ -FE_y \\ xE_x + yE_y \end{pmatrix} \quad \text{and} \quad \mathbf{v} = -\mathbf{s} \times \frac{\mathbf{r}}{F} = \begin{pmatrix} \frac{xy}{F}E_x + \left(F + \frac{y^2}{F}\right)E_y \\ -\left(F + \frac{x^2}{F}\right)E_x - \frac{xy}{F}E_y \\ yE_x - xE_y \end{pmatrix}$$

Equation 2.21 represents the FICE in the case of rigid body motion. Negahdaripour and Horn (1987) derived the equation in the constant irradiance case, i.e. when $\dot{E} = 0$ is assumed.

2.3.1 Motion and Structure Equations

A smooth C^p surface can be represented by a p^{th} order polynomial patch that corresponds to the truncated Taylor series expansion of the surface in the neighborhood of the line of sight. If the viewer frame is oriented so that the $\hat{\mathbf{z}}$-axis is along the focal axis of the camera, the p^{th} order polynomial patch is given by

$$Z(X, Y) = Z_0 + \sum_{i+j=1}^{p} \frac{1}{(i+j)!} \frac{\partial^{i+j} Z}{\partial X^i \partial Y^j} X^i Y^j.$$

The reciprocal of the depth can be expressed directly with a p^{th} order Taylor expansion in terms of the image plane coordinates x, y. If $\mathcal{O}(\epsilon^p)$ denotes higher order terms, the Taylor series of $1/Z$ is given by the expression

$$\frac{1}{Z} = \frac{1}{Z_0} - \frac{1}{Z_0} \sum_{i+j=1}^{p} \frac{C_{i,j}}{(i+j)!} \frac{\partial^{i+j}(1/Z)}{\partial x^i \partial y^j} \frac{x^i y^j}{F^{i+j}} + \mathcal{O}(\epsilon^p). \tag{2.22}$$

If the nonlinear terms are neglected, the patch is planar and can be represented by 3 quantities, $\mathbf{Z}_0 = (0\ 0\ Z_0)$ the intersection of the patch with the line of sight, and Z_X and Z_Y that give the orientation of the patch. The normal $\mathbf{n}$ to the patch can be expressed in terms of spatial derivatives of the depth Z by $\mathbf{n} = (-Z_X\ -Z_Y\ 1)^T$. A planar patch representation can be assumed if the object under consideration is a plane or a polyhedra, or if the curvature of the nonplanar patch is small. Since a planar patch is characterized by a single normal, the surface radiance will be uniform across the patch, if the shading variations due to a nearby light source are neglected and the reflectance function, e.g. Lambertian diffuser, is independent of the viewing direction.

A planar patch, in the viewer frame, can be represented by the equation

$$\begin{aligned} Z = Z_0 + Z_X X + Z_Y Y &\iff \mathbf{R}\cdot\mathbf{n} = Z_0 \\ &\iff \frac{\mathbf{R}\cdot\hat{\mathbf{z}}}{F}(\mathbf{r}\cdot\mathbf{n}) = Z_0 \\ &\iff \frac{1}{Z} = \frac{\mathbf{r}\cdot\mathbf{n}}{FZ_0} \end{aligned} \tag{2.23}$$

i.e. the reciprocal of the depth $\mathbf{R}\cdot\hat{\mathbf{z}} = Z$ can be expressed *exactly* as a linear functional in the image coordinates $\mathbf{r}$.

If the second–order terms are included in the Taylor expansion, the surface is approximated by a quadratic patch that is specified by the point $\mathbf{Z}_0$, the normal $\mathbf{n}$ at that point and three curvatures that can be reduced to two principal curvatures in an object frame oriented along the principal curvatures. Unlike the planar patch case, there is a different normal at every point and therefore there are shading variations induced by the curvature of the surface. As a result shading information is much richer, but is much harder to use since the normal at every point is required.

2.3.1.1 Equation, Measurement and Parameter Counting

Let us consider an arbitrary surface patch Σ undergoing a rigid motion specified, at time t, by the instantaneous, constant, motion vectors $\mathbf{t}, \boldsymbol{\omega}$ and let us denote σ its perspective projection onto the image plane. The surface Σ is specified by a Lambertian reflectance functional and a smooth albedo $\rho_\lambda(\alpha, \beta)$, and is illuminated by a single light source characterized by it position $\mathbf{l}$, in the camera centered frame, and its intensity L_0. For the purpose of counting equations,

parameters and measurements, the surface σ is assumed to fully cover the image plane of size $(n \times m)$.

The observables are, equivalently, the irradiance values of a sequence of images, the spatiotemporal derivatives of the irradiance or the irradiance values of the first frame and the temporal derivatives of the brightness values in subsequent frames, and we want to recover the motion parameters $\mathbf{t}, \boldsymbol{\omega}$, the structure of the surface $Z(X,Y)$, or a set of parameters characterizing the parametric surface patch, and, optionally, the direction and intensity of the light source. The problem is specified by two constraint equations, the FICE (2.21) that relates the spatiotemporal changes in irradiance to the motion and structure parameters and to the shading variations at every point of the projected surface σ, and the shading equation (SE) that specifies the value of the image irradiance at every point of σ from the source direction $\hat{\mathbf{L}}$, the surface normal $\hat{\mathbf{n}}$ and the value of the albedo at the corresponding point of Σ.

For a sequence of p frames, we observe, in general, $p \times (n \times m)$ independent irradiance values from which we can extract at most $(n-1) \times (m-1)$ spatial gradients and $(p-1) \times (n \times m)$ temporal derivatives. We need to estimate 6 motion parameters (the components of $\mathbf{t}$ and $\boldsymbol{\omega}$), 4 parameters for the light source (position and intensity) [2] and $n \times m$ albedo and $n \times m$ depth values. Naive counting would associate an unknown value to the depth and albedo for every point in *each* p frames without realizing that, once a depth and albedo is associated with an image plane point, or equivalently to a surface point, their variations in subsequent frames are driven by the rigid motion and the points are tracked throughout the sequence. In practice, the number of unknowns fluctuates slightly since points of the surface disappear and appear at the edge of the image plane but this first–order effect can be ignore in global, average counting. The previous, naive, count would suggest that a p–frame sequence produces $p \times n \times m$ independent measurements, $6+3+2nm$ unknown parameters and could therefore be solved using a minimum of 3 frames ! That analysis does not take into account that some parameters *cannot* be estimated independently, that a sequence of p frames does not yield $p \times n \times m$ independent measurements in most cases and that, in practice, the ratio of independent observables to unknown parameters is too small to produce a meaningful, let alone robust, solution. Specifically, only the direction of the translation vector can be determined since $\mathbf{t}$ only appears as $\mathbf{t}/Z$ in the FICE, or to put

[2] For a distant source like the sun, only the direction $\hat{\mathbf{l}}$ (2 unknowns) and intensity L_0 of the source are relevant. For a nearby source, the full position $\mathbf{l}$ of the source and its intensity are required.

it another way, $\mathbf{t}$ can only be determined within a global scaling factor. Similarly, the light source position $\mathbf{l}$, or direction $\hat{\mathbf{l}}$, and intensity L_0 cannot be recovered independently since they only appear as a product, $L_0\mathbf{l}$ or $L_0\hat{\mathbf{l}}$, in the FICE and SE. Consequently, the intensity of the source will be set to unity.

In order to build a robust solution, all the measurements, i.e. the full image σ, are used in the minimization and the number of unknowns is reduced by assuming a parametric surface patch, specified by a few parameters. However, parametric surface models not only reduce the number of parameters but can also reduce drastically the number of independent measurements. In some cases, for specific combinations of surface illumination, source models and geometric surface shape, the number of independent measurements is insufficient and the problem becomes underconstrained. Specific cases will be discussed in detail in chapter 4 for planar patches and in chapter 5 for quadratic patches.

2.3.1.2 General Minimization Equations

Given the irradiance values and the two constraint equations FICE and SE, we can either (a) minimize the square of the error in the FICE alone, (b) perform (a) subject to the SE constraint, (c) minimize a weighted sum of the square of the error in the FICE and the square of the error in the SE, or (d) minimize the square of the error in the SE alone. Scheme (a) is the traditional least–squares formulation, as used by Negahdaripour (1986) in the case where $\dot{E} = 0$, and is only applicable if the changes in illumination are negligible and the spatiotemporal variations are purely due to the texture. Algorithm (b) represents the general least–squares formulation that allows the determination of the motion and structure parameters for a patch with arbitrary smoothly varying albedo. The problem is set up as a constraint minimization where the unknown albedo values are determined by the shading constraint that relates directly the irradiance values on σ to the albedo values on Σ. The constrained minimization (b) is turned into an unconstrained minimization with the introduction of a stabilizer functional derived from the shading equation. Scheme (c) is the "regularization" formulation of algorithm (a). Scheme (d) only recovers the normal of the surface and ignores the motion information and will not be further discussed[3]. Each of the previous minimizations can be performed, when appropriate,

[3]Brooks and Horn (1985) used this approach to recover shape and source information from shading.

with added parameter constraints like $\|\hat{\mathbf{n}}\|^2 = 1$ or $\|\hat{\mathbf{L}}\|^2 = 1$. These additional constraints are required if unit vectors like the normal $\hat{\mathbf{n}}$ or the source direction $\hat{\mathbf{L}}$ are estimated. In general, the FICE depends both on the normal and the unit normal and either can be recovered with an unconstrained minimization for $\mathbf{n}$ or an equivalent constrained minimization for $\hat{\mathbf{n}}$. Once $\mathbf{n}$ or $\hat{\mathbf{n}}$ is determined, the other quantity is readily available since $\hat{\mathbf{n}} = \dfrac{\mathbf{n}}{|\mathbf{n}\|}$ and $\mathbf{n} = \dfrac{\hat{\mathbf{n}}}{\hat{\mathbf{n}} \cdot \hat{\mathbf{z}}}$. However, in practice, the complexity of the algebra and the numerical implementation, and ultimately the numerical accuracy of the estimates, is highly dependent on the type of minimization performed.

More specifically, if we let

$$\begin{aligned} A &= E_t - (\mathbf{v} \cdot \boldsymbol{\omega}) - \frac{1}{Z}(\mathbf{s} \cdot \mathbf{t}) - \dot{E} \\ \widetilde{A} &= \iint_\sigma A^2 d\mathbf{r} \\ B &= E(\mathbf{x}, t) - \rho_\lambda(\alpha, \beta) e(\hat{\mathbf{L}}, \hat{\mathbf{n}}) \\ \widetilde{B} &= \iint_\sigma B^2 d\mathbf{r} \\ C &= \|\hat{\mathbf{n}}\|^2 - 1 \text{ and } D = \|\hat{\mathbf{L}}\|^2 - 1 \end{aligned}$$

the following unconstrained and constrained minimizations can be performed :

$$(a) \quad \min_\sigma \widetilde{A}$$

$$(b) \quad \begin{cases} \min_\sigma \widetilde{A} \\ \text{subject to } B = 0, \ \forall \mathbf{r} \in \sigma \end{cases} \iff \min_\sigma \left(\widetilde{A} + \iint_\sigma \mu(\mathbf{r}) B d\mathbf{r} \right)$$

$$(c) \quad \min_\sigma (\widetilde{A} + \alpha \widetilde{B}), \quad \alpha \text{ weighting constant}$$

as well as the corresponding constrained minimizations that include additional parameter constraints like C and D,

$$(a') \quad \begin{cases} \min_\sigma \widetilde{A} \\ \text{subject to } C = 0, \ D = 0 \end{cases} \iff \min_\sigma (\widetilde{A} + \lambda C + \nu D)$$

$$(b') \quad \begin{cases} \min_\sigma \widetilde{A} \\ \text{subject to } B = 0, \ \forall \mathbf{r} \in \sigma \\ \text{subject to } C = 0, \ D = 0 \end{cases} \iff \min_\sigma \left(\widetilde{A} + \iint_\sigma \mu(\mathbf{r}) B d\mathbf{r} + \lambda C + \nu D \right) \quad (2.24)$$

where λ and ν are Lagrange multipliers and $\mu(\mathbf{r})$ is a Lagrange multiplier function.

The minimization equations developed in this section use a generic expression for the reciprocal of the depth, $1/Z = \zeta(\hat{\mathbf{n}}, *)$, and a generic shading model, $E(\mathbf{r}, t) = \rho_\lambda(\alpha, \beta) e(\hat{\mathbf{L}}, \hat{\mathbf{n}})$ and its associated temporal derivative $\dot{E} = dE(\mathbf{x}, t)/dt = \rho_\lambda(\alpha, \beta) e_t(\hat{\mathbf{L}}, \hat{\mathbf{n}}, \boldsymbol{\omega}, \mathbf{t})$. In general, shading depends on the source direction $\hat{\mathbf{L}}$, and therefore $\hat{\mathbf{I}}$ and $\mathbf{R}$, the unit surface normal $\hat{\mathbf{n}}$ and the surface albedo $\rho_\lambda(\alpha, \beta)$, assumed to be a smooth functional of the surface coordinates. The temporal derivative of the shading equation additionally depends on the motion vectors $\boldsymbol{\omega}$ and $\mathbf{t}$. For the parametrized patches considered in this work, $1/Z$ can always be expressed in terms of the normal $\mathbf{n}$ at a specific point and other optional variables represented symbolically by $*$. For example, $\mathbf{n}$ is the unique normal and $*$ is nil for a planar patch, while $\mathbf{n}$ is the normal at the point of expansion of the Taylor series for the patch, and $*$ represents the curvatures at the same point for a quadratic patch.

The next section presents a symbolic solution to the minimization problem b' where the light source direction $\hat{\mathbf{L}}$ is known. This generic solution outlines the general procedure used to solve such a minimization problem and gives an idea of the complexity of the resulting equations. Specific minimizations, with explicit structure $(1/Z)$ and shading model, and their numerical implementations, will be presented in chapter 4 for planar patches, and in chapter 5 for quadratic patches.

2.3.1.3 Generic Solution to a Constrained Minimization Problem

Let us consider the minimization problem b' with the only parameter constraint $C = 0$ and let us assume that the reciprocal of the depth $1/Z$ is only dependent on $\mathbf{n}$, i.e. $1/Z = \zeta(\mathbf{n})$. If we let $G(\boldsymbol{\omega}, \mathbf{t}, \mathbf{n}) = E_t - (\mathbf{v} \cdot \boldsymbol{\omega}) - \zeta(\mathbf{n})(\mathbf{s} \cdot \mathbf{t})$, b' can be written in the expanded form

$$\mathcal{E} = \min_{\sigma} \iint_{\sigma} \Big(\Big(G(\boldsymbol{\omega}, \mathbf{t}, \mathbf{n}) - \rho_\lambda e_t(\hat{\mathbf{L}}, \hat{\mathbf{n}}, \boldsymbol{\omega}, \mathbf{t}) \Big)^2 + \mu(\mathbf{r}) \Big(E(\mathbf{r}, t) - \rho_\lambda e(\hat{\mathbf{L}}, \hat{\mathbf{n}}) \Big) \Big) d\mathbf{r} + \lambda \big(\|\hat{\mathbf{n}}\|^2 - 1 \big) \tag{2.25}$$

For an extremum of $\mathcal{E}$ we must have

$$\frac{\partial \mathcal{E}}{\partial \boldsymbol{\omega}} = 0, \quad \frac{\partial \mathcal{E}}{\partial \mathbf{t}} = 0, \quad \text{and} \quad \frac{\partial \mathcal{E}}{\partial \hat{\mathbf{n}}} = 0$$

that result in the three vectorial equations[4]

$$\iint_\sigma \left(\mathbf{v} + \rho_\lambda \frac{\partial e_t}{\partial \omega}\right)\Big(G(\omega, \mathbf{t}, \mathbf{n}) - \rho_\lambda e\Big)d\mathbf{r} = 0 \quad (2.26)$$

$$\iint_\sigma \left(\zeta(\mathbf{n})\mathbf{s} + \rho_\lambda \frac{\partial e_t}{\partial \mathbf{t}}\right)\Big(G(\omega, \mathbf{t}, \mathbf{n}) - \rho_\lambda e\Big)d\mathbf{r} = 0 \quad (2.27)$$

$$\iint_\sigma \left(\frac{\mathbf{s}\cdot\mathbf{t}}{\hat{\mathbf{n}}\cdot\hat{\mathbf{z}}}\left(\mathbf{I}_3 - \frac{\hat{\mathbf{z}}\hat{\mathbf{n}}^T}{\hat{\mathbf{n}}\cdot\hat{\mathbf{z}}}\right)\frac{\partial \zeta}{\partial \hat{\mathbf{n}}} + \rho_\lambda \frac{\partial e_t}{\partial \hat{\mathbf{n}}}\right)\Big(G(\omega, \mathbf{t}, \mathbf{n}) - \rho_\lambda e\Big)d\mathbf{r} + \mu(\mathbf{r})\rho_\lambda \frac{\partial e}{\partial \hat{\mathbf{n}}} - \lambda\hat{\mathbf{n}} = 0 \quad (2.28)$$

where $\mathbf{I}_3$ represents the 3×3 identity matrix.

The Lagrange multiplier λ and the Lagrange multiplier functional $\mu(\mathbf{r})$ can be eliminated from (2.28) by taking the dot product of the vectorial equation with two vectors (e.g. $\hat{\mathbf{n}}$ and $\hat{\mathbf{L}}$) and by solving the resulting linear scalar system in the unknowns λ and $\iint_\sigma \mu(\mathbf{r})$. The choice of the vectors, used in the previous dot product, is shading and structure dependent and produces a more or less complex linear system. Once the system is solved, the solution is plugged back into (2.28) and we are left with a nonlinear system of three vectorial equations in the unknowns $\omega, \mathbf{t}$ and $\hat{\mathbf{n}}$. The specific degree of nonlinearity of each of these variables depends on the models used but, in general, all the variables will be at least quadratic and only a global, nonlinear method can be used to solve the system. In some cases, with some simple shading models, the system of equations is partially linear in some of the variables and iterative mixed (linear and nonlinear) methods can be used in these instances.

In all cases studied, the system of vectorial equations is too complicated for a direct vectorial solution. The vectorial system is transformed into a scalar system by projecting the vectorial equations onto suitable axes. In many instances, the best projection directions are the orthogonal unit vectors of the camera frame of reference $(\hat{\mathbf{x}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$. However, substantial algebraic simplifications can be achieved with some shading and structure models by using privileged directions. In particular, some of the vectorial equations $\frac{\partial \mathcal{E}}{\partial \hat{\mathbf{n}}} = 0, \frac{\partial \mathcal{E}}{\partial \mathbf{n}} = 0$ and $\frac{\partial \mathcal{E}}{\partial \hat{\mathbf{L}}} = 0$ contain rank deficient matrices whose eigenvalues are easy to compute, and projecting in the direction of the null space of the matrix and its orthogonal direction, results in significant algebraic simplifications. Once the vectorial equations are projected, the system consists of a rational or polynomial system of equations in the unknowns $\omega_x, \omega_y, \omega_z, t_x, t_y, t_z, n_x, n_y$ and n_z, for a planar patch for example. This system $\mathcal{S}$ can be represented globally, and more generally, by the

[4]For clarity's sake, the explicit dependency of some of the functionals on the various variables will be dropped when no confusion is possible.

vectorial equation $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ where $\mathbf{f} = (f_1 ... f_n)^T$ is the vector of nonlinear functionals in the unknowns $\mathbf{x} = (x_1 ... x_n)^T \in \mathrm{R}^n$. In the next section we review some of the methods available to solve nonlinear systems and discuss their drawbacks.

2.3.2 Discussion on Numerical Methods for Nonlinear Systems

The general characteristic of all available methods for solving nonlinear system of equations, or a single nonlinear equation for that matter, is the recursive nature of the solution and the lack of guaranteed global convergence. The convergence properties vary from algorithm to algorithm. Very often the radius of convergence is fairly small and these algorithms perform poorly for rough surfaces and fail to converge to the global minimum unless the initial estimate is close enough to the solution. This problem is exacerbated by the high dimensionality of the parameter space and the degree of nonlinearity.

Two main classes of methods are used in solving such systems: a minimization approach where the system is solved by least–squares minimization, $\min g(\mathbf{x}) = \sum_{i=1}^{n} f_i^2(\mathbf{x})$, $\mathbf{x} \in \mathrm{R}^n$ and a direct approach where the system is solved recursively while keeping the structure of the equations intact.

2.3.2.1 Minimization Techniques

Minimization techniques can be decomposed into several classes of methods: descent methods, conjugate–direction methods and Gauss–Newton type methods. Descent methods for $g(\mathbf{x})$ are algorithms for which the iteration decreases the function value at each step i.e.

$$g(\mathbf{x}^{(k+1)}) \leq g(\mathbf{x}^{(k)}) \tag{2.29}$$

where $g(\mathbf{x}) : R^n \longrightarrow R$, is a real–valued function. The general form of the iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{C}_k^{-1} \nabla_{\mathbf{x}} g(\mathbf{x}^{(k)})$$

where $\mathbf{C}_k$ is a symmetric, positive definite matrix. A simple descent method is damped Newton's iteration where $\mathbf{C}_k$ is the Hessian of $g(\mathbf{x})$. The damping constant α_k varies the step length and is chosen in accordance with (2.29). A simple and widely implemented method is steepest descent where $\mathbf{C}_k^{-1}$ is the transpose of the Jacobian of g. Steepest descent chooses the direction

of maximal local decrease of g for each iteration. It breaks down if the initial estimate lies in a neighborhood of a nonzero minimum of g, and can be *very* slow. More specifically, in the quadratic case $g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} - \mathbf{x}^T\mathbf{b}$, the convergence rate depends on the value of the eigenvalues of the positive definite $n \times n$, matrix $\mathbf{Q}$. The convergence rate of steepest descent slows as the contours of $g(\mathbf{x})$ become more eccentric and conversely the algorithm converges in one step for circular contours. In particular, even if $n-1$ of the n eigenvalues are equal and the remaining one is a great distance from these, the convergence will be slow and hence a single abnormal eigenvalue can destroy the effectiveness of steepest descent. On the other hand, steepest descent performs nicely when only a poor initial approximation of the solution is available. Levenberg–Marquardt (More 1977) is a composite algorithm that includes a parameter controlling the form of the iteration. For high values of the parameter, the iteration is like a steepest descent step while for a zero value of the parameter, the method reduces to a Newton step. This algorithm provides the speed of the Newton's iteration while avoiding the unpredictability of Newton's in the large residual problem where a steepest descent iteration is more favorable. Another highly popular method that is much faster than steepest descent is the Davidon–Powell–Fletcher (Fletcher and Powell 1963), which uses a variable metric algorithm.

The second class of methods are the conjugate direction methods. They were initially developed for linear system of equations. For the minimization of a quadratic functional $g(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T + \mathbf{c}$, $\mathbf{A}$ symmetric, positive definite matrix, the basic iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{p}^{(k)}$$

where the $\mathbf{p}^{(k)}$ are $\mathbf{A}$–orthogonal and $\alpha_k = \left(\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\right)\mathbf{p}^{(k)} / \left(\mathbf{A}\mathbf{p}^{(k)}\right)^T \mathbf{p}^{(k)}$. The main advantage of this method is its speed of convergence. If the function g is quadratic, $\{\mathbf{x}^*\}$ converge to the unique minimizer of g in at most n steps, where n is the number of equations.

The third basic method relies on the Gauss–Newton iteration,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}^{(k)}$$

where $\mathbf{s}^{(k)}$ is defined by the equation

$$\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{J}(\mathbf{x}^{(k)})\mathbf{s}^{(k)} = -\mathbf{J}(\mathbf{x}^{(k)})^T\mathbf{f}(\mathbf{x}^{(k)})$$

and $\mathbf{J}(\mathbf{x}^{(k)}) = \mathbf{f}'(\mathbf{x}^{(k)})$ is the Jacobian of $\mathbf{f}$ at $\mathbf{x}^{(k)}$. The main difference between it and Newton's is that the Newton model is based on the assumption that g can be adequately modeled by

a quadratic function, while Gauss–Newton results from the stronger assumption that $\mathbf{f}$ can adequately be modeled by an affine function. Augmented Gauss–Newton methods exist in which part of the Hessian matrix is computed exactly and part is approximated by a secant in a fashion analogous to the Davidon–Fletcher–Powell algorithm.

In our experiments, we implemented a modified Levenberg-Marquardt method (More 1977) when casting the equations into a minimization problem. This algorithm is reasonably fast, has nice convergence properties while avoiding the implementation complexity and slowness of the conjugate direction algorithm.

2.3.2.2 Direct Method for Solving Nonlinear Systems

Two radically different classes of methods are discussed. The first class of algorithms is based on the Newton–Raphson iteration and is closely related to the minimization methods described in the previous section. Not too surprisingly, the algorithms' performances are very similar, in terms of speed and region of convergence, to their minimization counterpart. The second class of algorithms is composed of algorithms based on homotopy methods.

The basic iteration is the Newton–Raphson, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta^{(k)}$ where the direction $\delta^{(k)}$ is defined by the linear equations $\mathbf{f}(\mathbf{x}^{(k)}) + \mathbf{J}\delta^{(k)} = 0$ where $\mathbf{J}$ represents the Jacobian matrix of the system of equations $\mathbf{f}$. A common modification is to retain the direction given by $\delta^{(k)}$ but restrict the length with the introduction of a parameter $\lambda^{(k)}$, $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \lambda^{(k)}\delta^{(k)}$. The parameter is chosen in a way that decreases the value of g at each iteration (descent method). The problem with this basic iteration is that the solution may converge to a point that is not a stationary point of $g(\mathbf{x})$. The basic iteration can be modified and improved with the introduction of a *switch* parameter μ. For $\mu = 0$, the classical iteration is unchanged while the direction $\delta^{(k)}$ tends to the steepest descent direction for large values of μ. A good implementation of this type of method is Powell's hybrid method (Powell 1970). In essence the hybrid method is very close to Levenberg–Marquardt but it does not require the explicit expression of the derivatives and uses successive values of $f_i(\mathbf{x}^{(k)}$ to build up a numerical approximation to the derivatives. The method does not impose the reduction of the sum of the squares at each iteration because this technique depends on the scaling of g and might cause convergence to a point at which the equations are not satisfied.

We implemented this method and observed that the results were very similar to the modified Levenberg–Marquardt minimization method in terms of speed and convergence but turned out to be slightly more flexible when dealing with mixed systems of linear and nonlinear equations.

2.3.2.3 Homotopy Methods

Homotopy methods, also known as continuation methods or methods of incremental loading, rely on a totally different mechanism to solve nonlinear equations. They construct a continuous map Φ (the homotopy) from a simple function $\mathbf{s}(\mathbf{x})$, with known zeros, to the function $\mathbf{f}(\mathbf{x})$ with the unknown zeros. More formally, let B be a Banach space and let $\mathbf{s}$ and $\mathbf{f}$ be defined on B. The homotopy map is defined by

$$\begin{aligned} \Phi: \quad \mathrm{B} \times [0,1] &\longrightarrow \mathrm{B} \\ (\mathbf{x}, \lambda) &\longmapsto \Phi(\mathbf{x}, \lambda) \end{aligned}$$

such that $\Phi(\mathbf{x}, 0) = \mathbf{s}(\mathbf{x})$, $\Phi(\mathbf{x}, 1) = \mathbf{f}(\mathbf{x})$. Then, by solving the equation $\Phi(\mathbf{x}, \lambda) = 0$ in $\mathrm{B} \times [0,1]$, one attempts to move from the known zero of $\mathbf{s}(\mathbf{x})$ (at $\lambda = 0$) to the unknown zero of $\mathbf{f}(\mathbf{x})$ (at $\lambda = 1$). There is a considerable amount of theory concerning when this procedure will work (Ortega and Rheinboldt 1970) but in general, moving from a zero of $\mathbf{s}(\mathbf{x})$ to a zero of $\mathbf{f}(\mathbf{x})$ may or may not be possible. For the calculation of fixed points, the supporting theory is much more satisfactory. Chow (Chow et al. 1978) have proved that computing a fixed point of $\mathbf{f}$ merely amounts to tracking a smooth zero curve of $\rho_{\mathbf{a}}(\lambda, \mathbf{x}) = \lambda(\mathbf{x} - \mathbf{f}(\mathbf{x})) + (1 - \lambda)(\mathbf{x} - \mathbf{a})$, where the index $\mathbf{a}$ represents the start of the curve, and global convergence is guaranteed with probability one. The same algorithm can be used to find zeros also, but then the global convergence is not guaranteed. The homotopy map for zeros is $\rho_{\mathbf{a}}(\lambda, \mathbf{x}) = \lambda\mathbf{f}(\mathbf{x}) + (1 - \lambda)(\mathbf{x} - \mathbf{a})$.

The earliest implementation of a similar, simplified, algorithm was done by Deist and Sefor (1967) but the idea underlying the method was first described by Davidĕko (1953). The success of their method largely depends on the cleverness of the implementator: given a set of nonlinear equations, one has to modify it to a form that can be handled analytically by introducing a set of parameters. These simpler equations, of which a solution can be found, are then changed to their original form while simultaneously tracing the roots. The solution amounts to the solution of a first–order system of ordinary differential equation. Their approach is less systematic than the homotopy mapping and only really works well for systems of equations that can be simplified

by a single parameter.

Our implementation is based on the homotopy algorithm described by Watson and Fenner (1980). It produces nice results in some cases but at a very high computational cost. The algorithm needs to perform fairly precise numerical integration in the root tracing process.

2.3.3 Dynamical Frame Unwarping Incremental FICE

The full brightness equation (2.21), developed in the previous section is afflicted by several major problems such as lack of flexibility and extendibility to multiple frames and poor performance for widely separated frames. The FICE was derived in the two–frame case and is based on the *differential* motion occurring between these two frames. However, in practice, there is no control on the frame rate and the resulting motions are far from being infinitesimally small. In addition, we want to compute the motion parameters at any instant in time (not necessarily aligned with a given input frame) and we want to easily extend the constraint equation to multiple frames. The first issue can be addressed by the use of multiresolution methods where the brightness images are filtered and spatially sampled to contract the motion. The second problem can, apparently, easily be solved by brightness values and gradient temporal interpolation[5] although a constraint equation formulation with a flexible virtual reference frame is easier to understand. The last and most important problem, the use of multiple frames in conjunction with the FICE, is unsatisfactory with the previous FICE, as shown in chapter 6, and a more sophisticated constraint equation is required. Most of the difficulties outlined here can be resolved with the use of a Dynamical Frame Unwarping (DFU) type of constraint equation, which is presented next.

The DFU scheme makes use of a displaced frame difference (DFD) type of quantity (such a DFD was introduced in section 2.2.3.2) in conjunction with an incremental computation of the motion parameters. In the incremental scheme, only the difference between the current estimates of the motion parameters and the true parameter is computed. Equation 2.15 is an example of such a formulation in the special case of optical flow computation from two known frames at time t and $t+T$, under the assumption of brightness constancy. The proposed DFU is best described in the context of optical flow recovery at time t from two successive arbitrary

[5]The general subject of interpolation and surface fitting will be discussed in section 2.5.1.

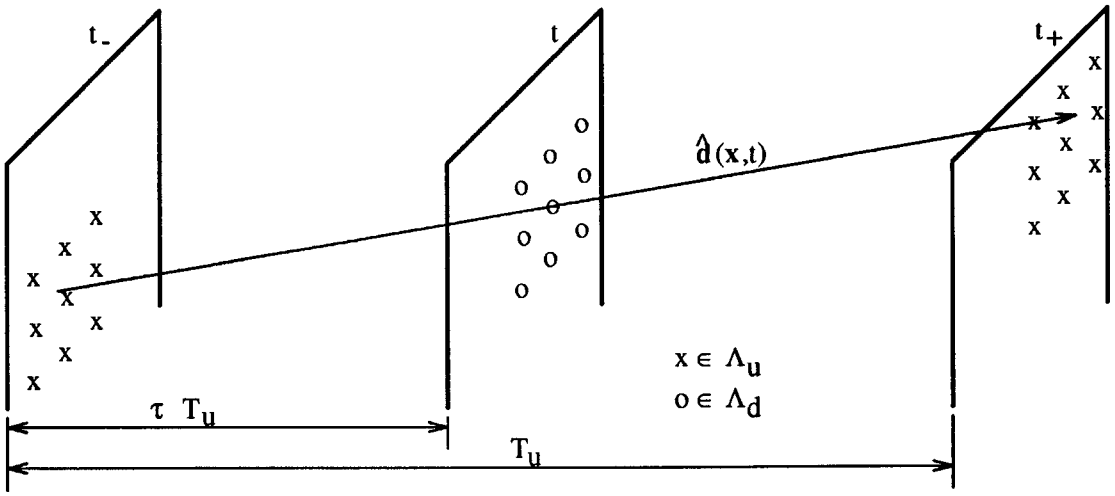

Figure 2.6: Geometry of velocity field frame with respect to the preceding and following brightness frame

frames at times t_1 and $t_1 + \Delta T$. The shading variation effects due to surface motion will be discussed throughout the presentation of the optical flow model, while the specialization of the equations to rigid body motion will be treated last.

Let Λ_u be a sampling lattice characterized by the sampling vector $\left(T_u^h, T_u^v, T_u^t\right)^T$ and associated with the given brightness frame $u(\mathbf{x}, t)$ and let $\Lambda_\mathbf{r}$ be characterized by the vector $\left(T_\mathbf{r}^h, T_\mathbf{r}^v, T_\mathbf{r}^t\right)^T$ the sampling lattice associated with the estimated velocity field frame. Let $\tau = \frac{t}{T_\mathbf{r}^t} - \lfloor \frac{t}{T_\mathbf{r}^t} \rfloor$ be the normalized (with respect to the interframe distance $T_\mathbf{r}^t$) temporal distance between the motion field, at time t, and the preceding image (see Figure 2.6). With the definition of τ, the preceding image is indexed in time by $t_- = t - \tau T_\mathbf{r}^t$ while the following image is indexed by $t_+ = t - (1-\tau)T_\mathbf{r}^t$.

The motion field $\hat{\mathbf{d}}_t$, estimated from the image fields $u(\mathbf{x}, t_+)$ and $u(\mathbf{x}, t_-)$, is computed at every point $(\mathbf{x}_i, t)$ of the $\Lambda_\mathbf{r}$ lattice. Assuming linear motion trajectories, the 2–D vector field of displacement is defined by $\hat{\mathbf{d}}_t = \{\hat{\mathbf{d}}_t(\mathbf{x}_i, t), \quad (\mathbf{x}_i, t) \in \Lambda_\mathbf{r}\}$ and is such that $\forall (\mathbf{x}_i, t) \in \Lambda_\mathbf{r}$, $\hat{\mathbf{d}}_t(\mathbf{x}_i, t)$ is the displacement between the point $(\mathbf{x}_i - \tau \hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_-)$ in the preceding frame and the point $(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_+)$ in the following frame. The displaced frame difference $\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}_t(\mathbf{x}_i, t))$ is defined by

$$\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}_t) = \tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_+) - \tilde{u}(\mathbf{x}_i - \tau \hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_-)$$

where $\tilde{u}(\mathbf{x}_i, t_k)$ denotes the brightness value at site $(\mathbf{x}_i, t_k) \notin \Lambda_u$ retrieved by spatial interpola-

tion. The DFD represents the unwarped difference between the preceding and following fields. Conceptually, the preceding and following fields are realigned so that every point in the same spatial location matches. If the unwarping is perfect, the DFD only represents the variation in shading due to the change in orientation of the 3–D point $\mathbf{X}_i$ that projects *identically* into $\mathbf{x}_i$ in the frames t_- and t_+. Under the constant shading assumption, the DFD is identically zero for perfect unwarping.

Let $\hat{\mathbf{d}}^{(n)}(\mathbf{x}_i, t)$ be an estimate of the displacement field at the n^{th} iteration. Using a first order Taylor expansion of $\tilde{u}(\mathbf{x}_i, t)$, $\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}_t)$ can be approximated by the known $\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)})$ and the spatial derivatives of $\tilde{u}(\mathbf{x}_i, t)$. In fact,

$$\begin{aligned}
\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}_t) &= \tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)} - (1-\tau)(\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t), t_+) - \\
&\qquad \tilde{u}(\mathbf{x}_i - \tau\hat{\mathbf{d}}^{(n)} + \tau(\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t), t_-) \\
&= \mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t)\Big(\tau\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i - \tau\hat{\mathbf{d}}^{(n)}, t_-) + \\
&\qquad (1-\tau)\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+)\Big) + \mathcal{O}(\|\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t\|).
\end{aligned}$$

At the optimum, i.e. when $\hat{\mathbf{d}}_t = \mathbf{d}_t$ the true motion field, the DFD—$\mathbf{D}(\mathbf{x}_i, t, \tau, \mathbf{d}_t)$–is zero under the brightness constancy assumption and is equal to $\dot{E}$, the temporal variation of shading, in the general case. In the optical flow case, the DFU full irradiance constraint equation takes the global form,

$$\mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t)\Big(\tau\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i - \tau\hat{\mathbf{d}}^{(n)}, t_-) + (1-\tau)\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+)\Big) = \dot{E}. \quad (2.30)$$

Equation 2.30 can be rewritten in the form

$$\begin{aligned}
\dot{E} &= \mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t)\Bigg(\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+) + \\
&\qquad \tau\Big(\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i - \tau\hat{\mathbf{d}}^{(n)}, t_-) - \nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+)\Big)\Bigg) \\
&= \mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t)\left(\mathbf{I}_3 - \tau\frac{\partial\hat{\mathbf{d}}^{(n)}}{\partial\mathbf{x}_i}\right)\Big(\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+)\Big) \\
&\simeq \mathbf{D}(\mathbf{x}_i, t, \tau, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t)\Big(\nabla_{\mathbf{x}}\tilde{u}(\mathbf{x}_i + (1-\tau)\hat{\mathbf{d}}^{(n)}, t_+)\Big).
\end{aligned} \quad (2.31)$$

The latter approximation is valid for a motion field $\hat{\mathbf{d}}_t = (\hat{dx}_t, \hat{dy}_t)$ that is either slowly varying, i.e. $\|\nabla\hat{dx}_t\|$ and $\|\nabla\hat{dy}_t\|$ are small, or locally translational or constant. The relationship between gradients and Hessians of corresponding points is presented in appendix B.

The preceding development makes it quite apparent that the concept of a dynamically unwarped constraint equation is not limited to two frames. Under the assumption of constant motion (or linear trajectories in the optical flow case), it is fairly obvious how to define a DFD with as many unwarped frames as wanted. Once unwarped by their respective displacement with respect to a central, possibly virtual, frame, all the frames are spatially aligned and can be stacked without any problem. If the alignment is perfect, the only irradiance difference between each unwarped frame is due to the underlying changes in surface orientation that can readily be computed from the rigid body motion parameters. The multiframe DFU constraint equation is developed in detail in chapter 6.

On the one hand, the DFU concept is very attractive because of it extendibility and flexibility as well as its increasing accuracy since, at every step, the quantities are smaller and the DFU equation is a better approximation to the original differential equation. On the other hand, the computing burden is far higher since, at every iteration, gradients and brightness values at different locations not belonging to the image sampling lattice need to be computed. Section 2.5 will describe how such computations can be achieved in an efficient manner.

Having derived a DFU constraint equation for the optical flow case, our attention will be restricted to the direct motion estimation and similar results derived in the rigid body motion case.

Let $\mathbf{s}(\mathbf{r}, \mathbf{d}_t, t) = (\nabla_{\mathbf{x}} u(\mathbf{x} + \mathbf{d}_t, t) \times \hat{\mathbf{z}}) \times \mathbf{r}$ and $\mathbf{v}(\mathbf{r}, \mathbf{d}_t, t) = \mathbf{s}(\mathbf{r}, \mathbf{d}_t, t) \times \mathbf{r}$ and let $(\boldsymbol{\omega}, \mathbf{t})$ represent the true motion parameters, $(\hat{\boldsymbol{\omega}}, \hat{\mathbf{t}})$ the estimated parameters and $(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)})$ the estimated parameters at the n^{th} iteration. With this notation, the dynamical frame unwarping constraint equation (2.30) can be written, for the rigid body motion, in the form

$$\begin{aligned} \mathbf{D}(\mathbf{x}_i, t, \tau, \mathbf{d}^{(n)}) + \Big(\tau \mathbf{v}(\mathbf{x}_i, \tau \mathbf{d}^{(n)}, t_-) + (1-\tau)\mathbf{v}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+)\Big)(\boldsymbol{\omega}^{(n)} - \boldsymbol{\omega}) + \\ \frac{1}{Z}\Big(\tau \mathbf{s}(\mathbf{x}_i, \tau \mathbf{d}^{(n)}, t_-) + (1-\tau)\mathbf{s}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+)\Big)(\mathbf{t}^{(n)} - \mathbf{t}) = \dot{E} \end{aligned} \tag{2.32}$$

and can be approximated, if the field is either locally constant, translational or slowly varying, by the simpler expression

$$\dot{E} = \mathbf{D}(\mathbf{x}_i, t, \tau, \mathbf{d}^{(n)}) + \mathbf{v}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+)(\boldsymbol{\omega}^{(n)} - \boldsymbol{\omega}) + \frac{1}{Z}\mathbf{s}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+)(\mathbf{t}^{(n)} - \mathbf{t}). \tag{2.33}$$

Both (2.32) and (2.33) require the computation of the $\mathbf{v}$ and $\mathbf{s}$ fields at each iteration since both fields depend on the current estimate of the displacement. The latter equation only

requires half as much work as the former equation in computing the fields $\mathbf{v}$ and $\mathbf{s}$ since they are only evaluated at time t_+. The efficiency in calculating both fields essentially depends on the efficiency in computing the unwarped spatial gradients $\nabla_{\mathbf{x}} u(\mathbf{x}, \tau \mathbf{d}^{(n)}, t_-)$ in frame t_- and $\nabla_{\mathbf{x}} u(\mathbf{x}, (1-\tau)\mathbf{d}^{(n)}, t_+)$ in frame t_+. Section 2.5 presents a fast method for computing the new unwarped gradients when the change in incremental displacement is moderate, that is $\|\mathbf{d}^{(n+1)} - \mathbf{d}^{(n)}\| \lesssim 1$ pixel.

2.4 First and Second Order Constraint Equations

In the previous sections, we derived constraint equations that were based on a first–order approximation of the irradiance function, or more specifically, only used spatio-temporal gradients of the irradiance. In this section, a second–order formulation that uses the gradients and Hessian of the irradiance function is presented and an approximation of the second order terms by a linear combination of the gradients of the preceding and following irradiance frames is used. The second–order formulation is first derived for the classical constraint equation case, i.e. under the assumption that the temporal variations of the shading dE/dt are zero and non-incremental displacements are computed. The new constraint equation is then extended to the incremental displacement computation by use of the DFD formulation. It should be noted that the equations initially presented in this section are only meaningful for irradiance functions continuous in *both* time and space. The situation is quite different once the equations are discretized on a grid. In particular, the choice of stencils or masks, for the computation of the spatiotemporal gradients truly determine the order of the discrete equations. This issue will be further discussed in section 2.4.3.

2.4.1 Classical Continuous Second–Order Constraint Equation

Let $u(\mathbf{r}, t)$ represents the irradiance at time t at the image plane location $\mathbf{r}$ and $u(\mathbf{r} + \mathbf{d}_t, t + dt)$ the irradiance at time $t + dt$ and spatial position $\mathbf{r} + \mathbf{d}_t$. Assuming that $\mathbf{d}_t$ is a first–order differential quantity, $u(\mathbf{r} + \mathbf{d}_t, t + dt)$ can be expanded into a second–order Taylor series around the point $(\mathbf{r}, t + dt)$

$$u(\mathbf{r} + \mathbf{d}_t, t + dt) = u(\mathbf{r}, t + dt) + \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt)\mathbf{d}_t + \frac{1}{2}\mathbf{d}_t^T \nabla\nabla_{\mathbf{r}} u(\mathbf{r}, t + dt)\mathbf{d}_t + \mathcal{O}(\|\mathbf{d}_t\|^2).$$

Assuming that $\mathbf{d}_t$ represents the displacement field between the frame t and $t + dt$, and that the temporal variations of shading are zero

$$\begin{aligned} u(\mathbf{r}, t) &= u(\mathbf{r} + \mathbf{d}_t, t + dt) \\ &= u(\mathbf{r}, t + dt) + \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt)\mathbf{d}_t + \mathcal{O}(\|\mathbf{d}_t\|). \end{aligned} \tag{2.34}$$

Taking the gradient of equation 2.34, we obtain

$$\begin{aligned} \nabla_{\mathbf{r}} u(\mathbf{r}, t) &= \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt) + \nabla\nabla_{\mathbf{r}} u(\mathbf{r}, t + dt)\mathbf{d}_t + \frac{\partial \hat{\mathbf{d}}^{(n)}}{\partial \mathbf{x}_i} \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt) + \mathcal{O}(\|\mathbf{d}_t\|^2) \\ &\simeq \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt) + \nabla\nabla_{\mathbf{r}} u(\mathbf{r}, t + dt). \end{aligned} \tag{2.35}$$

The latter approximation is valid for a motion field $\mathbf{d}_t$ that is either slowly varying or locally translational or constant. Combining equations 2.34 and 2.35 to eliminate the second–order terms and expanding $u(\mathbf{r}, t + dt)$ into a first order Taylor series in the variable t results in the constraint equation

$$\frac{\partial u(\mathbf{r}, t)}{\partial t} + \frac{1}{2}\Big(\nabla_{\mathbf{r}} u(\mathbf{r}, t) + \nabla_{\mathbf{r}} u(\mathbf{r}, t + dt)\Big)\frac{d\mathbf{d}_t}{dt} = 0. \tag{2.36}$$

Although the constraint equation 2.36 only requires the gradients of the irradiance in two frames, it implicitly contains the second–order variations of the irradiance function and is, therefore, more accurate and stable than the classical first–order constraint equation. Second–order variations are incorporated into the constraint equation by considering a smoothed version of the spatial gradients. The spatial gradients in the preceding or following frame, depending on the formulation, are replaced by an average of both spatial gradients. This formulation was first introduced by Bierling (1986) in the context of optical flow computation for TV imagery. This formulation is a special case of the previously presented unwarped equation where the constraint equation is evaluated at a virtual frame positioned in between two real frames. The DFU approach is more general because it computes the *unwarped* gradients at an arbitrary virtual frame. Equation 2.36 corresponds to evaluating the spatially corresponding gradients for a virtual frame situated exactly in the middle of two real frames, i.e. $\tau = 0$, $\hat{\mathbf{d}}^{(n)} = 0$ in equation 2.30 and the DFD replaced by the temporal gradient of the irradiance function. Bierling reported improved stability in displacement estimates using equation 2.36 instead of the classical first–order equation. Additional improvements were expected and measured using the unwarped gradients because gradients of *matching* neighborhoods are averaged, unlike Bierling's

formulation where gradients of potentially unrelated regions are averaged in the case of moderate to large displacement fields.

The parallel between the two formulations also makes apparent that the dynamical frame unwarping constraint equation captures the first and second order variations of irradiance function although it was derived without the explicit use of the Hessian of the irradiance function.

2.4.2 DFU Second–Order Constraint Equation

Bierling's method can be easily extended to the case of incremental motion by considering the second–order Taylor series development of $u(\mathbf{r} + \hat{\mathbf{d}}^{(n)}, t + dt)$ where $\hat{\mathbf{d}}^{(n)}$ represents an estimate of the displacement field at the n^{th} iteration. A development similar to the one in the previous section yields the incremental second–order constraint equation

$$\mathbf{D} + \frac{1}{2}\big(\nabla_{\mathbf{r}} u(\mathbf{r} + \hat{\mathbf{d}}^{(n)}, t + dt) + \nabla_{\mathbf{r}} u(\mathbf{r}, t)\big)(\mathbf{d} - \hat{\mathbf{d}}^{(n)}) = 0 \tag{2.37}$$

where the displaced frame difference is defined by $\mathbf{D} = u(\mathbf{r}+\hat{\mathbf{d}}^{(n)}, t+dt) - u(\mathbf{r}, t)$ and $\mathbf{d}$ represents the true displacement. Equation 2.37 is identical to (2.30) where the unwarped gradients are replaced by regular gradients.

2.4.3 Discretized Second–Order Constraint Equation

Section 2.4.1 presented a continuous second–order constraint equation that is approximated by first–order spatial gradients in two consecutive frames. This second–order equation 2.36 is a better approximation than the regular continuous first–order equation 2.10 because it takes into account both the gradient and Hessian of the irradiance data. However, when dealing with sampled data, these equations need to be discretized and the true order of the discrete equation depends on the type of stencils used in computing the spatiotemporal gradients. In particular, only a naive use of intra frame first difference or centered difference stencils will produce a different discrete version of equations 2.10 and 2.36.

Horn (1986) proposed a consistent way of estimating the spatiotemporal first–order derivatives by using first differences in a $2 \times 2 \times 2$ cube of irradiance value (see figure 2.7). If the indices i, j and k correspond to x, y and t, respectively, the three first partial derivatives of the

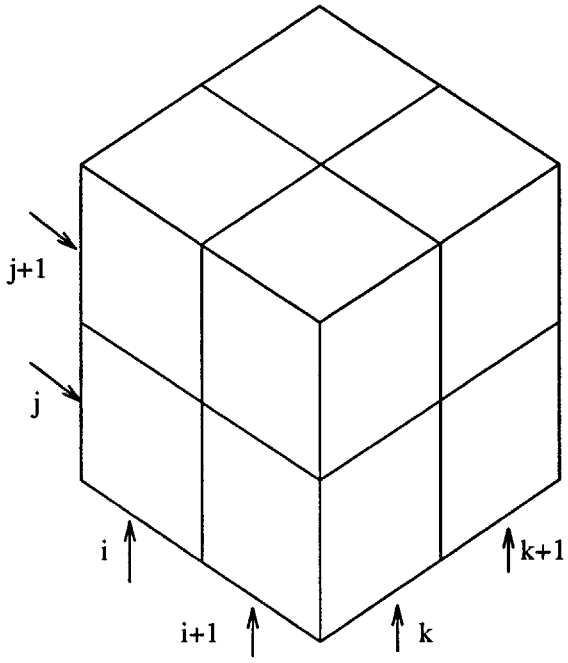

Figure 2.7: Cube of irradiance values for the estimation of the spatiotemporal gradients at the center pixel (from figure 12–7 of Horn (1986)).

irradiance are obtained by

$$
\begin{aligned}
E_x &= \frac{1}{2\delta x}\Big(\frac{1}{2}\Big((E_{i+1,j,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i,j+1,k})\Big) + \\
&\qquad \frac{1}{2}\Big((E_{i+1,j,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i,j+1,k+1})\Big)\Big) \\
E_y &= \frac{1}{2\delta y}\Big(\frac{1}{2}\Big((E_{i,j+1,k} - E_{i,j,k}) + (E_{i+1,j+1,k} - E_{i+1,j,k})\Big) + \\
&\qquad \frac{1}{2}\Big((E_{i,j+1,k+1} - E_{i,j,k+1}) + (E_{i+1,j+1,k+1} - E_{i+1,j,k+1})\Big)\Big) \\
E_t &= \frac{1}{4\delta t}\Big((E_{i,j,k+1} - E_{i,j,k}) + (E_{i+1,j,k+1} - E_{i+1,j,k}) + \\
&\qquad (E_{i,j+1,k+1} - E_{i,j+1,k}) + (E_{i+1,j+1,k+1} - E_{i+1,j+1,k+1})\Big).
\end{aligned}
$$

A lot of the problems in estimating consistent spatiotemporal gradients, and in particular temporal gradients, are suppressed by the use of the DFU constraint equation that is applied directly to the sampled data.

2.5 Spatiotemporal Derivatives Computation

A significant difficulty with differential methods is the requirement for accurate estimates of spatiotemporal irradiance gradients from the sampled irradiance sequences. We saw in section 2.2.3.3 that, if the original irradiance image is suitably bandlimited and noise free, an error–free reconstruction is possible by means of the interpolation formula

$$u(\mathbf{x},t) = \sum_{\mathbf{x}_i,\tau} u_s(\mathbf{x}_i,\tau)\Phi_i(\mathbf{x}-\mathbf{x}_i, t-\tau)$$

where the interpolation kernel $\Phi_i(\mathbf{x},t)$ is the product of three ideal low–pass filters $\phi(\alpha, i, T_u^\alpha)$ with impulse response defined by

$$\phi(\alpha, i, T_u^\alpha) = \text{sinc}\left(\frac{\pi}{T_u^\alpha}(\alpha - iT_u^\alpha)\right)$$

where T_u^α represents the sampling period in the α–dimension.

Once the continuous band–limited signal $u(\mathbf{x},t)$ has been reconstructed in terms of the discrete samples $u_s(\mathbf{x}_i,\tau)$, the components of the spatial gradient $\nabla_{\mathbf{r}} u(\mathbf{x},t)$ and temporal gradient $\dfrac{\partial u(\mathbf{x},t)}{\partial t}$ can be expressed by

$$\begin{cases} \nabla_x u(\mathbf{x},t) &= \displaystyle\sum_{i,\tau} u_s(\mathbf{x}_i,\tau)\frac{\partial\phi}{\partial x}(x,x_i,T_u^h)\phi(y,y_i,T_u^v)\phi(t,\tau\, T_u^t) \\ \nabla_y u(\mathbf{x},t) &= \displaystyle\sum_{i,\tau} u_s(\mathbf{x}_i,\tau)\phi(x,x_i,T_u^h)\frac{\partial\phi}{\partial y}(y,y_i,T_u^v)\phi(t,\tau\, T_u^t) \\ \dfrac{\partial u(\mathbf{x},t)}{\partial t} &= \displaystyle\sum_{i,\tau} u_s(\mathbf{x}_i,\tau)\phi(x,x_i,T_u^h)\phi(y,y_i,T_u^v)\frac{\partial\phi}{\partial t}(t,\tau,T_u^t) \end{cases}.$$

Unfortunately, real irradiance sequences are neither bandlimited nor noiseless. In particular, we showed in section 2.2.3.3 that, in practice, the signal is never sampled fast enough in the temporal direction, severe temporal aliasing always occurs and the temporal gradient estimates will be poor. On the other hand, temporal aliasing is usually not too visible in moving pictures[6], because of the motion blurs. One way of avoiding this problem is to reformulate the constraint equation in terms of a displaced frame difference to eliminate the need for temporal gradients computation.

[6] One notable exception is the highly visible phenomenon of the wagons wheels that are turning backwards in movies or on TV.

The problem of estimating reliable gradients from sampled noisy, possibly aliased data is far from being solved and many tradeoffs have to be considered before deciding on a method. A wide spread and computationally cheap technique is to approximate the gradients by finite differences. In that case, the computation of gradients is equivalent to convolving the irradiance images with a filter mask, or stencil, and can be implemented very efficiently since most of the stencils used have very small supports. Typically, stencil support for gradient computations range from 1 by 2 to 5 by 5.

An "alternate" solution, very much in favor in recent motion estimation work (Martinez and Lim 1986, Krause 1987), is to parameterize the irradiance function by fitting a surface to the data and computing the spatiotemporal gradients from the parametrized, continuous function. A close look at these two methods reveals that, in most cases, they are identical and that the more complex surface fitting method is only justified in particular cases. A detailed discussion of the topic will be offered in section 2.5.2.

2.5.1 Surface Parameterization

The problem of finding a local, continuous parameterization of the sampled, noisy irradiance function is equivalent to the problem of designing an interpolation filter that performs noise reduction during the reconstruction process. There is no optimal way of reconstructing a continuous signal from noisy, aliased observations. Instead, we will consider a specific set of N orthogonal interpolation functions Ψ_i that simplify the parameter estimation while offering a flexible and adequate fit to the irradiance function. The irradiance function $u(\mathbf{x},t)$ is approximated by the parametric signal $\tilde{u}(\mathbf{x},t)$ defined by

$$\begin{aligned} \tilde{u}(\mathbf{x},t) &= \sum_{i=1}^{N} u_s(\mathbf{x}_i,\tau)\Psi_i(\mathbf{x}-\mathbf{x}_i,t-\tau) \\ &= \mathbf{U}^T\mathbf{\Psi} \end{aligned} \tag{2.38}$$

irradiance samples. The function that is minimized is

$$\sum_{(\mathbf{x}_i,\tau)\in W_{\mathbf{x}}\times W_t} \Big(u_s(\mathbf{x}_i,\tau) - \mathbf{U}^T\Psi(\mathbf{x}_i,\tau)\Big)^2$$

where $W_{\mathbf{x}}$ and W_t respectively represent the spatial and temporal analysis window used in determining the signal coefficient vector $\mathbf{U}$. In order to obtain robust estimates of the coefficients, the analysis window is chosen large enough to overdetermine the fit, that is, to supply more observations than necessary to uniquely determine the coefficient vector $\mathbf{U}$. Overdetermined fitting essentially performs a low–pass filtering of the irradiance function and reduces the noise. The least–squares estimate is easily computed and is given by

$$\mathbf{U} = \left(\sum_{\mathbf{x}_i,\tau} \Psi\Psi^T\right)^{-1} \left(\sum_{\mathbf{x}_i,\tau} \Psi u_s(\mathbf{x}_i,\tau)\right) = \mathbf{A}^{-1}\mathbf{b}. \tag{2.39}$$

The matrix $\mathbf{A} = \sum \Psi\Psi^T$ is independent of the data and is constant if the current pixel, on which the window analysis is centered, is used as the origin. Therefore, the matrix $\mathbf{A}^{-1}$ can be computed off–line in advance and only the vector $\mathbf{b}$ needs to be computed on the fly. Efficient computations of $\mathbf{b}$ are possible depending on the choice of the basis functions.

A versatile and efficient set of basis functions is a set of orthogonal polynomials $x^i y^j t^t$. Polynomial curve fitting has been used both by Martinez (1986) and Krause (1987). The choice of the specific degrees of the polynomials is a tradeoff between computational complexity and accuracy in modeling the irradiance function. A high order spatial fit $x^i y^j$ allows very accurate modeling of the spatial variations of the irradiance function; a high order temporal model increases the flexibility in tracking the variations of the irradiance from frame to frame. Experimental evidence shows that a quadratic spatial fit, that is $\tilde{u}(\mathbf{x}) = u_0 + u_x x + u_y y + u_{xx}x^2 + u_{xy}xy + u_{yy}y^2$ and $\tilde{\Psi}^T = (1 \;\; x \;\; y \;\; x^2 \;\; xy \;\; y^2)$ for basis functions, is a reasonable compromise. A higher dimensional model is not advantageous because, in many cases, the data are noisy enough that a higher than quadratic fit produces worse results and a lower order fit (planar) is too crude to model the spatial irradiance variations. In the DFU formulation, irradiance frames are dynamically unwarped; the unwarped temporal slices, within the analysis window, become very similar as the estimates of the displacement are refined. In fact, at the optimum, the only temporal variations are caused by the temporal variations of the shading model. For this reason, the temporal variations of the model are chosen to be at most linear. Higher

than globally quadratic basis functions (tx^2, txy, ty^2) are discarded because no performance improvements are observed in keeping the full 12 basis functions given by $(\tilde{\Psi}, t\tilde{\Psi})$. As a result, the vector Ψ of orthogonal basis functions is chosen to be $\Psi^T = (1 \;\; x \;\; y \;\; t \;\; xt \;\; yt \;\; xy \;\; x^2 \;\; y^2)$.

With this choice of basis functions, the most efficient scheme for computing the vector $\mathbf{b}$ is to use a sliding–block technique and to shift the center of the spatiotemporal window to $(0,0,0)$. Components of $\mathbf{b}$ that are independent of x can be computed by sliding the window along the $\hat{\mathbf{x}}$–axis. At each step, the next value is obtained by subtracting out the effect of the first column at the left edge and adding the new column to the right of the analysis window. A similar method is used to compute the components of $\mathbf{b}$ independent of y by sliding the window along the $\hat{\mathbf{y}}$–axis and subtracting and adding top and bottom rows. In this fashion, only the b_{xy} component needs to be computed with a full summation at each point. Further computation savings are achieved by precomputing the components b_1, b_x, b_y, b_{xy}, b_{xx}, b_{yy} in each frame and only performing the temporal summation to obtain b_t, b_{xt}, b_{yt} on the fly after the frames are unwarped by the current displacement value. However, in practice, the frames are not unwarped but indices in the various frames of coefficients are shifted by the value of the displacement. The precomputation and storage of the primary components of $\mathbf{b}$ result in substantial saving in computation in the incremental DFU method where the gradients need to be evaluated many times at different offset locations during the iteration. The gradients are computed from the parametrized irradiance function by differentiation of equation 2.38. With the chosen basis function the gradients are simply expressed by

$$\begin{cases} \dfrac{\partial u}{\partial x} &= \mathbf{U}_x + 2x\mathbf{U}_{x^2} + y\mathbf{U}_{xy} + t\mathbf{U}_{xt} \\ \dfrac{\partial u}{\partial y} &= \mathbf{U}_y + 2y\mathbf{U}_{y^2} + x\mathbf{U}_{xy} + t\mathbf{U}_{yt} \\ \dfrac{\partial u}{\partial t} &= \mathbf{U}_t + x\mathbf{U}_{xt} + y\mathbf{U}_{yt} \end{cases} . \tag{2.40}$$

If the gradients are only required at the original irradiance sampling lattice points, the partial derivatives in equation 2.40 are evaluated at the spatial location $\mathbf{x} = (0,0)$,

$$\begin{cases} \dfrac{\partial u}{\partial x} &= \mathbf{U}_x + t\mathbf{U}_{xt} \\ \dfrac{\partial u}{\partial y} &= \mathbf{U}_y + t\mathbf{U}_{yt} \\ \dfrac{\partial u}{\partial t} &= \mathbf{U}_t \end{cases} .$$

In order to provide the fastest gradient evaluation for the usual case, i.e. computation of the gradient at the central pixel, midway between the two real frames, the parameterization coefficient vector $\mathbf{U}$ is temporally offset so that $t = 0$ corresponds to the temporal position midway between the two frames; the spatiotemporal gradients are given directly by the coefficients $(\mathbf{U}_x,\ \mathbf{U}_y,\ \mathbf{U}_t)$ of the parameterization coefficient vector.

The next section focuses on the relationship between surface fitting and the stencil method of computing derivatives.

2.5.2 Relationship Between Stencils and Surface Fitting

At first glance, it appears that the surface fitting method is a much more flexible and accurate method than the stencil method since a least–squares fit for *each* point of the irradiance field is computed and the resulting coefficients are used to compute the spatiotemporal gradients. However, a closer look reveals that the two methods produce exactly the same results in most situations at the expense of a far higher computational cost for the surface fitting technique. In many cases, only the gradients at locations falling on the sampling lattice grid and at a fixed temporal position are computed. For this case, at most five coefficients of the parameterization are used $(\mathbf{U}_x,\ \mathbf{U}_y,\ \mathbf{U}_t,\ \mathbf{U}_{xt},\ \mathbf{U}_{yt})$; the computation of the gradients is space invariant. Then, the computation of the gradients with the surface fitting method and with a fixed stencil method are identical. Equation 2.39, which computes the values of the coefficient vector $\mathbf{U}$ in terms of the basis functions and the sampled data, is in fact an equivalent mathematical expression of the convolution of the data with a *fixed* filter mask determined by the choice of the basis functions and the size and shape of the spatiotemporal analysis window. The matrix $\mathbf{A}^{-1}$ is fixed, space invariant and only depends on the basis functions and the size of the window[7], while the vector $\mathbf{b}$ both depends on the vector Ψ and on the data. Vector $\mathbf{b}$ can be rewritten in the more explicit form

$$\mathbf{b} = \mathbf{QD}$$

where $\mathbf{Q} = (\Psi(\mathbf{x}_1, t_1)\ \ \Psi(\mathbf{x}_2, t_1)\ \ \ldots\ \ \Psi(\mathbf{x}_n, t_n))$, is the matrix of the basis vector Ψ evaluated at each arbitrarily numbered point $(\mathbf{x}_i, t_i)$ of the analysis window in the local pixel–centered coordinate frame, and $\mathbf{D}^T = \Big(u(\mathbf{x}_1, t_1)\ \ u(\mathbf{x}_2, t_2)\ \ \ldots\ \ u(\mathbf{x}_n, t_n)\Big)$ represents the vector of

[7]This statement is only true when $\mathbf{A}$ is evaluated in a local coordinate system centered at the current pixel.

the values of the irradiance function evaluated at the same points. Appendix C symbolically computes the matrix **A** for the chosen basis functions for a general and symmetric window and gives the numerical values of the matrix and its inverse for 3×3 and 5×5 spatial windows.

2.5.2.1 Example of Stencils Generated by Surface Fitting

In this section, the results of appendix C are used to show what types of stencils are implicitly used in usual surface fitting configurations. $\mathbf{A}^{-1}_{\alpha,\beta}$ denotes the α,β element of the inverse of the **A** matrix where the elements are indexed by the basis functions (eg. $\mathbf{A}^{-1}_{xt,yt}$) and the vector $\mathbf{P}^{i,\tilde{\Psi}} = (\mathbf{P}_1^{i,\tilde{\Psi}} \;\; \mathbf{P}_x^{i,\tilde{\Psi}} \;\; \mathbf{P}_y^{i,\tilde{\Psi}} \;\; \mathbf{P}_{xx}^{i,\tilde{\Psi}} \;\; \mathbf{P}_{yy}^{i,\tilde{\Psi}} \;\; \mathbf{P}_{xy}^{i,\tilde{\Psi}})^T$ represents the spatial summation of the product of the $\tilde{\Psi}$ basis functions with the irradiance function within the spatial analysis window in frame i, that is $\mathbf{P}_1^{i,\tilde{\Psi}} = \sum_{W_\mathbf{x}} u(\mathbf{x},t_i)$, $\mathbf{P}_x^{i,\tilde{\Psi}} = \sum_{W_\mathbf{x}} x u(\mathbf{x},t_i)$ etc. With this notation, the vector **b** is expressed by

$$\left(\sum_t \mathbf{P}_1^{t,\tilde{\Psi}} \;\; \sum_t \mathbf{P}_x^{t,\tilde{\Psi}} \;\; \sum_t \mathbf{P}_y^{t,\tilde{\Psi}} \;\; \sum_t t\mathbf{P}_1^{t,\tilde{\Psi}} \;\; \sum_t \mathbf{P}_{xx}^{t,\tilde{\Psi}} \;\; \sum_t \mathbf{P}_{yy}^{t,\tilde{\Psi}} \;\; \sum_t \mathbf{P}_{xy}^{t,\tilde{\Psi}} \;\; \sum_t t\mathbf{P}_{xt}^{t,\tilde{\Psi}} \;\; \sum_t t\mathbf{P}_{yt}^{t,\tilde{\Psi}}\right)^T.$$

Let us examine the value of the temporal coefficient $\mathbf{U}_t$. For a symmetric, square window (that is, $W_x = [-k..0..k] = W_y$ and $W_t = [-l..0..l]$), $\mathbf{U}_t$ simplifies to

$$\mathbf{U}_t = \mathbf{A}^{-1}_{1,t}\left(\sum_t \mathbf{P}_1^{t,\tilde{\Psi}}\right) + \mathbf{A}^{-1}_{t,t}\left(\sum_t t\mathbf{P}_1^{t,\tilde{\Psi}}\right). \tag{2.41}$$

For a 2–frame temporal window ($t = 0$ and $t = 1$) and an $N \times N$–point spatial window, $\mathbf{A}^{-1}_{t,t} = -2\mathbf{A}^{-1}_{1,t} = \frac{-2}{N^2}$, $\sum_t \mathbf{P}_1^{t,\tilde{\Psi}} = \mathbf{P}_1^{0,\tilde{\Psi}} + \mathbf{P}_1^{1,\tilde{\Psi}}$, $\sum_t t\mathbf{P}_1^{t,\tilde{\Psi}} = \mathbf{P}_1^{1,\tilde{\Psi}}$ and equation 2.41 evaluates to

$$\begin{aligned}\mathbf{U}_t &= \frac{1}{N^2}\left(\mathbf{P}_1^{1,\tilde{\Psi}} - \mathbf{P}_1^{0,\tilde{\Psi}}\right)\\ &= \frac{1}{N^2}\left(\sum_{W_\mathbf{x}} u(\mathbf{x},1) - \sum_{W_\mathbf{x}} u(\mathbf{x},0)\right)\end{aligned}$$

i.e. the temporal gradient is computed by the average of the frame difference over the $N \times N$ window. Similarly, the spatial gradient in the first frame is expressed by

$$\nabla_\mathbf{x} u(\mathbf{x},t)^T = \frac{1}{2N}\left(\frac{1}{\sum_{W_\mathbf{x}} x^2}\sum_{W_\mathbf{x}} x, \frac{1}{\sum_{W_\mathbf{x}} y^2}\sum_{W_\mathbf{x}} y\right)$$

and represents the N line (row) average of the $N/2$ centered differences. For example, with a $3 \times 3 \times 2$ analysis window, the corresponding stencil is

$$\frac{1}{3} \times \frac{1}{2} \left(\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \right). \tag{2.42}$$

2.5.2.2 Advantages of Curve Fitting

In the previous section, we saw that surface fitting has no advantage over stencils if the only task is to compute the spatiotemporal gradients at lattice points; it is a waste of time to determine a local parameterization of the surface. The major advantages of the continuous parameterization of the irradiance function are the direct evaluation of the function and gradients at non–grid points without interpolation and the cheap computation of multiple irradiance values and gradients at neighboring locations from a single local parameterization. The latter feature is very important and desirable in the DFU method because it provides a fast way of evaluating multiple gradients at low cost within the inner incremental parameters computation loop. Once the iteration has reached the correct parameters within one pixel, the same local parametrization can be used to compute the gradients by equation 2.40 and convergence to the optimum parameters is very fast.

Finally, the next section explores the parallels and differences of surface fitting and direct interpolation of the irradiance function and its gradients.

2.5.3 Approximation versus Interpolation

The previous discussion highlighted the major advantage of the surface fitting method as opposed to the stencil technique: the determination of a continuous signal model that approximates the irradiance function. This property is essential in the DFU algorithm because most of the computations occur at neighboring, non–grid points and the continuous, local approximation of the irradiance field allows efficient irradiance and irradiance gradient determination. An alternate path in obtaining a continuous model of the irradiance function is to use spatial interpolation. One fundamental difference between interpolation and surface fitting, or approximation, is that in the former case the values of the interpolated function and of the

original function match at the sample data points (e.g. Lagrange interpolation polynomial). One drawback of interpolation is that, in the case of noisy data, it makes little sense to match the noisy values and it is preferable to have a global approximation of the data and interpolated points. The problem can be alleviated by first smoothing the data with a low–pass filter before interpolating. Low–pass filtering is implicit (see section 2.5.1) in the process of surface fitting. A more global way of looking at the problem is to realize that both interpolation and surface fitting are special cases of the linear filtering problem of noisy, sampled data.

Let us examine the properties of the linear (in the system sense) bilinear, biquadratic and bicubic interpolator that models the irradiance function by a first–, second– and third–degree surface. Since these operations are separable, only the 1–D case is discussed. Once the surface is modeled, the gradient estimates at arbitrary locations are inferred from the local irradiance model and, due to the linearity of the system, the spatial gradients are linear functions of derivative of the impulse response of the filter. The three linear (in the system sense) interpolators, linear, quadratic and cubic are Lagrange polynomial interpolators. Let $g(\mathbf{x}_i))_{i\in D}$ represent the values of the discrete function g for the points $\mathbf{x}_i$ belonging to the domain D. The Lagrange interpolator functions $\tilde{g}_{lin}(h)$, $\tilde{g}_{qua}(h)$ and $\tilde{g}_{cub}(h)$ are defined by

$$\begin{aligned}
\tilde{g}_{lin}(h) &= (1-h)g(x_k) + hg(x_{k+1}) \\
\tilde{g}_{qua}(h) &= \frac{(h-1)(h-2)}{2}g(x_k) - h(h-2)g(x_{k+1}) + \frac{h(h-1)}{2}g(x_{k+2}) \\
\tilde{g}_{cub}(h) &= -\frac{h(h-1)(h-2)}{6}g(x_k) + \frac{(h-1)(h+1)(h-2)}{2}g(x_{k+1}) - \\
&\qquad \frac{h(h-1)(h-2)}{2}g(x_{k+2}) + \frac{h(h-1)(h+1)}{6}g(x_{k+3})
\end{aligned}$$

where $h \in [x_k, x_{k+1}]$. The interpolation kernels are inferred from these interpolator functions and the impulse responses are given by the expressions

$$\text{Linear Interpolator} \quad \begin{cases} u(x) = 1 - |x| & \text{for } 0 < |x| < 1 \\ \quad = 0 & \text{outside} \end{cases}$$

$$\text{Quadratic Interpolator} \quad \begin{cases} u(x) = -|x|^2 + 1 & \text{for } 0 < |x| < \frac{1}{2} \\ \quad = -\frac{1}{2}|x|^2 - \frac{3}{2}|x| + 1 & \text{for } \frac{1}{2} < |x| < \frac{3}{2} \\ \quad = 0 & \text{outside} \end{cases}$$

$$\text{Cubic Interpolator} \begin{cases} u(x) = \frac{1}{2}|x|^3 - |x|^2 + \frac{1}{2}|x| + 1 & \text{for } 0 < |x| < 1 \\ = -\frac{1}{6}|x|^3 + |x|^2 - \frac{1}{6}|x| + 1 & \text{for } 1 < |x| < 2 \\ = 0 & \text{outside} \end{cases}$$

and the impulse responses of their derivatives are obtained simply by differentiating the impulse response with respect to the variable x. Figure 2.8 (a) shows the impulse response of the linear, quadratic and cubic interpolators. Figure 2.8 (b) displays the impulse response of their derivatives. Note that the linear and cubic interpolators have continuous impulse responses and that the quadratic impulse response is discontinuous at $x = -1.5$, $-.5$, $.5$ and 1.5, while the impulse response derivatives are discontinuous for the linear and cubic interpolator but continuous for the quadratic interpolator. Experiments have shown that motion estimation algorithms are far more sensitive to gradient discontinuities than irradiance discontinuities. Discontinuous gradients can produce instability and divergence of the algorithms. Bilinear and bicubic interpolators have consistently produced far poorer results than the biquadratic one. Algorithms tend to be unstable with the former interpolators and badly behaved optical flows are produced when these interpolators are implemented in optical flow algorithms.

If the extra accuracy of the bicubic interpolator is required, it is possible to design an bicubic C^1 interpolator. Keys (1981) interpolator is a bicubic interpolator that provides both a continuous impulse and continuous derivative impulse (see figures 2.9 (a) and 2.9 (b)). In fact, Keys' interpolator matches the Taylor series development of the function at the origin up to the third derivative. Keys and biquadratic interpolators tend to perform equally well for optical flow computation although the optical flow produced by the Keys' interpolator is slightly smoother.

Quadratic surface fitting on raw data and Keys cubic interpolator on low–passed data, have comparable computational complexity and give similar results when used in the FICE method. However, the quadratic surface fitting scheme displays a clear advantage in the DFU method because the fit is computed using multiple unwarped frames that increase the noise smoothing and produce a more accurate parameterization of the irradiance function than the one generated by the pure intra–frame Keys cubic interpolator. After testing different interpolators, quadratic surface fitting was used exclusively in the implementation because it offered the most flexibility and very good results with noisy data at a reasonable computational cost.

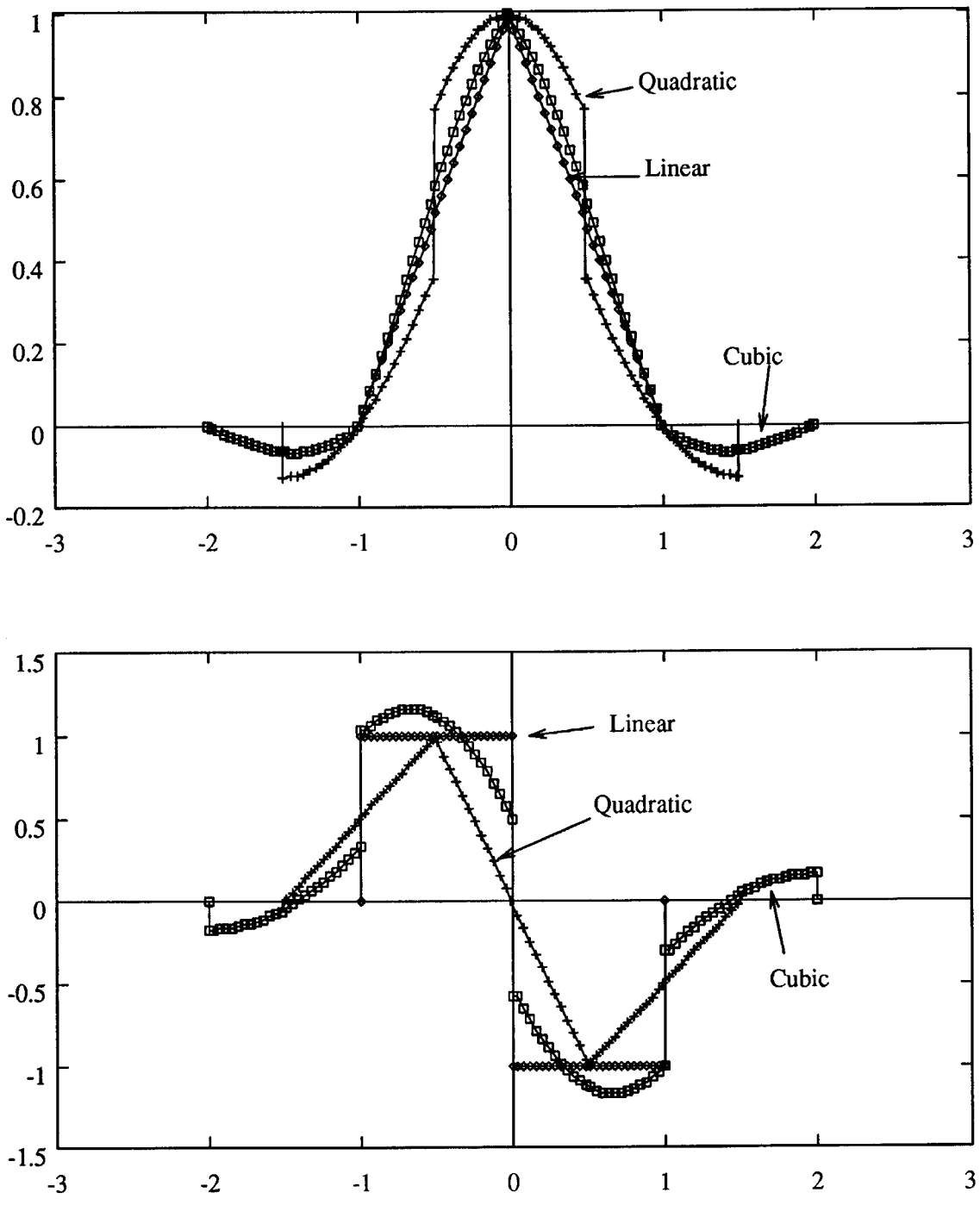

Figure 2.8: (a) Impulse responses of the linear, quadratic and cubic interpolators, (b) derivatives of impulse response of the linear, quadratic and cubic interpolators

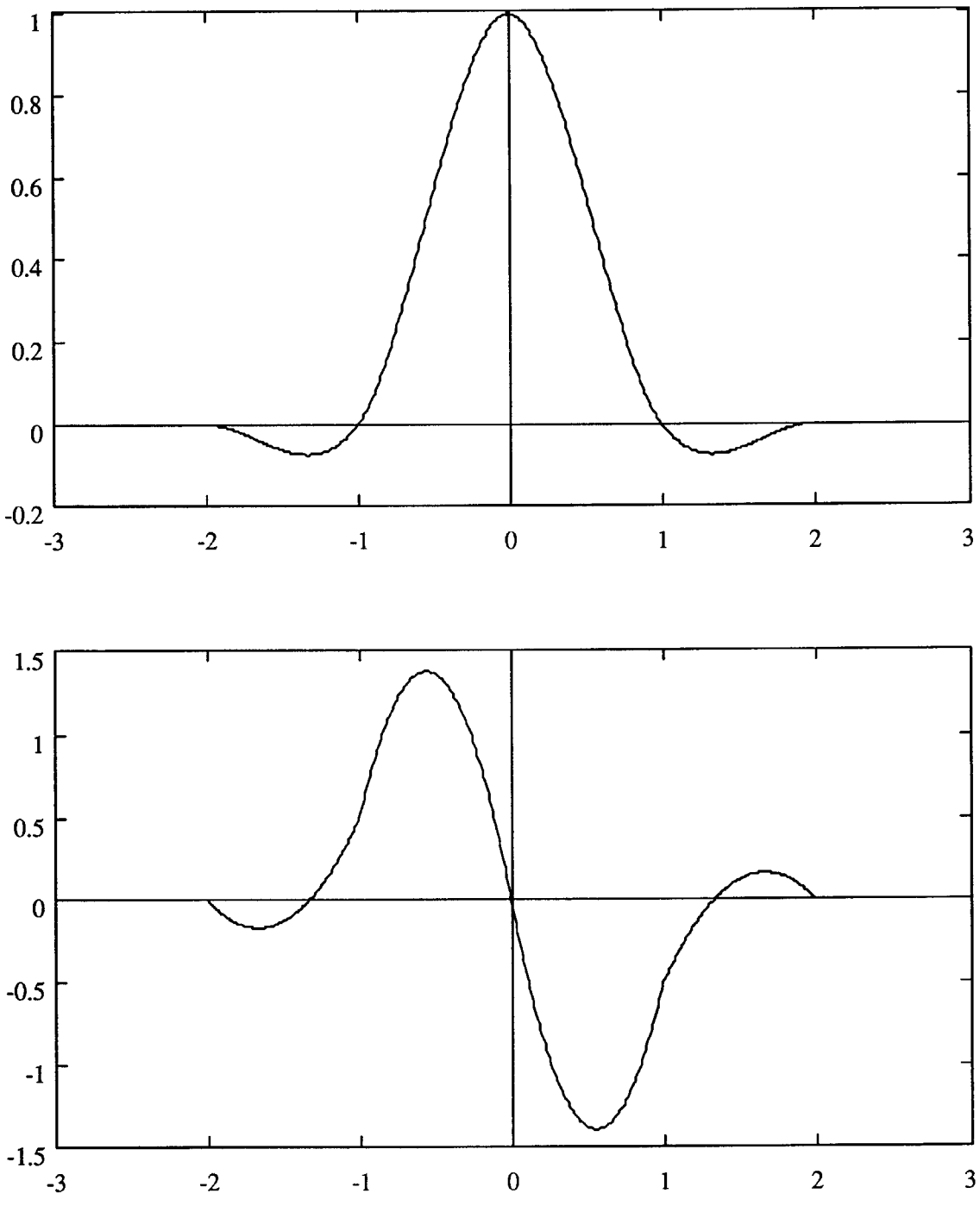

Figure 2.9: (a) Impulse response of the Keys cubic interpolator, (b) derivative of impulse response of the Keys cubic interpolator.

2.6 Summary

This chapter derived a differential constraint equation that links the irradiance spatiotemporal gradients, the motion and structure parameters and the temporal variations of the shading model of the surface. The constraint equation was derived for a general surface, or equivalently for an arbitrary depth map and a generic shading model. The surface was then specialized to be a polynomial patch in order to characterize it by a small number of parameters that could be estimated directly. Several constrained and unconstrained minimizations were presented in the general case for polynomial patches and one of the most complete and general models was solved to explicitly derive the system of vectorial nonlinear equations that defines the motion and structure parameters. The complexity and structure of the nonlinear system was described and several potential numerical methods presented and evaluated in terms of their ease of implementation, convergence properties and speed. Finally, the computation of the irradiance spatiotemporal gradients was discussed in detail and the advantages and drawbacks of several methods argued and compared in terms of their behavior in the presence of noise, efficiency in computation and flexibility.

Chapter 3

Shading Models

In general, the reflectance function is very complex and cannot be modeled easily. One possible approximation for the image irradiance is to consider it as the sum of an ambient (indirect) light term and of incident (direct) light term that is composed of the weighted sum of Lambertian and specular components. The ambient illumination represents the light incident from the environment, i.e. reflections of the direct light from all the materials in the scene, while the incident illumination is the light originating from specific light sources with no intermediate reflections. In practice, ambient light is not easy to estimate and subtract out but is an irrelevant issue when differential techniques to estimate the motion are used.

The use of specular reflections in conjunction with motion estimation is not easy and might not provide useful and reliable information. One of the problems is that the resulting expressions are very cumbersome and quite useless. Computer graphics has produced very realistic models for specular reflections and the more realistic models, like Bui-Tong's (1975) or Blinn-Torrance's (Torrance and Sparrow 1967), are very powerful in image synthesis but are rather awkward to use in analysis. Another problem is the unreliability of the information: specular and glossy reflections vary greatly with surface finish and a smudge or a fingerprint can change them a lot. Moreover, unless one is very careful in acquiring the data, sensors can saturate, in which case the data represent the sensor properties, not the highlight characteristics.

If the only goal is to eliminate the undesirable specular reflections, a possible segmentation of a patch into Lambertian and specular surfaces can be based on the heuristic that, most of the time, the light sources are "white" and a specular reflection corresponds to an image of

the source and, therefore, appears to be very bright and achromatic. If we are given a color representation of the data (like RGB or XYZ) we can compute the chromaticity components (rgb) or (xyz) and segment the latter based on the occurrence of bright, equal chromaticity patches.

In the rest of the thesis, we will assume that we are dealing with a purely Lambertian surface or a segmented Lambertian surface where the purely specular and glossy regions have been removed. Such a model is a good approximation to real data and allows us to account for the variations of irradiance due to shading in the constraint equation.

3.1 Lambertian Model

In general, the amount of light captured by a surface patch will depend on its inclination relative to the incident beam. As seen from the source, the surface is foreshortened and its projected area is its true area multiplied by the cosine of the incident angle i. Thus the irradiance is proportional to $\cos i$. In order to have the radiance of the surface patch proportional to $\cos i$, we need a surface that reflects all the incident light and appears equally bright from all viewing directions. The ideal diffuser, or Lambertian reflector, is such a surface.

3.1.1 General Lambertian Model

If we assume that the Lambertian surface has a continuously varying albedo ρ_λ with respect to the surface coordinates (α, β) and a radiance $\mathcal{L}(\mathbf{R})$, that can be expressed in the observer frame, the shading equation can be written as

$$E(\mathbf{r},t) = \mathcal{L}(\mathbf{R}) = \rho_\lambda(\alpha,\beta) L_0 \cos i = \rho_\lambda(\alpha,\beta) L_0 (\hat{\mathbf{L}} \cdot \hat{\mathbf{n}}) \tag{3.1}$$

with

$$\hat{\mathbf{L}} = \frac{(\mathbf{l} - \mathbf{R})}{\|\mathbf{l} - \mathbf{R}\|} \quad \text{and} \quad \mathbf{L} = \mathbf{l} - \mathbf{R}$$

where $\mathbf{l}$ represents the source position in the observer frame, $\mathbf{R}$ is a point on the surface measured in the observer frame, $\hat{\mathbf{n}}$ is the unit normal at the point $\mathbf{R}$ and L_0 is the intensity of the source. With these definitions, $\mathbf{L}$ represents the illumination vector from the point on the surface to the light source.

The irradiance equation (3.1) is the product of two distinct terms, a purely surface dependent term $\rho_\lambda(\alpha,\beta)$ that represents the markings or texture of the surface, and an illumination term $L_0(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})$. In general, the illumination term depends *both* on the position and normal of the illuminated point on the surface. Thus the spatial shading variations are due to the relative position of the point with respect to the source and the local orientation at that point. The spatial variations of the irradiance, with a nearby source, are complex to analyze even for simple surfaces like planar patches. In most cases, the source will be far enough from the surface (i.e. $\hat{\mathbf{L}}\simeq\hat{\mathbf{l}}$) and the shading variations will only depend on the local curvature. In particular, for a planar patch, the illumination term is constant for a distant source and the irradiance variations are purely due to the texture.

We saw in Section 2.3, that in the FICE, the shading effects are represented by the temporal variations of the irradiance equation. The temporal derivative of the shading can be expressed (see appendix D) as

$$\begin{aligned}\dot{E} &= \frac{\rho_\lambda(\alpha,\beta)L_0}{\|\mathbf{L}\|}\Big(\big(-(\boldsymbol{\omega}\times\mathbf{R}+\mathbf{t})\cdot\hat{\mathbf{n}}\big)+\big(\mathbf{L}\cdot(\boldsymbol{\omega}\times\hat{\mathbf{n}})\big)+\big((\boldsymbol{\omega}\times\mathbf{R}+\mathbf{t})\cdot\hat{\mathbf{L}}\big)(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})\Big)\\ &= \frac{\rho_\lambda(\alpha,\beta)L_0}{\|\mathbf{L}\|}\Big(\underbrace{[\mathbf{l},\boldsymbol{\omega},\hat{\mathbf{n}}]+(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}]}_{\dot{E}_{\boldsymbol{\omega}}}-\underbrace{\Big(\big(\mathbf{t}-(\mathbf{t}\cdot\hat{\mathbf{L}})\hat{\mathbf{L}}\big)\cdot\hat{\mathbf{n}}\Big)}_{\dot{E}_{\mathbf{t}}}\Big)\end{aligned}\tag{3.2}$$

where the term $\dot{E}_{\boldsymbol{\omega}}$ is due to rotational motion and the term $\dot{E}_{\mathbf{t}}$ is due to translation. The notation $[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}]$ represents the mixed product (determinant) of the three vectors $\hat{\mathbf{L}}$, $\boldsymbol{\omega}$ and $\mathbf{R}$.

The rotational component of the temporal derivative can itself be decomposed in two distinct elements, a "distant" term $\dot{E}^{\infty}_{\boldsymbol{\omega}}=[\mathbf{l},\boldsymbol{\omega},\hat{\mathbf{n}}]/\|\mathbf{L}\|$ and a "near" term $(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}]$. The former term can be approximated by $[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}]$ if $\|\mathbf{R}\|/\|\mathbf{l}\|\ll 1$ and captures the contribution to the temporal shading variations of a infinitely distant point source, while the latter term represents the temporal variations of the shading induced by the nearby spatial variations of shading.

The translational component only depends on that part of translation orthogonal to $\hat{\mathbf{L}}$, $\mathbf{t}^{\mathbf{f}}_{\perp}=\mathbf{t}-(\mathbf{t}\cdot\hat{\mathbf{L}})\hat{\mathbf{L}}$. This result could have been predicted directly since a translation does not change the orientation of the patch; the incident angle of the rays is the same in the case of a translation along the rays and irradiance values of a Lambertian surface only depend on the relative angle between the surface normal and light source direction.

The expression (3.2) for the temporal shading is, in general, complicated to use in the constraint equation. Part of the complexity is due to the presence of the variable $\mathbf{R}$ (explicit in $\dot{E}_{\boldsymbol{\omega}}$ and implicit in $\|\mathbf{L}\|$). In fact, expressing $\mathbf{R}$ in terms of the image plane variable $\mathbf{r}$, $\mathbf{R} = \frac{Z\mathbf{r}}{F}$ makes explicit the dependency of $\dot{E}$ on the depth, or structure. To remove this dependency, special cases of motion, like pure translation or rotation, and/or approximate expressions for $\dot{E}$ need to be considered.

3.1.2 First–Order Lambertian Model

In most situations, the relative distance of the light source from the object and object to camera is very small i.e. $\|\mathbf{R}\|/\|\mathbf{l}\| \ll 1$, and $\dot{E}$ can be approximated (see appendix D.2) by a first–order Taylor series with respect to $\|\mathbf{R}\|/\|\mathbf{l}\|$,

$$\dot{E} = \rho_\lambda(\alpha,\beta)L_0\left([\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}] + (\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{R}}]\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} - (\mathbf{t}_\perp^{\hat{\mathbf{l}}}\cdot\hat{\mathbf{n}})\frac{1}{\|\mathbf{l}\|}\right) \tag{3.3}$$

where $\mathbf{t}_\perp^{\hat{\mathbf{l}}} = \mathbf{t} - (\mathbf{t}\cdot\hat{\mathbf{l}})\hat{\mathbf{l}}$ and $\hat{\mathbf{R}} = \hat{\mathbf{r}}$ is the unit viewing direction vector.

Unlike the rotational component $\dot{E}_{\boldsymbol{\omega}}$, the translational term $-\rho L_0(\mathbf{t}_\perp^{\hat{\mathbf{l}}}\cdot\hat{\mathbf{n}})/\|\mathbf{l}\|$ in (3.3) only depends on the position of the light source, on the translation and orientation of the patch and is independent of the structure of the patch.

Assuming that the light source is at infinity ($\|\mathbf{l}\| \to \infty$), in the direction $\hat{\mathbf{l}}$, the previous equation simplifies greatly to

$$\dot{E} = \rho_\lambda(\alpha,\beta)L_0[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}]. \tag{3.4}$$

The previous expression (3.4), for a distant point source, can be derived directly from the distant Lambertian shading equation,

$$E = \rho_\lambda(\alpha,\beta)L_0(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}) \tag{3.5}$$

by differentiation with respect to time

$$\dot{E} = \rho_\lambda L_0\left(\frac{d\hat{\mathbf{n}}}{dt}\cdot\hat{\mathbf{l}}\right) = \rho_\lambda L_0\big((\boldsymbol{\omega}\times\hat{\mathbf{n}})\cdot\hat{\mathbf{l}}\big) = \rho_\lambda L_0[\boldsymbol{\omega},\hat{\mathbf{n}},\hat{\mathbf{l}}].$$

3.1.2.1 Lambertian Model With Hemispherical Source Along $\hat{\mathbf{l}}$–Axis

In this model we assume that the light source is an hemispherical uniform source along the axis $\hat{\mathbf{l}}$. The shading equation, see (Horn and Sjoberg 1979), takes the form

$$E(\mathbf{r},t) = \mathcal{L}(\mathbf{R}) = \frac{1}{2}\rho_\lambda(\alpha,\beta)L_0\big(1 + (\hat{\mathbf{n}}\cdot\hat{\mathbf{l}})\big)$$

and the temporal variations of this shading model are expressed as

$$\dot{E} = \frac{1}{2}\rho_\lambda(\alpha,\beta)L_0[\omega,\hat{\mathbf{n}},\hat{\mathbf{l}}]. \tag{3.6}$$

This hemispherical source model can be generalized to an arbitrary light distribution. It can be shown that an arbitrary light distribution on a unit sphere is equivalent to a single, infinitely distant, collimated light source in a direction $\hat{\mathbf{l}}$. The punctual light source direction $\hat{\mathbf{l}}$ is defined by the unit vector that links the origin of the sphere to the center of mass of the light distribution on the unit sphere.

It is remarkable that equation 3.4, for the general distant Lambertian case, and equation 3.6, for the hemispherical extended source along the $\hat{\mathbf{l}}$–axis, are formally identical within a multiplicative constant factor 1/2. However, this formal identity is only superficially true. In fact, the irradiance of a Lambertian surface illuminated by a collimated source is expressed by

$$E(\mathbf{r},t) = \rho_\lambda(\alpha,\beta)L_0\max\big(0,(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})\big),$$

with the result that the temporal variations of the irradiance of a Lambertian surface for a distant source and a uniform hemispherical source along the $\hat{\mathbf{l}}$–axis are only identical, within a scale factor, for surface orientation such that $(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}) > 0$ and differ otherwise. Figure 3.1 shows the irradiance function as a function of the product $(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})$ for a distant source and an hemispherical source. In the following sections, distant point sources and hemispherical extended sources along the $\hat{\mathbf{l}}$-axis will not be distinguished although they are *not* identical.

3.2 Attenuated Lambertian Model

The Lambertian model presented in the previous section 3.1.1 only applies to sources and conditions where the apparent brightness of the source does not significantly change with distance.

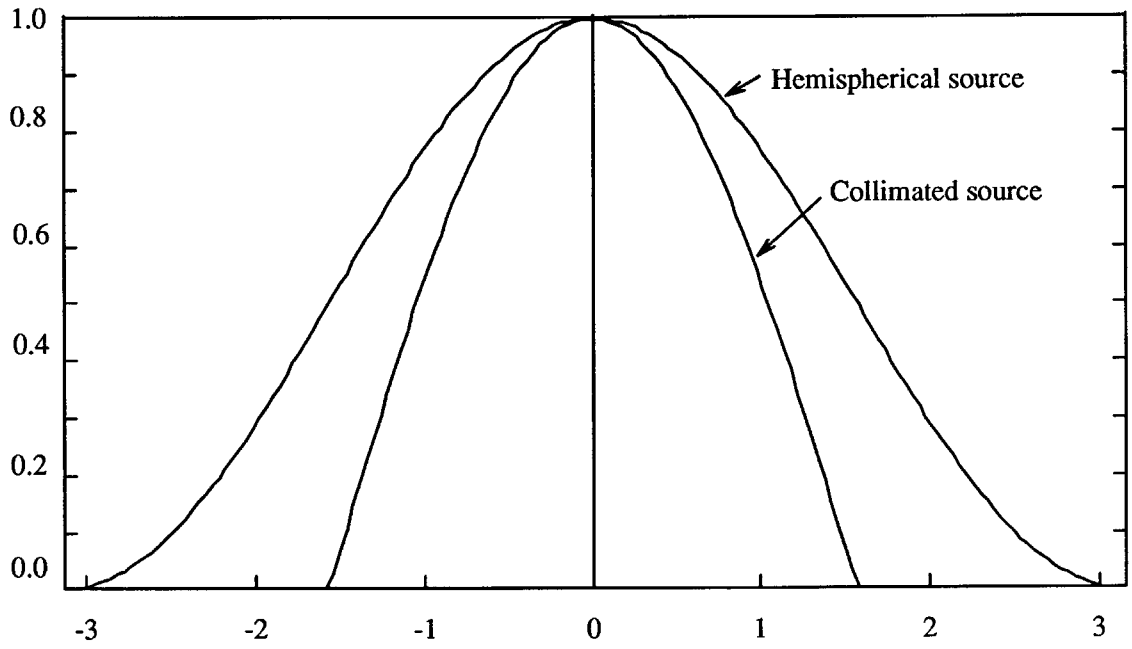

Figure 3.1: Lambertian reflectance for a collimated distant source and for an hemispherical uniform source

When dealing with artificial light sources the attenuation of the intensity of the light source with distance must be taken into account. The power emitted by a light source is dissipated isotropically in a spherical volume resulting in an effective attenuation that is a function of the inverse of the square of the distance between the light source and the object. This attenuation is not an issue for a natural light source like the sun because the sun is so far away that no changes in attenuation can be measured on earth.

In general, additional attenuation is produced by scattering and diffraction but its study in different mediums is complex and requires the knowledge of the physical properties of that medium. In this section, we consider a case of practical importance where the attenuation needs to be taken into account, the case of underwater photography or filming, with artificial lights.

3.2.1 General Attenuated Model

In clean, undisturbed water, a Lambertian surface illuminated by a punctual source can be modeled by a Lambertian reflectance function where the apparent intensity of the light source decreases as the square of the distance between the source and the patch. This model does

not take into account the phenomenon of light scattering and diffraction caused by microscopic particles in suspension in the water and the variations in transmission and refraction indices of the water due to temperature gradients and underwater currents. However, this attenuated Lambertian model is a reasonable approximation of underwater vision and the irradiance of the Lambertian surface can be expressed as

$$E(\mathbf{r},t) = \frac{\Lambda}{R^2}(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}}) \tag{3.7}$$

where $R = \|R\|$ represents the distance from the camera to the patch.

It is straightforward to compute the temporal derivative of equation 3.7 using the expression of the temporal derivative of the general shading equation. If $\dot{E}_{\text{General Lamb}}$ represents the temporal derivative of the irradiance of a Lambertian patch, as defined by equation (3.2), the underwater shading equation can be written as

$$\begin{aligned}\dot{E} &= \frac{1}{R^2}\dot{E}_{\text{General Lamb}} - \frac{2\rho_\lambda L_0}{R^4}(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})(\mathbf{R}\cdot\mathbf{t}) \\ &= \frac{\rho_\lambda L_0}{R^2\|\mathbf{L}\|}\left([\mathbf{l},\boldsymbol{\omega},\hat{\mathbf{n}}] + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}] - (\mathbf{t}_\perp^{\hat{\mathbf{L}}}\cdot\hat{\mathbf{n}}) - 2\frac{(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})(\mathbf{R}\cdot\mathbf{t})}{R^2}\right).\end{aligned} \tag{3.8}$$

The resulting expression is similar to the simpler expression (3.2) but is of little practical value unless reasonable simplifying assumptions can be made. Unlike the regular Lambertian case, it cannot be assumed that the light source is distant, since beyond a small zone, all light is absorbed by the water and none would reach the object. On the other hand, the problem can be simplified by assuming a realistic, particular geometry of light source and camera.

3.2.2 Light Source, Viewer Approximation

In a typical underwater set–up, the lights are mounted directly on the camera[1] and, to a first approximation, light and camera can be assumed to be coincident. Under this assumption, the light source and the camera are located at the origin of the coordinates, $\mathbf{L} = -\mathbf{R}$, $\mathbf{l} = 0$, and the general equation 3.8 simplifies to

$$\begin{aligned}\dot{E} &= -\frac{\rho_\lambda L_0}{R^3}((\mathbf{t}\cdot\hat{\mathbf{n}}) - 3(\hat{\mathbf{R}}\cdot\mathbf{t})(\hat{\mathbf{R}}\cdot\hat{\mathbf{n}})) \\ &= -\frac{\rho_\lambda L_0}{R^3}\left(\left(\mathbf{I}_3 - 3\hat{\mathbf{R}}\hat{\mathbf{R}}^T\right)\mathbf{t}\cdot\hat{\mathbf{n}}\right) = -\frac{\rho_\lambda L_0}{R^3}\left(\left(\mathbf{I}_3 - 3\hat{\mathbf{r}}\hat{\mathbf{r}}^T\right)\mathbf{t}\cdot\hat{\mathbf{n}}\right).\end{aligned} \tag{3.9}$$

[1] Although convenient for a single manned camera, much better results are obtain by widely separating the camera and light sources.

The latter expression is formally equivalent to a Lambertian–like expression where the light source vector $\hat{\mathbf{L}}$ is replaced by $\left(\mathbf{t} - 3(\hat{\mathbf{R}} \cdot \mathbf{t})\hat{\mathbf{R}}\right)/R^3$, a space and motion variant attenuated "light source". Since the light source and the camera are spatially coincident, the temporal variations of the irradiance only depend on the translation of the object and not on the rotation measured in the camera frame.

3.3 Constraint Equation vs FICE

The validity of the classical constraint equation (CE) has been qualitatively challenged due to its unjustified use. Opponents of the CE argue that the brightness constancy assumption is physically impossible and there is always *some* variation of the irradiance as the object or camera moves. Proponents argue that the CE is a good *approximation* of the physical reality and use optical flow estimates computed from the CE as evidence.

Part of the confusion is that although the experimental conditions and the set–up of the problem widely vary, the CE is used indiscriminately. At least three factors are important in determining the validity of the CE: the physical structure of the problem, the nature of the surface and the type of motion. Two different classes of problems are generally considered, the passive navigation problem where the environment is fixed relative to the moving observer (camera) and the situation where an object is moving in the environment with respect to a fixed camera. These two problems are in general *not* the inverse of each other. They are equivalent in a kinematic sense, i.e. the motion of the camera with respect to the environment is the inverse of the motion of the environment relative to a fixed camera, but they are not, in general, photometric inverses because of the light sources and because of the reflectance properties of the objects. In the passive navigation problem, the overall environment moves with respect to the camera, but imaged objects are fixed relative to the light sources. Therefore, there are no surface shading variations induced by the relative motion of the object and light sources.

The second factor, the nature of the surface, can be divided into three distinct components: the surface reflectance property, the albedo and the shape of the surface. The surface reflectance of the imaged object dictates the amount and type of shading variations that occur when either the light sources move relative to the object, the camera moves relative to the object or both happen. We saw in section 2.1.2 that the surface reflectance is described by the bidirectional

reflectance distribution function that depends on *both* the illumination and viewing directions. For example, the brightness of a Lambertian surface is independent of the viewing direction, therefore no temporal irradiance variations occur in the passive navigation situation and the constraint equation is exact. In general, no surface is truly Lambertian but the changes in irradiance caused by the variation of the viewing angle are usually negligible. The albedo of the surface, or texture, is a major factor in determining the validity of the CE approximation. Horn (1988) shows in the passive navigation case, using a sinusoidal grating on the image plane, that the changes of brightness of the surface with time are small as long as there is significant contrast at high spatial frequencies, i.e. the image spatial gradients due to the texture are dominant in the CE with respect to the temporal changes in shading (texture driven CE). The approximation becomes worse as the surface markings become weaker. At the limit, for a textureless surface, all variations are due to the shading (shading driven CE) and, by design, the classical CE cannot deal with these cases.

Surface shape is another important element in determining the validity of the CE approximation. The approximation is the best for a planar surface and is increasingly worse for highly curved surfaces. Shading and its variations depends on the normal of the patch; as the normal varies rapidly, the spatial and temporal variations of shading become larger.

The third factor, the type of motion, is also a key element in the validity of the CE in the case of an object moving relative to the light sources. Intuitively, rotational changes are expected to be substantial because rotation causes a change in the surface patch orientation and therefore, a modification of the irradiance. On the other hand, few variations are anticipated in the translation case. This intuitive analysis is consistent with the expressions obtained for the temporal variations of the shading (equation 3.2 in the general case). In fact, looking at the first–order expansion (3.3) of the general equation (3.2), shows that the temporal variations of shading depends on the rotational motion $\boldsymbol{\omega}$ with a zero–order term $[\hat{\mathbf{I}}, \boldsymbol{\omega}, \hat{\mathbf{R}}]$, but on the translational motion $\mathbf{t}$ with a first-order term $(\mathbf{t}^{\hat{\mathbf{I}}}_{\perp} \cdot \hat{\mathbf{n}}) \frac{1}{\|\mathbf{I}\|}$.

The analysis of the effects of the temporal variations of the shading can be carried out analytically to some extent and experimentally in a broader sense. The next section focuses on the analytical analysis of some configurations of texture and shading models in order to evaluate the importance of the various terms in the FICE.

3.3.1 Analytical Analysis of the FICE

The goal of the analysis is to separately compute the three components of the FICE and evaluate their relative magnitude as a function of the texture and the motion. The three constituents of the FICE are $(\nabla_{\mathbf{r}}\mathbf{E} \cdot \dot{\mathbf{r}})$, the product of the spatial gradient with the optical flow, E_t, the interframe change of the irradiance and $\frac{dE}{dt}$, the rate of change of the irradiance. In order to compute the various quantities in terms of texture, the texture of the object on the image plane needs to be tracked to express the image coordinates of the projected object in terms of the original surface coordinates of the texture on the object.

In general, E_t is the most complex quantity to evaluate analytically because it relates the irradiance of two separate surface points that have different albedos and surface normals. $\nabla_{\mathbf{r}}E$ is complicated to compute for a general surface since the variations of irradiance are due to the variations of both surface normals and texture. However, for planar surfaces, the normal is constant and the spatial variations are only due to the spatial variations of the albedo, unless the source is close to the surface and each point of the surface sees a light source at a different spatial location. $\dot{E}$ relates the irradiance of two image points that represent the same point on the object and, therefore, have the same albedo but a different unit normal and is easily computed.

In order to compute the different quantities of the FICE in terms of the texture on the surface, we need to express the geometric relationship between the surface coordinates $\mathbf{U}^T = (u, v, w)$ of the object, in which the texture is fixed, and the world coordinates $\mathbf{X}$ or image coordinates $\mathbf{x}$. For clarity, the equations are shown for the planar case only. The formulation for a general surface is similar, although the mapping between the surface and world coordinates can be much more complex. Let us consider the rotation matrix $\mathbf{P}$ and the translation $\mathbf{X}_0$ such that

$$\mathbf{X} = \mathbf{P}\mathbf{U} + \mathbf{X}_0$$

or, equivalently,

$$\mathbf{U} = \mathbf{P}^T(\mathbf{X} - \mathbf{X}_0). \tag{3.10}$$

It is always possible to find a nonunique matrix $\mathbf{P}$ and vector $\mathbf{X}_0$ that maps the arbitrary plane in space to a frontal plane at the origin of the world coordinates, i.e. the 2–D surface coordinates

$\mathbf{u}^T = (u, v)$ coincide with the 3–D world coordinates $\mathbf{X}^T = (X, Y, 0)$ and with the 2–D image coordinates $\mathbf{x}^T = (x, y)$ at the origin. Equation 3.10 enables us to directly compute the albedo, given by a function of the surface coordinates, at an arbitrary point $\mathbf{X}$ of the object. In most cases, it is more convenient to directly consider the relationship between the image and surface coordinates, since we observe the irradiance function $E(\mathbf{x}, t)$ which is expressed as a function of the image coordinates. If $\hat{\mathbf{n}}$ represents the unit normal of the planar surface, we have the dual relations (assuming a unity focal distance)

$$\mathbf{r} = \frac{1}{(\mathbf{PU} \cdot \hat{\mathbf{z}}) + Z_0}(\mathbf{PU} + \mathbf{X_0})$$

$$\mathbf{U} = \mathbf{P}^T \left(\frac{\mathbf{X_0} \cdot \hat{\mathbf{n}}}{\mathbf{r} \cdot \hat{\mathbf{n}}} \mathbf{r} - \mathbf{X_0} \right). \tag{3.11}$$

The irradiance spatial gradients are computed from the albedo gradients by means of the Jacobian $\mathbf{J_{ux}}$ of the transformation from $\mathbf{x}$ to $\mathbf{u}$. For a planar surface defined by the normal $\hat{\mathbf{n}}$ and illuminated by a infinitely distant light source in the direction $\hat{\mathbf{l}}$, $E(\mathbf{x}, t) = L_0(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})\rho(\mathbf{u})$. The spatial irradiance gradients are expressed by

$$\begin{aligned} \nabla_{\mathbf{x}} E &= L_0(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}) \mathbf{J_{ux}} \nabla_{\mathbf{u}} \rho \\ &= L_0(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}) \frac{\mathbf{X_0} \cdot \hat{\mathbf{n}}}{(\mathbf{r} \cdot \hat{\mathbf{n}})^2} \left(\left((\mathbf{x} \cdot \tilde{\mathbf{n}}) \mathbf{I}_2 - \tilde{\mathbf{n}} \mathbf{x}^T \right) \tilde{\mathbf{P}} + (\tilde{\mathbf{P}} - \tilde{\mathbf{n}} \tilde{\mathbf{p}}^T) \right) \nabla_{\mathbf{u}} \rho, \end{aligned}$$

where $\tilde{\mathbf{P}} = \mathbf{I}_{2,3} \mathbf{P} \mathbf{I}_{3,2}$ is a restriction of the rotation matrix $\mathbf{P}$, $\tilde{\mathbf{p}}^T = (\mathbf{P}_{31}, \mathbf{P}_{32})$ is the vector formed by the first two components of the last column of the matrix $\mathbf{P}$ and $\frac{(\mathbf{X_0} \cdot \hat{\mathbf{n}})}{(\mathbf{r} \cdot \hat{\mathbf{n}})} = \frac{1}{Z}$. Using the expression of optical flow (equation 2.9) rewritten in the matrix form

$$\dot{\mathbf{r}} = \mathbf{A}\boldsymbol{\omega} + \frac{1}{Z} \tilde{\mathbf{T}}$$

where

$$\mathbf{A} = \begin{pmatrix} -xy & 1 + x^2 & -y \\ -(1 + y^2) & xy & x \end{pmatrix} \quad \text{and} \quad \tilde{\mathbf{T}} = \begin{pmatrix} t_x - x t_z \\ t_y - y t_z \end{pmatrix}$$

the term $(\nabla_{\mathbf{x}} \mathbf{E} \cdot \dot{\mathbf{r}})$ can be computed in term of the motion parameters, the image coordinates and the analytical expression of the albedo spatial gradients

$$(\nabla_{\mathbf{x}} \mathbf{E} \cdot \dot{\mathbf{r}}) = L_0(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}) \frac{\mathbf{X_0} \cdot \hat{\mathbf{n}}}{(\mathbf{r} \cdot \hat{\mathbf{n}})^2} (\boldsymbol{\omega}^T \mathbf{A} + \frac{1}{Z} \tilde{\mathbf{T}}^T) \left(\left((\mathbf{x} \cdot \tilde{\mathbf{n}}) \mathbf{I}_2 - \tilde{\mathbf{n}} \mathbf{x}^T \right) \tilde{\mathbf{P}} + (\tilde{\mathbf{P}} - \tilde{\mathbf{n}} \tilde{\mathbf{p}}^T) \right) \nabla_{\mathbf{u}} \rho. \tag{3.12}$$

In the case of a distant light source, the temporal variations of the shading are given by (3.4), and can be rewritten in the form

$$\dot{E}(\mathbf{x},t) = L_0\rho(\mathbf{u})[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{R}}] \tag{3.13}$$

with $\mathbf{u}$ is expressed in terms of the image coordinates by means of (3.11).

Equations 3.12 and 3.13 allow the computation of the two terms $\dot{E}$ and $(\nabla_{\mathbf{x}}\mathbf{E}\cdot\dot{\mathbf{r}})$ as a function of the motion and texture. The ratio δ of $\dot{E}$ and $(\nabla_{\mathbf{x}}\mathbf{E}\cdot\dot{\mathbf{r}})$ that can be computed using (3.12) and (3.13) when the denominator is nonzero is a meaningful quantity in judging the accuracy of the CE approximation. This criterion is used in the examples of the next section to display the accuracy of the constraint equation.

It is difficult to draw general conclusions from the expressions (3.12) and (3.13) because their analytic form is fairly complex in the general case. If the simplified case of a sinusoidal grating $\rho(\mathbf{u}) = \big(1+\sin(\mathbf{f}\cdot\mathbf{u})\big)$, where $\mathbf{f}^T = (f,g)$ represents the vector of spatial frequencies of the grating on a frontal plane (i.e. $\mathbf{P} = \mathbf{I}_3$) is used, the previous equations take the simplified form

$$\dot{E} = L_0[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{R}}]\big(1+\sin(\mathbf{f}\cdot\mathbf{x})\big) \tag{3.14}$$

and

$$(\nabla_{\mathbf{x}}\mathbf{E}\cdot\dot{\mathbf{r}}) = L_0(f\dot{x}+g\dot{y})\cos(\mathbf{f}\cdot\mathbf{x}). \tag{3.15}$$

Equation 3.15 suggests that the term $(\nabla_{\mathbf{x}}\mathbf{E}\cdot\dot{\mathbf{r}})$ of (3.14) can be very large compared to the term $\dot{E}$ due to the multiplicative factor $(f\dot{x}+g\dot{y})$, provided the motion is not parallel to the ridge of the grating, i.e. $(f\dot{x}+g\dot{y}) = 0$. However, for a given spatial frequency $\mathbf{f}$, $\dot{E}$ may not be negligible with respect to the other term, since $\dot{E}$ depends on the rotational velocity $\boldsymbol{\omega}$ and *both* terms are multiplied by the texture contrast L_0. This example demonstrates that when the object moves with respect to the light sources, Horn's conclusions (Horn and Weldon 1988) cannot be inferred even in the same simple case.

The previous discussion shows that, even in very simplified situations, it is not obvious that the temporal shading variations are negligible in the constraint equation. In practice, a texture contains many different spatial frequencies and the accuracy of the CE approximation can only be judged by the tedious process of numerically evaluating the different terms of the FICE and comparing their magnitudes, or by performing the parameter estimation with the CE and

Figure	Rotation ω in degrees	Translation $\mathbf{t}$ in pixels
a–b	(.5,-.4,.6)	(.64,.4,-.32)
c–d	(.5,-.4,.6)	(.064,.04,-.032)
e–f	(.05,-.04,.06)	(.064,.04,-.032)

Table 3.1: δ–plots motion parameters.

FICE on known surfaces with a known motion and by comparing the accuracy of the motion and structure estimates. The next section provides a few examples that qualitatively show the goodness of the CE approximation for different textures and amounts of motion.

3.3.2 Qualitative Assessment of the CE Approximation

The examples presented in this section use the ratio δ as a measure of accuracy of the CE approximation. Specifically, an 8–bit grey level, thresholded plot of the ratio δ, expressed as a percentage, is used. The overall image is shifted by 128 to represent signed quantities. Consequently, neutral grey represents 0% ratio, i.e. $\dot{E} = 0$. Once a threshold T is chosen, all values of δ such that $\delta > T$, are mapped into black (0), and similarly values of δ such that $\delta < -T$ are mapped into white (255). All other values $\delta \in [-T, T]$ are uniformly mapped by a grey scale ramp. Two thresholds, 5% and 10%, are used in the plots.

The first example shows the influence of rigid motion, rotation and translation, on the CE for a multiplicative sinusoidal grating. Figure 3.2(a) displays the irradiance of a plane rotated 10° around the $\hat{\mathbf{x}}$–axis, 15° around the $\hat{\mathbf{y}}$–axis and 30° around the $\hat{\mathbf{z}}$–axis, and mapped with the texture $\rho(\mathbf{u}) = 128\big(1 + \cos(\pi u)\cos(\pi v)\big)$, with $u, v \in [-1, 1]$. Figure 3.2(b–d) represents the temporal gradient, x–gradient and y–gradient respectively for a rigid motion of the plane given by the parameters $\omega^T = (.5, -.4, .6)$, expressed in degrees of rotation around the elementary axis, and $\mathbf{t}^T = (.64, .4, -.32)$ expressed in image pixels. Figures 3.3(a-f) represent the δ–plots with a threshold of 10% for the images on the left and 5% for the images on the right. Table 3.1 summarizes the values of the motion parameters for the various images of figure 3.3(a–f).

The results in the figures 3.3(a–f) confirm the intuition and the analytical results of the previous section. Translation has a very small effect on the CE accuracy while rotation has a major impact. Only a slight difference is visible between the images 3.3(a–b) and 3.3(c–d)

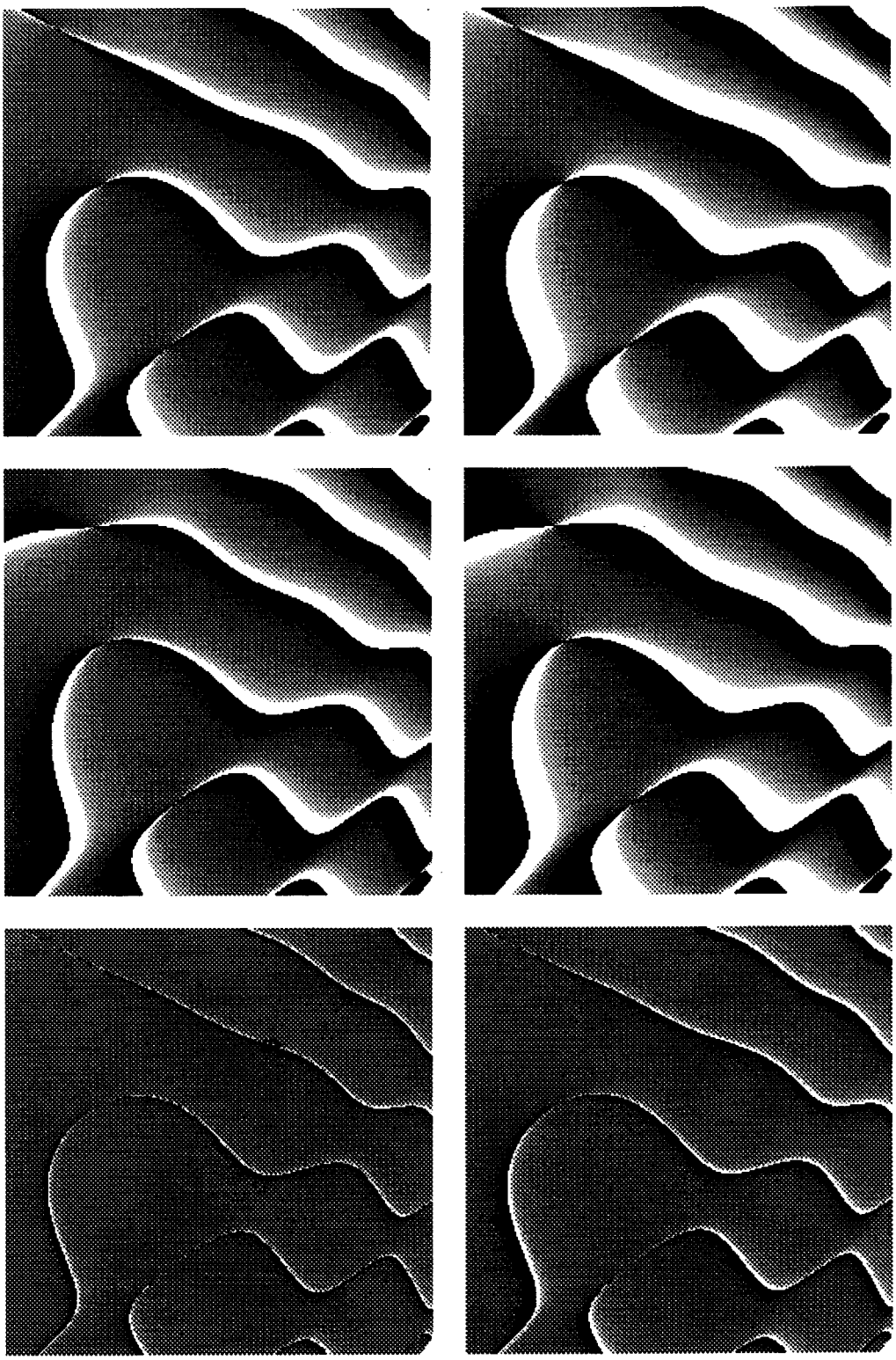

Figure 3.3: 10% and 5% δ–plots for three sets of motion parameters, given by table 3.1, with the cosine grating. The plots on the left ((a) top, (c) middle, (e) bottom) are the 10% plots, the plots on the right ((b) top, (d) middle, (f) bottom) the 5% plots.

although the translation has been decreased by an order of magnitude. On the other hand, a dramatic change is observed when the rotation is decreased by an order of magnitude from figure 3.3(c–d) to figure 3.3(e–f).

The second and third example shows the influence of texture on the accuracy of the CE. The example of figure 3.4 uses a cosine grating of multiplicative ramps, defined by $\rho(\mathbf{u}) = 128\big(1 + \cos(\pi^2 uv)\big)$, with $(u, v) \in [-1, 1]$, as a texture mapped onto a plane identical to the one used in figure 3.2. The example of figure 3.5 uses an exponentially damped cosine defined by $\rho(\mathbf{u}) = 128\big(1 + e^{-1.5|\mathbf{u}|}\cos(2\pi|\mathbf{u}|)\big)$, with $(u, v) \in [-1, 1]$ and is mapped on the same planar surface as before. The motion parameters of the moving plane in these two cases are the same as the one used in figure 3.3(a–b), namely $\boldsymbol{\omega}^T = (.5, -.4, .6)$ and $\mathbf{t}^T = (.64, .4, -.32)$. These examples demonstrate that for some types of texture the CE is a fairly bad approximation and the FICE should be used instead.

3.4 Summary

In this chapter, several Lambertian models were presented and their temporal variations computed and examined. The general Lambertian model was considered first. This model captures both the shading effects of an infinitively distant source and the spatiotemporal variations induced by a nearby source and is therefore fairly complex especially for high order surfaces. Two simplifications of this general model were derived, a first–order expansion in terms of the ratio of the distance between the object and the camera and the distance between the light source and the camera, and the case of an infinitely distant source. The former simplified model is helpful for sources that are neither distant nor immediately next to the surface, while the latter model represents the usual situation of natural light illuminating objects on earth and was shown to be very similar, but not identical, to the case of extended light sources. An attenuated Lambertian model was proposed and simplifying assumptions introduced in the case of underwater photography. Finally, an analytical analysis of the FICE was done and the validity of the CE examined. More specifically, the importance of shading and the influence of the type of motion and texture was evaluated in order to assess the validity of the CE approximation.

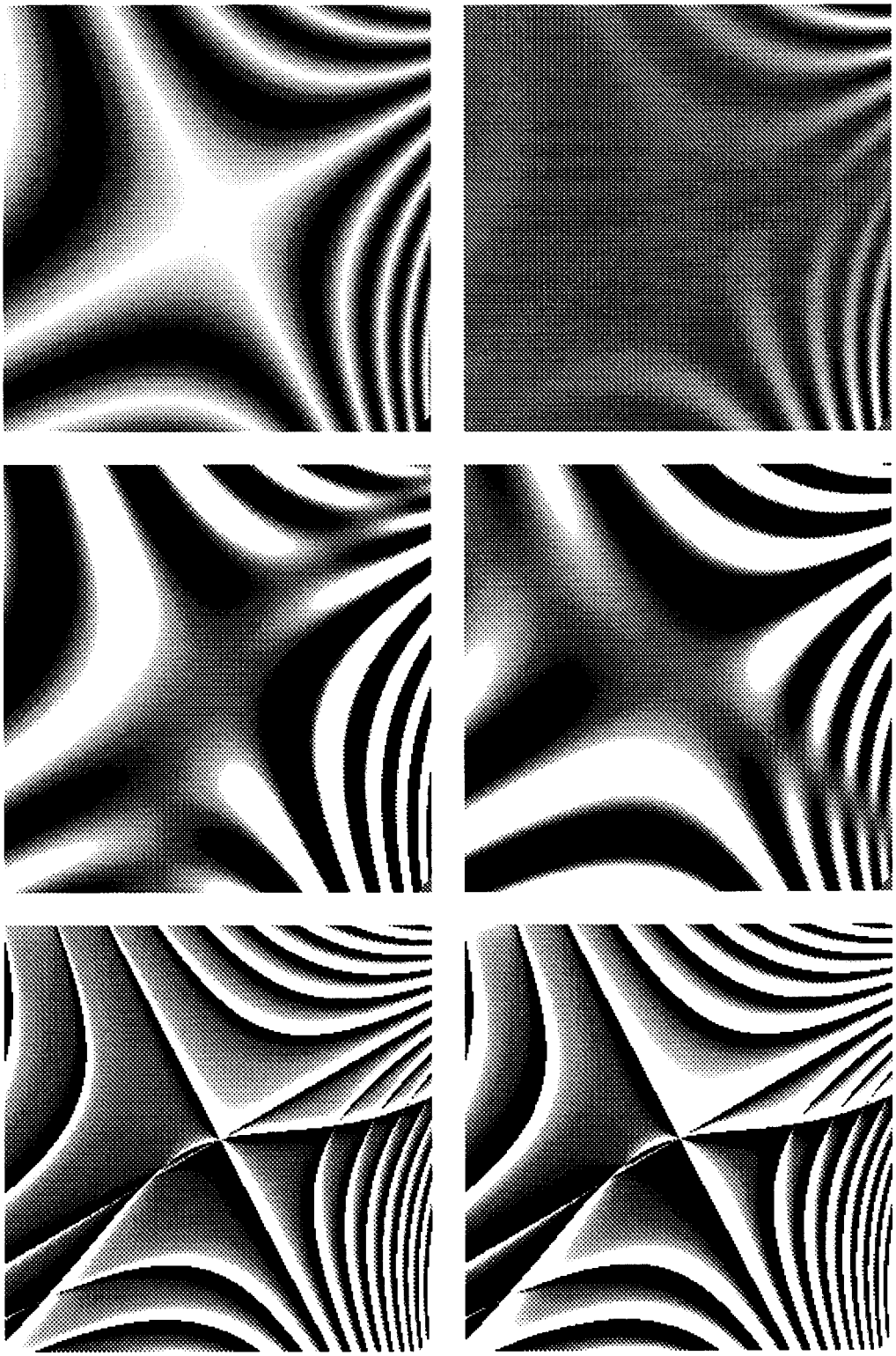

Figure 3.4: Multiplicative cosine grating on slanted plane. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (middle left and right) the x and y gradients, (e) (bottom left) the 10% δ–plot and (f) the 5% δ–plot.

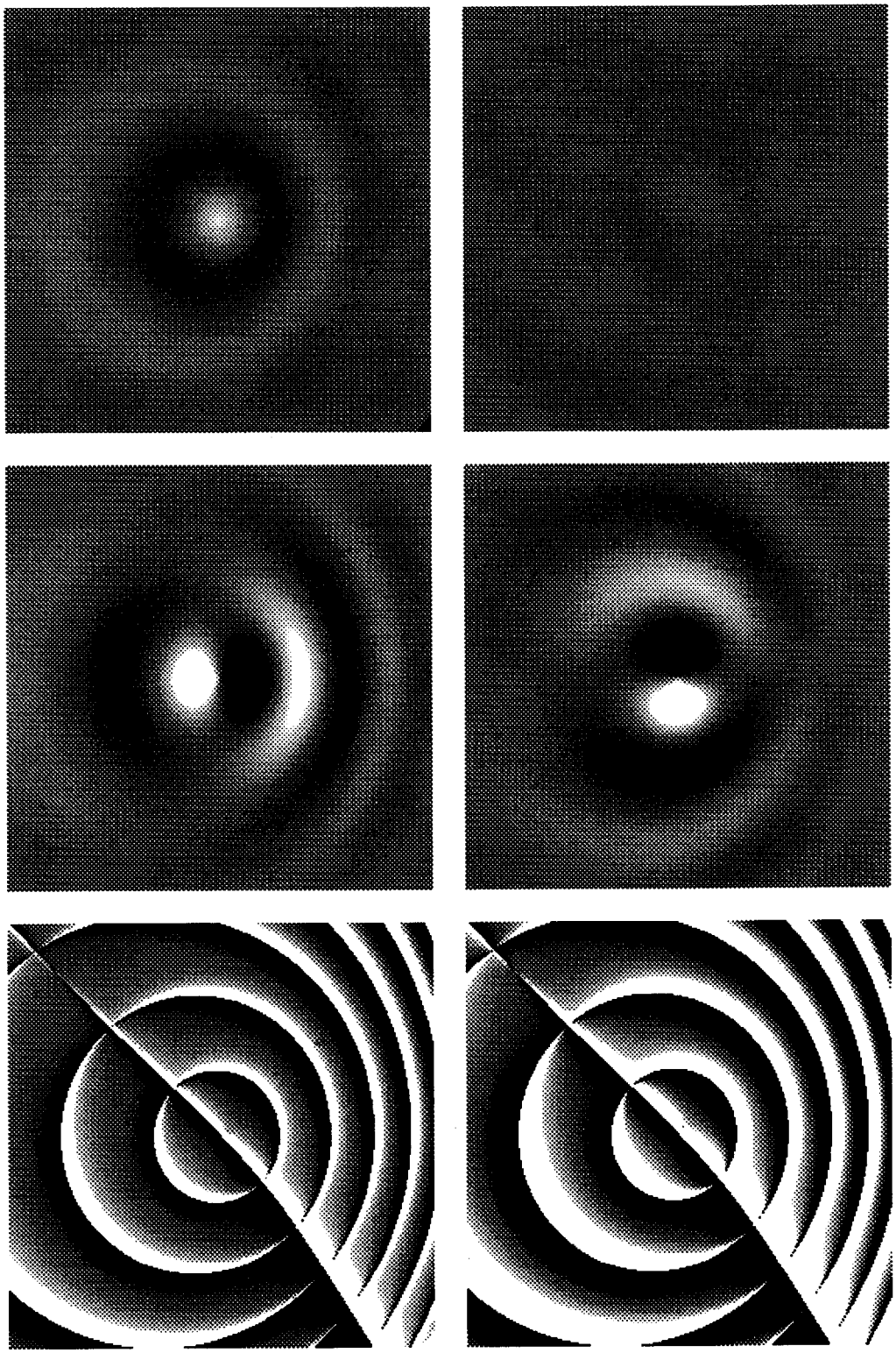

Figure 3.5: Exponentially damped cosine on slanted plane. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (middle left and right) the x and y gradients, (e) (bottom left) the 10% δ–plot and (f) the 5% δ–plot.

Chapter 4

Planar Patch Estimation

This chapter discusses the specific problems and forms of the general minimization equations, developed in chapter 2, when applied to the planar patch case. Specific implementations with various shading models are derived and results of the algorithms on real and synthetic data presented.

A planar patch is the easiest surface structure that can be considered but is still a fairly complex case to study in the context of the FICE for a general Lambertian shading model. The planar case is well studied and understood; Negahdaripour and Horn (1987) presented a closed–form solution in the general motion case. Their solution amounts to solving an eigenvalue problem in the case of the traditional constraint equation. However, in the FICE formulation, no closed–form solution exists; the parameter estimates are determined by a system of nonlinear vectorial equations that vary in complexity depending on the choice of the shading model and albedo function. Once the vectorial system is determined, a scalar system is obtained by projection of the vectorial equations onto various axes. Although, in theory, the solution of the original vectorial system does not depend on the choice of the projection axes, the algebraic complexity and numerical behavior of the resulting scalar system vary greatly; a proper choice of the projection axes is required to obtain accurate parameter estimates.

Several different implementations are derived for various shading and surface models; examples of parameter estimation are presented in the next sections.

4.1 Implementations of the Planar Patch Case

A planar patch is *exactly* represented by equation (2.23), $\frac{1}{Z} = \frac{(\mathbf{r}\cdot\mathbf{n})}{FZ_0}$, in the viewer frame, that is, the reciprocal of the depth is expressed as a linear functional in the image coordinates $\mathbf{r}$. In the FICE, the structure term, $\frac{(\mathbf{r}\cdot\mathbf{n})}{FZ_0}$, only appears in the product with the term $(\mathbf{s}\cdot\mathbf{t})$, resulting in a scale ambiguity between $\mathbf{n}$ and $\mathbf{t}$. In fact, the same solution to the problem is obtained if $\mathbf{n}$ is replaced by $k\mathbf{n}$ and $\mathbf{t}$ by $\mathbf{t}/k$, therefore, the translation $\mathbf{t}$ can only be recovered within a global scale factor, or in other words, only its direction can be determined. As a consequence, the explicit scale factor $1/FZ_0$ can be omitted from the equations without loss of generality. However it is required in the practical implementation with real data, where the focal distance of the camera F and the absolute distance Z_0 are known, to directly compare the estimated parameters with the known experimental parameters of the set–up. This scale ambiguity can be turned into an advantage.

Very powerful simplifications can be obtained in cases where the temporal shading variations only depend on the unit normal $\hat{\mathbf{n}}$ as opposed to *both* $\mathbf{n}$ and $\hat{\mathbf{n}}$. This distinction comes about because the shading equation usually only depends on the direction of the normal, i.e. $\hat{\mathbf{n}}$, while the reciprocal of the depth is expressed in terms of the full normal $\mathbf{n}$, and *both* the direction and the magnitude of the normal are relevant information. Therefore, the temporal derivative of the shading expression can depend on $\mathbf{n}$ and $\hat{\mathbf{n}}$ concurrently when the distance of the patch to the source or camera is involved (like the nearby source case). The normal $\mathbf{n}$ can be replaced by the unit normal $\hat{\mathbf{n}}$ in the term $(\mathbf{r}\cdot\mathbf{n})$, that arises from the expression of the reciprocal of the depth in terms of the surface coordinates, producing yet another scaling of the translation vector $\mathbf{t}$. This substitution results in a FICE that *only* depends on the unit normal, and the vectorial equation (2.28), obtained by differentiation of the minimization equation (2.25) in section 2.3.1.3, simplifies to

$$\iint_{\sigma} \left(\left(-(\mathbf{s}\cdot\mathbf{t})\frac{\partial\zeta}{\partial\hat{\mathbf{n}}} - \rho_\lambda \frac{\partial e_t}{\partial\hat{\mathbf{n}}} \right) \left(G(\boldsymbol{\omega},\mathbf{t},\mathbf{n}) - \rho_\lambda e() \right) - \mu(\mathbf{r})\rho_\lambda \frac{\partial e}{\partial\hat{\mathbf{n}}} \right) d\mathbf{r} + \lambda\hat{\mathbf{n}} = 0.$$

In order to fully specify the planar patch FICE, a specific shading model is required. Only Lambertian surface reflectances are considered in this thesis, as previously mentioned. The Lambertian shading model is determined by the type (punctual or extended) and position (nearby or distant) of the light source(s) and by the characteristics of the albedo function.

In this study, the albedo $\rho_\lambda(\alpha,\beta)$ is either constant, the surface is textureless and all the spatiotemporal variations of the irradiance function are due to shading, or continuously varying and the spatiotemporal variations are due to both shading and texture. More specifically, the spatiotemporal gradients of a textured surface have a component caused by the surface markings (texture) and a component that is due to the spatiotemporal variations of the shading on the surface.

4.1.1 Distant Punctual Source Illuminating a Lambertian Patch

The simplest Lambertian reflectance model that can be considered is the one with a distant punctual source and a constant albedo ρ_λ. It is easily seen that the irradiance (equation 3.5) is spatially constant and that the temporal rate of change of the irradiance (equation 3.4) is constant. Since the irradiance is spatially constant, the spatial gradient field $E_{\mathbf{r}}$ is zero and so are the vector fields $\mathbf{v}$ and $\mathbf{s}$. It is clear, without further equations, that the problem is grossly underconstrained and cannot be solved even by considering multiple frames.

Assuming a set of discrete constant patches leads to a similar problem. The problem is underconstrained within each patch, and the previous equations are not valid at the patch boundaries.

In order to obtain more information, in the distant source case, a Lambertian surface with a continuously varying albedo is required. The source strength L_0, which cannot be estimated, is set to unity and the albedo $\rho_\lambda(\alpha,\beta)$ is denoted by $\rho(\mathbf{r})$ to emphasize the fact that it is, indirectly, a function of the image coordinates. It should be noted that the notation $\rho(\mathbf{r})$ is deceptive because the albedo depends on the surface coordinates (α,β) and not on the image coordinates $\mathbf{r}$. However, (α,β) can be expressed (see section 3.3.1) in terms of the image coordinates $\mathbf{r}$ and vice–versa. This issue is not relevant in the two–frame case because texture does not need to be tracked there. The two–frame case can be exclusively specified by the spatiotemporal gradients of the irradiance. In practice, a lot of synthetic data used in the two–frame situation are produced by directly generating the spatial gradients and computing the temporal gradients directly from them by means of the CE. However, this approach failed in the case of multiple (more than two) frames because texture needs to be tracked in this case.

4.1.1.1 Solution Using the FICE

This section deals with the case of a planar Lambertian surface with a smoothly varying albedo and illuminated by a single punctual source of known direction with straight FICE used as a constraint equation. This case is solved by a minimization problem of type (b') (equation 2.24) with a unique parameter constraint on the unit normal $\hat{\mathbf{n}}$. This minimization is identical to the generic minimization of section 2.3.1.3, where the unit normal $\hat{\mathbf{n}}$ is substituted for the normal $\mathbf{n}$ because the temporal variations of the shading model only depend on $\hat{\mathbf{n}}$. The generic functions $\zeta(\mathbf{n})$, $e(\hat{\mathbf{L}}, \hat{\mathbf{n}})$ and $e_t(\hat{\mathbf{L}}, \hat{\mathbf{n}}, \boldsymbol{\omega}, \mathbf{t})$ are specialized to

$$\begin{cases} \zeta(\mathbf{n}) = (\mathbf{r} \cdot \hat{\mathbf{n}}) \\ e(\hat{\mathbf{L}}, \hat{\mathbf{n}}) = \rho(\mathbf{r})(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}}) \\ e_t(\hat{\mathbf{L}}, \hat{\mathbf{n}}, \boldsymbol{\omega}, \mathbf{t}) = \rho(\mathbf{r})[\hat{\mathbf{l}}, \boldsymbol{\omega}, \hat{\mathbf{n}}]. \end{cases}$$

If we let $F(\boldsymbol{\omega}, \mathbf{t}, \hat{\mathbf{n}}) = E_t - (\mathbf{v} \cdot \boldsymbol{\omega}) - (\mathbf{r} \cdot \hat{\mathbf{n}})(\mathbf{s} \cdot \mathbf{t}) - \rho(\mathbf{r})[\hat{\mathbf{l}}, \boldsymbol{\omega}, \hat{\mathbf{n}}]$, the unconstrained minimization equation 2.25 can be written in the form

$$\min \left(\iint_\sigma \left(F^2 + \mu(\mathbf{r})\big(E(\mathbf{r},t) - \rho(\mathbf{r})(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})\big) \right) d\mathbf{r} + \lambda\big(\|\hat{\mathbf{n}}\|^2 - 1\big) \right) \tag{4.1}$$

and the resulting vectorial system, obtained by differentiation of (4.1) with respect to the parameters $\mathbf{t}$, $\boldsymbol{\omega}$ and $\hat{\mathbf{n}}$, takes the form

$$\iint_\sigma F\big(-\mathbf{v} - \rho(\mathbf{r})(\hat{\mathbf{n}} \times \hat{\mathbf{l}})\big) d\mathbf{r} = 0 \tag{4.2}$$

$$\iint_\sigma F\big(-\mathbf{s}(\mathbf{r} \cdot \hat{\mathbf{n}})\big) d\mathbf{r} = 0 \tag{4.3}$$

$$\iint_\sigma \Big(F\big(-(\mathbf{s} \cdot \mathbf{t})\mathbf{r} - \rho(\mathbf{r})(\boldsymbol{\omega} \times \hat{\mathbf{l}})\big) - \mu(\mathbf{r})\rho(\mathbf{r})\hat{\mathbf{l}} \Big) d\mathbf{r} + \lambda \hat{\mathbf{n}} = 0. \tag{4.4}$$

The Lagrange multiplier λ and the Lagrange multiplier function $\mu(\mathbf{r})$ can be eliminated by taking the dot product of equation (4.4) with the vectors $\hat{\mathbf{n}}$ and $\hat{\mathbf{l}}$ and by solving the resulting scalar linear system in the unknowns λ and $\iint_\sigma \mu(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$. If we let $\mathbf{l}_\perp^{\hat{\mathbf{n}}} = \dfrac{\hat{\mathbf{n}} \times (\hat{\mathbf{l}} \times \hat{\mathbf{n}})}{\|\hat{\mathbf{l}} \times \hat{\mathbf{n}}\|^2}$ and $\mathbf{n}_\perp^{\hat{\mathbf{l}}} = \dfrac{(\hat{\mathbf{l}} \times \hat{\mathbf{n}}) \times \hat{\mathbf{l}}}{\|\hat{\mathbf{l}} \times \hat{\mathbf{n}}\|^2}$, the Lagrange multipliers are given by

$$\iint_\sigma \mu(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} = -\iint_\sigma F\Big(\mathbf{l}_\perp^{\hat{\mathbf{n}}} \cdot \big((\mathbf{s} \cdot \mathbf{t})\mathbf{r} - \rho(\mathbf{r})(\boldsymbol{\omega} \times \hat{\mathbf{l}})\big)\Big) d\mathbf{r}$$

$$\lambda = -\iint_{\sigma} F\Big(\mathbf{n}_{\perp}^{\hat{\mathbf{l}}} \cdot \big((\mathbf{s}\cdot\mathbf{t})\mathbf{r} - \rho(\mathbf{r})(\boldsymbol{\omega}\times\hat{\mathbf{l}})\big)\Big)d\mathbf{r}$$

and can be eliminated from (4.4) to yield (if $(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}) \neq 0$)

$$\iint_{\sigma} F\big(-(\mathbf{s}\cdot\mathbf{t})(\mathbf{I}_3 - \mathbf{l}_{\perp}^{\hat{\mathbf{n}}}\hat{\mathbf{l}}^T)\mathbf{r} + \rho(\mathbf{r})(\mathbf{I}_3 - \mathbf{n}_{\perp}^{\hat{\mathbf{l}}}\hat{\mathbf{n}}^T)(\boldsymbol{\omega}\times\hat{\mathbf{l}})\big)d\mathbf{r}, \tag{4.5}$$

where the albedo $\rho(\mathbf{r})$ is expressed in term of the irradiance data by means of the shading equation, $\rho(\mathbf{r}) = E(\mathbf{r},t)/(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})$.

Equations (4.2), (4.3) and (4.5) form a system $\mathcal{S}$ of three nonlinear vectorial equations and can be solved by any of the methods described in section 2.3.2, once it has been projected onto suitable axes. An alternate, more efficient, way of solving $\mathcal{S}$ is to notice that the first two equations (4.2) and (4.3) are linear in $\boldsymbol{\omega}$ and $\mathbf{t}$; that is, the system $\mathcal{S}$ is in fact semilinear and can be broken into a linear system $\mathcal{L}$ formed by the first two equations and a nonlinear equation $\mathcal{N}$ (4.5) in $\hat{\mathbf{n}}$. This semilinear system can be solved iteratively: $\mathcal{L}$ is solved using the current estimates of $\hat{\mathbf{n}}$, then $\mathcal{N}$ is solved using the estimates of $\boldsymbol{\omega}$ and $\mathbf{t}$. The iteration is initially started with an arbitrary value for $\hat{\mathbf{n}}$. The advantages of this method, as opposed to a general global nonlinear method, are the speed, the improved convergence performance and its elegance. The increase of speed is partly due to the fact that two of the vectorial variables are solved by a linear system and partly due to the fact that we are dealing with a nonlinear equation as opposed to a system of nonlinear equations. The improvement in convergence is caused by the lower dimensionality of the nonlinear part of the system. Experiments suggest that the convergence of the higher dimensional nonlinear system is far worse than the semilinear method. Unfortunately, the convergence of the semilinear iterative implementation is difficult to prove or support, even intuitively. The elegance results from the ability to *directly* solve the vectorial linear system without need of projecting it onto a specific axis. $\mathcal{L}$ can be written in the matrix form

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} \tag{4.6}$$

where

$$
\begin{aligned}
\mathbf{M}_1 &= \iint_\sigma \Big[(\mathbf{v}+\rho(\mathbf{r})(\hat{\mathbf{n}}\times\hat{\mathbf{l}}))\,(\mathbf{v}+\rho(\mathbf{r})(\hat{\mathbf{n}}\times\hat{\mathbf{l}}))^T\Big]\,d\mathbf{r} \\
&= \iint_\sigma \Big[(\mathbf{v}\mathbf{v}^T)+\rho(\mathbf{r})\Big((\mathbf{v}(\hat{\mathbf{n}}\times\hat{\mathbf{l}})^T)+((\hat{\mathbf{n}}\times\hat{\mathbf{l}})\mathbf{v}^T)\Big)+\rho(\mathbf{r})^2((\hat{\mathbf{n}}\times\hat{\mathbf{l}})(\hat{\mathbf{n}}\times\hat{\mathbf{l}})^T)\Big]\,d\mathbf{r} \\
\mathbf{M}_2 &= \iint_\sigma (\mathbf{r}\cdot\hat{\mathbf{n}})\Big((\mathbf{v}+\rho(\mathbf{r})(\hat{\mathbf{n}}\times\hat{\mathbf{l}}))\mathbf{s}^T\Big)d\mathbf{r} \\
\mathbf{M}_4 &= \iint_\sigma (\mathbf{r}\cdot\hat{\mathbf{n}})^2(\mathbf{s}\mathbf{s}^T)d\mathbf{r} \\
\mathbf{e}_1 &= \iint_\sigma E_t\big(\mathbf{v}+\rho(\mathbf{r})(\hat{\mathbf{n}}\times\hat{\mathbf{l}})\big)d\mathbf{r} \\
\mathbf{e}_2 &= \iint_\sigma E_t(\mathbf{r}\cdot\hat{\mathbf{n}})\,\mathbf{s}\,d\mathbf{r}
\end{aligned}
$$

and the solution to the system (4.6) is given by

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

In order to numerically implement the nonlinear vectorial equation (4.5), it needs to be broken down into scalar components. It is possible that the eigenvectors of the two matrices $\mathbf{M}_{\hat{\mathbf{l}}}^T=(\mathbf{l}_\perp^{\hat{\mathbf{n}}}\hat{\mathbf{l}}^T-\mathbf{I}_3)^T$ and $\mathbf{M}_{\hat{\mathbf{n}}}^T=(\mathbf{n}_\perp^{\hat{\mathbf{l}}}\hat{\mathbf{n}}^T-\mathbf{I}_3)^T$ are advantageous projection axes. The eigenvectors of the matrix $\mathbf{M}_{\hat{\mathbf{l}}}^T$ are $\hat{\mathbf{l}}$, associated with the single eigenvalue 0, and any two linearly independent vectors orthogonal to $\mathbf{l}_\perp^{\hat{\mathbf{n}}}$, associated with the double eigenvalue 1. The eigenvectors of the matrix $\mathbf{M}_{\hat{\mathbf{n}}}^T$ are $\hat{\mathbf{n}}$, associated with the single eigenvalue 0, and any two linearly independent vectors orthogonal to $\mathbf{n}_\perp^{\hat{\mathbf{l}}}$ associated with the double eigenvalue 1. The eigenvectors $\hat{\mathbf{n}}$ and $\hat{\mathbf{l}}$ are unsuitable because the dot product of $\hat{\mathbf{n}}$ with the nonlinear equation (4.5) produces a scalar equation linearly dependent on the equation (4.3) of the system $\mathcal{S}$, while the dot product with $\hat{\mathbf{l}}$ produces a null equation. However, the use of the common eigenvector $(\hat{\mathbf{l}}\times\hat{\mathbf{n}})$ of $\mathbf{M}_{\hat{\mathbf{l}}}^T$ and $\mathbf{M}_{\hat{\mathbf{n}}}^T$ results in a simple scalar equation. In our numerical implementation, the two other scalar equations were obtained by projection onto the basis $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$–axis.

At first glance, it might appear that solving the different equations and the linear system is very costly, since the integrations over the whole image I need to be performed at each iteration. Appendix F shows that most of the components of the vectors and matrices in the previous equations can be precomputed resulting in a very efficient numerical implementation.

4.1.1.2 Solution Using the Dynamical Frame Unwarping FICE

The previous section assumed that the simple FICE was used in the minimization procedure. In practice however, the DFU incremental FICE is used, and the previous equations need to be slightly altered. More specifically, the temporal irradiance gradients are replaced by the displaced frame difference, $D_E(\mathbf{r}, t, \tau, \boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)})$, defined in section 2.3.3, the vector fields $\mathbf{v}$ and $\mathbf{s}$ now depend on the current estimate of the motion and, finally, the motion parameters are replaced by the incremental motion parameters. Under this framework, the linear system $\mathcal{L}$ (4.6) becomes

$$\begin{pmatrix} \mathbf{M}_1(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) & \mathbf{M}_2(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) \\ \mathbf{M}_2^T(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) & \mathbf{M}_4(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} - \boldsymbol{\omega}^{(n)} \\ \mathbf{t} - \mathbf{t}^{(n)} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_1(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) \\ \mathbf{d}_2(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) \end{pmatrix} \tag{4.7}$$

where

$$\begin{aligned} \mathbf{d}_1(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) &= \iint_\sigma D_E(\mathbf{r}, t, \tau, \boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) \big(\mathbf{v}(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) + \rho(\mathbf{r})(\hat{\mathbf{n}} \times \hat{\mathbf{l}})\big) d\mathbf{r} \\ \mathbf{d}_2(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) &= \iint_\sigma D_E(\mathbf{r}, t, \tau, \boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)}) (\mathbf{r} \cdot \hat{\mathbf{n}})\, \mathbf{s}(\boldsymbol{\omega}^{(n)}, \mathbf{t}^{(n)})\, d\mathbf{r}, \end{aligned}$$

and the system (4.7) is solved for $\boldsymbol{\omega}^{(n+1)}$ and $\mathbf{t}^{(n+1)}$, the $(n+1)^{th}$ estimates of the translation $\mathbf{t}$ and rotation $\boldsymbol{\omega}$.

Similar adjustments of variables and parameters are performed on the nonlinear equation (4.5) or on the overall system $\mathcal{S}$ when considered globally as a system of nonlinear equations. In practice, the straight FICE implementation is identical to the DFU implementation, where the extra iterative loop, which computes the incremental motion parameter, is disabled and the field unwarping step suppressed, resulting in motion independent vector fields $\mathbf{v}$ and $\mathbf{s}$.

4.1.2 General Punctual Light Source Illuminating a Lambertian Patch

The previous section dealt with the case where there were no shading variations induced by the proximity of the light source. The advantage of the previous case is its simplicity; its drawback is its inability to deal with textureless surfaces, because in the distant case, the irradiance and the temporal variations of the irradiance on the surface are constant. This section examines the use of the general Lambertian shading model (3.1), presented in section 3.1, in the FICE. The complexity of the temporal variations of the shading model (equation (3.2) in the general

case and equation (3.3) in the first–order Lambertian model) is far greater than in the distant case and is mostly due to the presence of the structure term $\mathbf{R} = \frac{Z_0\mathbf{r}}{F(\mathbf{r}\cdot\mathbf{n})}$, which depends on $\mathbf{n}$, in $\frac{dE}{dt}$. Due to its complexity, the full model is more appropriate in the case of textureless surfaces (ρ constant) and/or in the case of special motions like pure translation or rotation, than in the general motion and texture case.

In order to slightly simplify the presentation of the general case, the equations shown in this section are derived for the textureless surface case. In this instance, a minimization with a single constraint on the parameter $\hat{\mathbf{n}}$ is performed and the vectorial equations only contain a single Lagrange multiplier λ associated with the parameter constraint $\|\hat{\mathbf{n}}\|^2 = 1$. The general texture case would be treated in a fashion similar to the distant case of section 4.1.1, and the Lagrange multipliers eliminated in an identical way.

4.1.2.1 General Model

Let $F(\boldsymbol{\omega},\mathbf{t},\hat{\mathbf{n}}) = E_t - (\mathbf{v}\cdot\boldsymbol{\omega}) - (\mathbf{r}\cdot\hat{\mathbf{n}})(\mathbf{s}\cdot\mathbf{t}) - \dot{E}_{gal}$ where $\dot{E}_{gal}$ represents the temporal variations of the general Lambertian case, i.e.

$$\dot{E}_{gal} = \frac{\rho}{\|\mathbf{L}\|}\Big([\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}] + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}] - (\mathbf{t}_\perp^{\hat{\mathbf{L}}}\cdot\hat{\mathbf{n}})\Big).$$

The unconstrained minimization equation can be written in the form

$$\mathcal{E} = \min_{\sigma}\left(\iint_\sigma F^2 d\mathbf{r} + \lambda\big(\|\hat{\mathbf{n}}\|^2 - 1\big)\right),$$

and the resulting vectorial equations take the form

$$\iint_\sigma F\Big(-\mathbf{v} - \frac{\rho}{\|\mathbf{L}\|}\big((\hat{\mathbf{l}}\times\hat{\mathbf{n}}) + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})(\hat{\mathbf{L}}\times\mathbf{R})\big)\Big)d\mathbf{r} = 0 \tag{4.8}$$

$$\iint_\sigma F\Big(-\mathbf{s}(\mathbf{r}\cdot\hat{\mathbf{n}}) - \frac{\rho}{\|\mathbf{L}\|}(\mathbf{I}_3 - \hat{\mathbf{L}}\hat{\mathbf{L}}^T)\hat{\mathbf{n}}\Big)d\mathbf{r} = 0 \tag{4.9}$$

$$\iint_\sigma F\Big(-(\mathbf{s}\cdot\mathbf{t})\mathbf{r} + \frac{\partial\dot{E}_{gal}}{\partial\hat{\mathbf{n}}}\Big)d\mathbf{r} + \lambda\hat{\mathbf{n}} = 0 \tag{4.10}$$

where

$$\frac{\partial\dot{E}_{gal}}{\partial\hat{\mathbf{n}}} = \frac{\dot{E}}{\|\mathbf{L}\|^2}\left(\frac{\partial\mathbf{R}}{\partial\hat{\mathbf{n}}}\right)^T\mathbf{L} + \frac{\rho}{\|\mathbf{L}\|}\Big((\hat{\mathbf{l}}\times\boldsymbol{\omega}) + \hat{\mathbf{L}}[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}] + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})\left(\frac{\partial\mathbf{R}}{\partial\hat{\mathbf{n}}}\right)^T(\hat{\mathbf{L}}\times\boldsymbol{\omega}) - \mathbf{t}_\perp^{\hat{\mathbf{L}}}\Big)$$

with $\frac{\partial \mathbf{R}}{\partial \hat{\mathbf{n}}} = \frac{-1}{(\mathbf{r}\cdot\mathbf{n})^2}\mathbf{r}\mathbf{r}^T\mathbf{N}_{\hat{\mathbf{n}}}$ and $\mathbf{N}_{\hat{\mathbf{n}}} = \left(\frac{\partial \mathbf{n}}{\partial \hat{\mathbf{n}}}\right)^T = \frac{1}{\hat{\mathbf{n}}\cdot\hat{\mathbf{z}}}\left(\mathbf{I}_3 - \frac{\hat{\mathbf{z}}\hat{\mathbf{n}}^T}{(\hat{\mathbf{n}}\cdot\hat{\mathbf{z}})}\right)$.

It is noteworthy that the above system is also semilinear in $\mathbf{t}$ and $\boldsymbol{\omega}$ and can be solved with a method similar to the one described in section 4.1.1 and the linear portion of the system is only sightly more complicated than its distant counterpart. On the other hand, the nonlinear equation is far more difficult to implement numerically in the general case. More robust numerical implementations are achieved in special motion cases, as in the pure translation case ($\boldsymbol{\omega} = 0$), where the nonlinear equation (4.10) simplifies to

$$\iint_\sigma F\left(-(\mathbf{s}\cdot\mathbf{t})\mathbf{r} - \frac{\rho}{\|\mathbf{L}\|}\left(\mathbf{I}_3 + \left(\frac{\partial \mathbf{R}}{\partial \hat{\mathbf{n}}}\right)^T \frac{\mathbf{L}}{\|\mathbf{L}\|^2}\hat{\mathbf{n}}^T\right)\mathbf{t}_\perp^{\hat{\mathbf{L}}}\right) d\mathbf{r} + \lambda\hat{\mathbf{n}} = 0. \tag{4.11}$$

The general case described in this section is only relevant when the light source is very close to the surface relative to distance of the plane to the lens. Very often the light source is further away, and it can be assumed that $\|\mathbf{R}\|/\|\mathbf{l}\| \ll 1$. The relative proximity of the light source still induces a variable shading across the plane surface but the overall implementation is simpler as we will see in the next section.

4.1.2.2 First–Order Model

Under the assumption that the relative distance from light source to object and from object to camera is small, the shading equation and the temporal variations of shading equation can be approximated by a first–order Taylor series with respect to $\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}$ (see section 3.1.2). For a planar patch, the shading equation takes the form

$$E(\mathbf{r},t) = \rho\left((\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}) - \frac{Z_0}{F(\mathbf{r}\cdot\mathbf{n})}\Big((\mathbf{r} - (\mathbf{r}\cdot\hat{\mathbf{l}})\hat{\mathbf{l}})\cdot\hat{\mathbf{n}}\Big)\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right),$$

i.e. the distant uniform shading is modulated by a first–order, in $\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}$, shading variation across the planar surface. The temporal variations $\dot{E}_{first}$ are computed by equation (3.3) and can be rewritten, in the planar patch case, in the form

$$\dot{E}_{first} = \rho Z_0\left([\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}] + \frac{Z_0(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})}{F|\mathbf{r}\cdot\mathbf{n}|}[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{r}}]\frac{\|\mathbf{r}\|}{\|\mathbf{l}\|} - (\mathbf{t}_\perp^{\hat{\mathbf{l}}}\cdot\hat{\mathbf{n}})\frac{1}{\|\mathbf{l}\|}\right).$$

If we let $F = E_t - (\mathbf{v}\cdot\boldsymbol{\omega}) - (\mathbf{r}\cdot\hat{\mathbf{n}})(\mathbf{s}\cdot\mathbf{t}) - \dot{E}_{first}$, the resulting vectorial equations, which define the parameters $\mathbf{t}$, $\boldsymbol{\omega}$ and $\mathbf{n}$, are, for a textureless surface,

$$\iint_\sigma F\left(-\mathbf{v} - \rho\Big(\hat{\mathbf{l}}\times\hat{\mathbf{n}} + \frac{Z_0(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}})}{F|\mathbf{r}\cdot\mathbf{n}|}(\hat{\mathbf{L}}\times\hat{\mathbf{r}})\frac{\|\mathbf{r}\|}{\|\mathbf{l}\|}\Big)\right)d\mathbf{r} = 0$$

$$\iint_\sigma F\Big(-\mathbf{s}(\mathbf{r}\cdot\hat{\mathbf{n}}) - \rho(\mathbf{I}_3 - \hat{\mathbf{l}}\hat{\mathbf{l}}^T)\frac{\hat{\mathbf{n}}}{\|\mathbf{l}\|}\Big)d\mathbf{r} = 0$$

$$\iint_\sigma F\Big(-(\mathbf{s}\cdot\mathbf{t})\mathbf{r} + \frac{\partial\dot{E}_{first}}{\partial\hat{\mathbf{n}}}\Big)d\mathbf{r} + \lambda\hat{\mathbf{n}} = 0$$

where

$$\frac{\partial\dot{E}_{first}}{\partial\hat{\mathbf{n}}} = \rho\left((\hat{\mathbf{l}}\times\boldsymbol{\omega}) + \frac{\|\mathbf{r}\|\,[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{r}}]}{\|\mathbf{l}\|\,|\mathbf{r}\cdot\mathbf{n}|}\Big(\mathbf{I}_3 - \frac{\|\mathbf{r}\|}{\|\mathbf{l}\|}\mathbf{N}_{\hat{\mathbf{n}}}\frac{\mathbf{r}\hat{\mathbf{n}}^T}{(\mathbf{r}\cdot\mathbf{n})}\Big)\hat{\mathbf{l}} - \frac{\mathbf{t}_\perp^{\hat{\mathbf{l}}}}{\|\mathbf{l}\|}\right).$$

These equations are numerically more stable than the equivalent general shading equations of the previous section. Moreover, they represent a reasonable approximation of the general equations for ratio $\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}$ up to about 10% and should be preferred if the speed required for solving the equations is more important than the accuracy of the solution.

4.1.3 Attenuated Lambertian Model

The last model presented is the attenuated Lambertian model under the assumption that the light source and the viewer are coincident. This model was developed in section 3.2 for the case of underwater photography with the light source mounted on the camera. It is described by the shading equation

$$E(\mathbf{r},t) = \frac{\rho(\alpha,\beta)}{\|\mathbf{L}\|^2}(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}}) = -\frac{\rho(\alpha,\beta)}{\|\mathbf{R}\|^2}(\hat{\mathbf{r}}\cdot\hat{\mathbf{n}});$$

its temporal variations of shading, expressed by equation (3.9) can be rewritten in the form

$$\dot{E}_{att} = \frac{\rho Z_0^3}{F^3(\mathbf{r}\cdot\mathbf{n})^3}\big((\mathbf{I}_3 - 3\hat{\mathbf{r}}\hat{\mathbf{r}}^T)\mathbf{t}\cdot\hat{\mathbf{n}}\big).$$

If we let $F = E_t - (\mathbf{v}\cdot\boldsymbol{\omega}) - (\mathbf{r}\cdot\hat{\mathbf{n}})(\mathbf{s}\cdot\mathbf{t}) - \dot{E}_{att}$, the system of vectorial equations in the parameters $\mathbf{t}$, $\boldsymbol{\omega}$ and $\mathbf{n}$, for a textureless surface, is simply given by

$$\iint_\sigma F\mathbf{v}d\mathbf{r} = 0 \qquad (4.12)$$

$$\iint_\sigma F\Big(-\mathbf{s}(\mathbf{r}\cdot\hat{\mathbf{n}}) - \rho\frac{Z_0^2(\mathbf{r}\cdot\mathbf{n})^3}{F^2\|\mathbf{r}\|^3}(\mathbf{I}_3 - 3\hat{\mathbf{r}}\hat{\mathbf{r}}^T)\hat{\mathbf{n}}\Big)d\mathbf{r} = 0 \qquad (4.13)$$

$$\iint_\sigma F\left(-(\mathbf{s}\cdot\mathbf{t})\mathbf{r} - \frac{\rho(\mathbf{r}\cdot\mathbf{n})^3}{\|\mathbf{r}\|^3}\left(\mathbf{I}_3 - 3\mathbf{N}_{\hat{\mathbf{n}}}\frac{\mathbf{r}\hat{\mathbf{n}}^T}{\mathbf{r}\cdot\hat{\mathbf{n}}}\right)(\mathbf{I}_3 - 3\hat{\mathbf{r}}\hat{\mathbf{r}}^T)\mathbf{t}\right)d\mathbf{r} + \lambda\hat{\mathbf{n}} = 0. \qquad (4.14)$$

4.1.4 Conclusions on the Planar Implementations

The previous sections presented several implementations of the rigid body motion and structure recovery problem with shading in the planar patch case. The vectorial systems of equations defining the motion parameters $\mathbf{t}$, $\boldsymbol{\omega}$ and the structure parameter $\hat{\mathbf{n}}$ were derived for several shading models, different types of albedo functions and for the regular and dynamically unwarping frame FICE. In most instances, the equations were given in the simpler case of a textureless surface and used the FICE as a constraint equation. Section 4.1.1 showed how to derive the relevant equations when the shading equation is used as a constraint in the general texture case. The method used in the distant case is general, and only the form and complexity of the Lagrange multipliers change in the other cases. Section 4.1.1.2 explained how to modify the equations derived with the FICE, in order to obtain the equations under the DFU case. The modifications amount to introducing an extra iterative loop for the computation of the incremental motion parameters and to updating the now variable vector fields $\mathbf{s}$ and $\mathbf{v}$. No attempt was made to present the scalar equations that were actually numerically implemented because this operation is very tedious and no additional information is gained in displaying the scalar equations.

The next section presents a set of examples that illustrates the performance of the proposed algorithms, compares the results to those given by a classical implementation, when appropriate, and explains the numerical difficulties present in the implementations.

4.2 Examples

The examples presented in this section attempt to demonstrate the performance of the algorithms in various situations with both synthetic and real data. Synthetic data are used in most examples because, in many cases, the experiments could not be simulated, easily and reliably, with real data, and synthetic data provide a way of evaluating the accuracy of the estimates by comparing them to the true values. Synthetic data were produced by ray–tracing an analytical texture function on a moving plane. This method allows perfect control of all the lighting, structure and motion parameters and produces high quality images at various resolutions. The irradiance data were quantized to eight bits and only the quantized data were used in most of the

experiments where the algorithm estimates the spatiotemporal gradients from these irradiance data.

Real data acquisition is very delicate and it was extremely hard, given the equipment and experimental set–up available, to control lighting and to measure accurately the motion parameters and camera parameters such as focal distance. Nevertheless, real data were used in the experiments because they were the driving force behind the development of the multiple–frame DFU algorithms which allow the estimation of the rigid body motion and the structure directly from the irradiance data obtained from a video camera. However, the structure and motion estimates that are obtained are difficult to interpret, as far as accuracy is concerned, because the real parameters are only approximatively known.

Real data are used in the distant Lambertian case, while synthetic data are produced in the distant, nearby and attenuated Lambertian case for a general albedo function and textureless surfaces. The use of synthetic data in the nearby punctual source and the attenuated cases is a necessity because no real, meaningful data could be obtained in these instances. Most of the examples presented in this section use the DFU formulation of the FICE, and multiple frames are used in the real data cases. In this section, the only goal of the experiments with real data is to obtain the best estimates possible without regard to the specifics of the multiframe algorithm. The discussion of the performance of the algorithms, as a function of the number of frames used, is delayed until chapter 6.

The examples shown in this section demonstrate the superiority of the FICE with respect to the CE in two ways: more accurate results are obtained in the case of textured surfaces and the FICE formulation is able to solve the case of textureless surface, where all the spatiotemporal variations are due to the shading, unlike the CE formulation that cannot deal with weakly marked or textureless surfaces. However, these new, improved results have a higher computational cost than those obtained by the less accurate and general CE formulation.

4.2.1 Distant Source

The distant, or hemispherical, source[1] illuminating a Lambertian surface case represents the most basic and fundamental situation. Under the assumption that the surface being imaged

[1]Distant and hemispherical sources are not *equivalent* but have *similar* behaviors as it was explained in section 3.2.

has a reflectance that is a good approximation of a Lambertian reflectance function, this case corresponds to the generic video recording situation. As such, the images can be processed, to compare the performances, with the algorithms described in this thesis and with conventional algorithms that use the classical constraint equation.

The closest, comparable classical algorithm is the direct motion algorithm for planar surfaces of Negahdaripour and Horn (1987). A direct, fair comparison with the experiments published in the cited paper is unfortunately impossible for two reasons: they did not specify the parameters of the multiplicative sinusoidal patterns used in the experiments and the image gradients are computed analytically *assuming* that the constraint equation is satisfied. Under these conditions, the motion and structure parameters are recovered perfectly and their algorithm displays perfect performance. Unfortunately, these data are unusable in the general case, since they assume, a priori, that the CE holds and discard, by construction, all shading variations information. No experimental results are provided in the situations where the spatiotemporal gradients are estimated directly from an irradiance pair of images (synthetic or real), and the motion and structure parameters are computed from these gradient fields. Since no method is provided to compute the required gradients from the irradiance images, the FICE implementation is used in *both* the CE and the FICE cases when comparing the performance of the two algorithms in the presence of shading induced variations. When used in a CE mode, the FICE implementation estimates all the gradients using the DFU method, but the temporal shading variations are assumed to be zero. In this way, the best possible spatiotemporal gradients are used in the two implementations and the difference in performance is only due to the additional use of the shading information in the FICE implementation.

Three types of experiments were run for the distant source case. The first experiment uses purely synthetic gradient data and demonstrates the ability of the system to correctly determine the solution under idealized conditions. The data of the first experiment are also used in the CE case, to show the bias that is obtained in the parameter estimates when the temporal shading variations are neglected. The second experiment is similar to the first one but uses synthetic, 8 bit quantized irradiance data, from which the spatiotemporal gradients are estimated. The third experiment uses real data.

4.2.1.1 Synthetic Data

The first experiment requires exact synthetic irradiance gradient data in order to evaluate the accuracy and the sensitivity of the nonlinear methods with respect to the initial estimates in the FICE case. The data are generated by analytically computing the irradiance spatial gradients in the image plane from the gradients of an analytical texture function $\rho(\alpha,\beta)$ on a mapping plane and from the position of a plane in space using the Jacobian of the transformation that maps the original mapping plane to the image plane (see section 3.3.1 for the relevant equations). The temporal gradients are computed directly from the FICE equation given the spatial gradients, the motion and structure parameters and the light source direction, that is,

$$E_t(\mathbf{x},t) = -\big(\mathbf{v}(\mathbf{r},t)\cdot\boldsymbol{\omega}\big) - (\mathbf{r}\cdot\hat{\mathbf{n}})\big(\mathbf{s}(\mathbf{r},t)\cdot\mathbf{t}\big) + \rho(\mathbf{r},t)[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}]. \tag{4.15}$$

The irradiance data $E(\mathbf{r},t)$ are also computed analytically since they are required in the FICE formulation when the shading equation, which relates the irradiance values to the value of the texture on the surface, is used as a constraint. Figure 4.1(a) shows the irradiance data, figure 4.1(b), the temporal gradients and figures 4.1(c,d), the spatial gradients (E_x and E_y respectively). The texture used in the previous example is a sinusoidal grating with a sinusoidal phase variation. More specifically, the texture is specified by the expression

$$\begin{cases} E(x,t) = \sin\Big(\omega_x x + \Big(\frac{\pi}{2}\sin\big(\frac{\omega_x}{2}y\big)\Big)\Big) \\ E(y,t) = \sin\Big(\omega_y y + \Big(\frac{\pi}{2}\sin\big(\frac{\omega_y}{2}x\big)\Big)\Big) \end{cases}.$$

In the first phase of the experiment, the analytically computed fields $E(\mathbf{r},t)$, $\nabla_{\mathbf{r}}E(\mathbf{r},t)$ and E_t are used directly as an input to a two–frame, non–DFU implementation of the FICE algorithm, and the structure and motion parameters evaluated with various numerical implementations. The purposes of the experiment are to validate a given numerical implementation, to analyze its performance in terms of the initial estimate and number of iterations, and to demonstrate the validity of the algorithm, i.e. its ability to recover perfectly the parameters under these idealized conditions. Five numerical implementations were tried, three semilinear where the nonlinear equation was implemented with a minimization method (Levenberg–Marquardt), a direct method (Powell's hybrid) and an homotopy method, and two globally nonlinear methods: Levenberg–Marquardt and Powell's hybrid. In general, the globally nonlinear methods

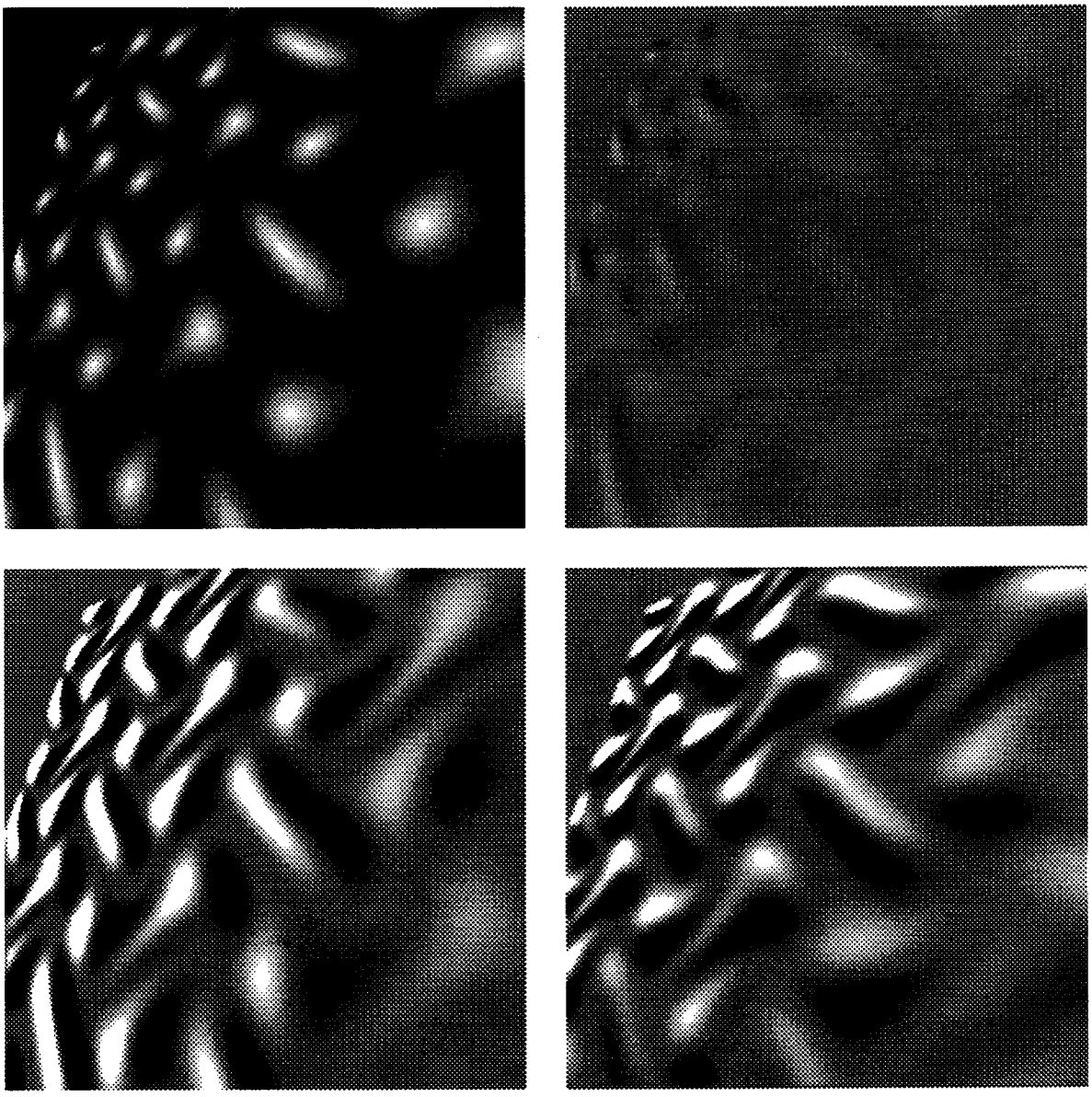

Figure 4.1: Cosine grating with sinusoidal phase variations on a slanted plane. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (middle left and right) the x– and y–gradients.

failed to converge to the correct solution or simply failed to converge for an arbitrary initial estimate of all the parameters, unless the initial estimate was close enough to the correct solution, in which case the two methods perform equally well and converge to the correct solution within the specified tolerance (usually 10^{-6} or 10^{-8}). To insure convergence to the correct solution, the globally nonlinear methods require a fairly accurate estimate of *all* the parameters. Such an estimate can be obtained by the simpler CE implementation for an arbitrary surface orientation and rigid motion, or for specific initial values of the normal to the plane, the translation and the rotation. The initial estimate of the normal can be set to (0 0 1) (frontal plane) and zero for the translation and rotation vectors in situations where the slant and tilt of the plane is small (less than 10°) and the motion is small (less than .5–.8° for the elementary rotations and less than .5 to 1 pixel translation in either directions).

The semilinear methods are somewhat more forgiving as far as the initial estimate is concerned but are surprisingly slow due to the huge number of iterations required to converge to the correct solution. Up to 1000 iterations may be required to converge for a moderately tight tolerance δ of 10^{-6}, i.e. the relative variation between two successive estimates is less than δ. A random initial value for the normal resulted in the convergence to the wrong solution in a few cases, although an initial estimate of (0 0 1) (frontal plane) never failed to converge to the right solution and was chosen as the default initial value unless a more specific value was determined by other means. The homotopy method converges to the correct estimates in far fewer iterations, but the saving in number of iterations is somewhat offset by the cost of each iteration (10 to 20 times the cost of a typical nonlinear implementation using Levenberg–Marquardt or Powell's). Table 4.1 shows the true motion and structure parameters as well as the initial value used for the simulation. Table 4.2 presents the results of a typical simulation using a semilinear method which implements the nonlinear equation with the Powell hybrid method, and table 4.3 displays the results for a typical simulation using an homotopy method.

In the second phase of the first experiment, the same synthetic data are run through a classical CE implementation of the algorithm. Table 4.4 shows the final results and relative errors corresponding to the parameters of table 4.1, and also displays the results for a case similar to the previous one with the exception of the rotation vector that is set to one–tenth of the previous value. As expected, both simulations produce biased estimates and the bias

True rotation in radians:	$\omega_1 = .00698$	$\omega_2 = -.00524$	$\omega_3 = .00873$
True rotation in degrees:	$\omega_1 = .4$	$\omega_2 = -.3$	$\omega_3 = .5$
True translation:	$t_1 = .00781$	$t_2 = .00469$	$t_3 = -.01172$
True translation in pixels:	$t_1 = 1.0$	$t_2 = .4$	$t_3 = -1.5$
True normal:	$n_1 = .4663$	$n_2 = -.2956$	$n_3 = 1.0$
Orientation of plane:	Slant = 15°	tilt = 25°	
Initial guess for normal:	$n_1 = .0$	$n_2 = .0$	$n_3 = 1.0$

Table 4.1: Motion and structure parameters and initial values used in the planar experiments with synthetic data.

Iter	Rotation			Translation			Normal		
No	ω_1	ω_2	ω_3	t_1	t_2	t_3	n_1	n_2	n_3
1	.00929	−.00192	.01103	.00326	.00657	−.01131	.12152	.01013	1.0
2	.00888	−.00220	.01077	.00349	.00616	−.01129	.17112	−.00552	1.0
3	.00880	−.00239	.01064	.00368	.00617	−.01139	.19854	−.01630	1.0
4	.00876	−.00254	.01055	.00385	.00617	−.01143	.21689	−.02486	1.0
5	.00872	−.00267	.01049	.00401	.00617	−.01146	.23116	−.03237	1.0
10	.00858	−.00316	.01024	.00464	.00612	−.01151	.28042	−.06353	1.0
20	.00842	−.00356	.01001	.00518	.00603	−.01153	.31890	−.09374	1.0
30	.00819	−.00401	.00973	.00584	.00585	−.01156	.36072	−.13369	1.0
40	.00808	−.00417	.00961	.00608	.00576	−.01158	.37563	−.15016	1.0
50	.00790	−.00443	.00943	.00647	.00560	−.01159	.39817	−.17782	1.0
100	.00733	−.00499	.00896	.00739	.00505	−.01166	.44635	−.25316	1.0
150	.00716	−.00512	.00884	.00761	.00487	−.01169	.45698	−.27452	1.0
200	.00705	−.00519	.00877	.00774	.00476	−.01171	.46298	−.28778	1.0
300	.00699	−.00523	.00873	.00780	.00470	−.01172	.46585	−.29452	1.0
400	.00698	−.00524	.00873	.00781	.00469	−.01172	.46624	−.29545	1.0
449	.00698	−.00524	.00873	.00781	.00469	−.01172	.46629	−.29556	1.0

Table 4.2: Evolution of the motion and structure parameter estimates for experiment one using FICE. Semi–linear implementation with a Powell hybrid method for the nonlinear equation.

Iter	Rotation			Translation			Normal		
No	ω_1	ω_2	ω_3	t_1	t_2	t_3	n_1	n_2	n_3
1	.00919	−.00166	.01108	.00293	.00648	−.01119	.04712	−.16159	1.0
2	.00824	−.00211	.01066	.00352	.00582	−.01165	.10701	−.21627	1.0
3	.00783	−.00283	.01027	.00441	.00548	−.01175	.18785	−.23505	1.0
4	.00765	−.00369	.00984	.00549	.00530	−.01168	.28642	−.23649	1.0
5	.00753	−.00447	.00942	.00654	.00515	−.01157	.37468	−.23860	1.0
10	.00705	−.00522	.00877	.00778	.00474	−.01167	.46232	−.28711	1.0
15	.00699	−.00523	.00873	.00781	.00469	−.01171	.46592	−.29476	1.0
20	.00698	−.00524	.00873	.00781	.00469	−.01172	.46626	−.29552	1.0
25	.00698	−.00524	.00873	.00781	.00469	−.01172	.46630	−.29559	1.0
30	.00698	−.00524	.00873	.00781	.00469	−.01172	.46630	−.29560	1.0

Table 4.3: Evolution of the motion and structure parameter estimates for experiment one using FICE. Semi–linear implementation with an homotopy method for the nonlinear equation.

increases with the magnitude of the rotation vector. The relative error in the rotation vector is higher for the smaller rotation, but it is not significant since the magnitude of the rotation is very small and the absolute error is barely 3×10^{-5} radians, i.e. 1.7×10^{-3} degrees. This experiment demonstrates that the FICE provide better performance in the case where the temporal variations of shading, albeit small, should not be neglected and the computational cost is justified.

The second experiment relies only on the 8–bit quantized, synthetic irradiance values. The synthetic image is the same as the one used in the first set of experiments and shown in figure 4.1. In this experiment, the spatiotemporal gradients are estimated directly from the quantized irradiance data. On a gray level picture, the estimates of the gradients are virtually impossible to distinguish from the analytically computed gradients, but the effect of the quantization of the irradiance and of the defects due to the estimation process can be seen, for example, on the contour plot of the gradient field E_x. Figure 4.2 shows the contour plot of the analytical gradient field, while figure 4.3 displays the contour plot of the estimated field. Small variations in levels and contour smoothness can be observed. These variations in turn affect the accuracy of the estimates of the motion and structure parameters. Table 4.5 recapitulates the final estimates of the parameters for the synthetic gradients and synthetic irradiance cases using the FICE and CE algorithms. As expected, the estimated parameters are not identical to the true

True rotation in degrees: $\omega_1 = .4$, $\omega_2 = -.3$, $\omega_3 = .5$									
CE algo.	Rotation			Translation			Normal		
Final iter.	.00700	−.00531	.00871	.00794	.00467	−.01166	.48020	−.2835	1.0
Rel err (%)	.3	1.3	.2	1.7	.4	.5	3	4.2	.0
True rotation in degrees: $\omega_1 = .04$, $\omega_2 = -.03$, $\omega_3 = .05$									
CE algo.	Rotation			Translation			Normal		
Final iter.	.00073	−.00054	.00087	.00784	.00472	−.01172	.46609	−.2927	1.0
Rel err (%)	6.4	14	1	.4	.6	.0	.06	.1	.0

Table 4.4: Final estimates and relative errors of the motion and structure parameters using the CE for two values of the rotation vector. Other parameters are determined by table 4.1.

	Rotation			Translation			Normal		
True para.	.00698	−.00524	.00873	.00781	.00469	−.01172	.4663	−.2956	1.0
FICE est.	.00702	−.00533	.00870	.00798	.00466	−.01160	.4852	−.2895	1.0
Rel err (%)	.6	2.1	.3	2.2	.6	1.0	4.0	4.1	0.0
CE est.	.00706	−.00552	.00867	.00829	.00462	−.01149	.5191	−0.2654	1.0
Rel err (%)	1.1	5.3	.7	6.1	1.5	2.0	11.3	10.2	0.0

Table 4.5: Final estimates and relative errors of the motion and structure parameters using the FICE and CE on the synthetic quantized irradiance sequence depicted in figure 4.1. All the parameters are specified by table 4.1.

ones, although the variations are small (on the order of 1–4%) and the bias between the results computed with the FICE and CE is about constant. This result is not surprising since the implementations of the FICE and CE are almost identical.

This experiment demonstrates that the algorithm is robust in the case of noiseless, quantized data and very good performances are achieved in the estimation of the motion and structure parameters. No noise study of the two–frame implementation is performed in this chapter since a study of the performance of the general multi–frame algorithm with respect to noise is done in chapter 6.

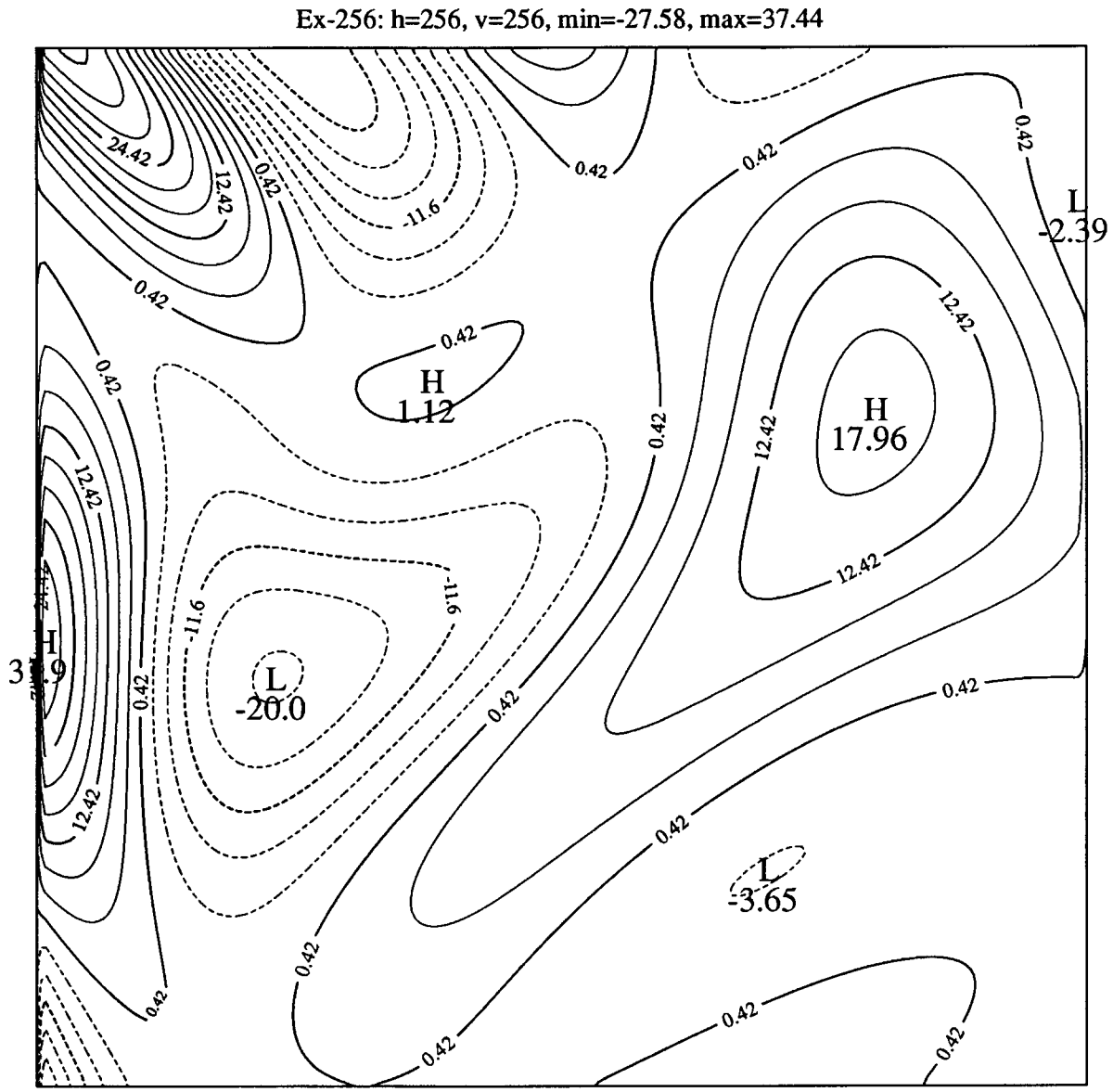

Figure 4.2: Contour plot of the analytically computed horizontal gradients of the lower right quadrant of the irradiance image shown in figure 4.1(a).

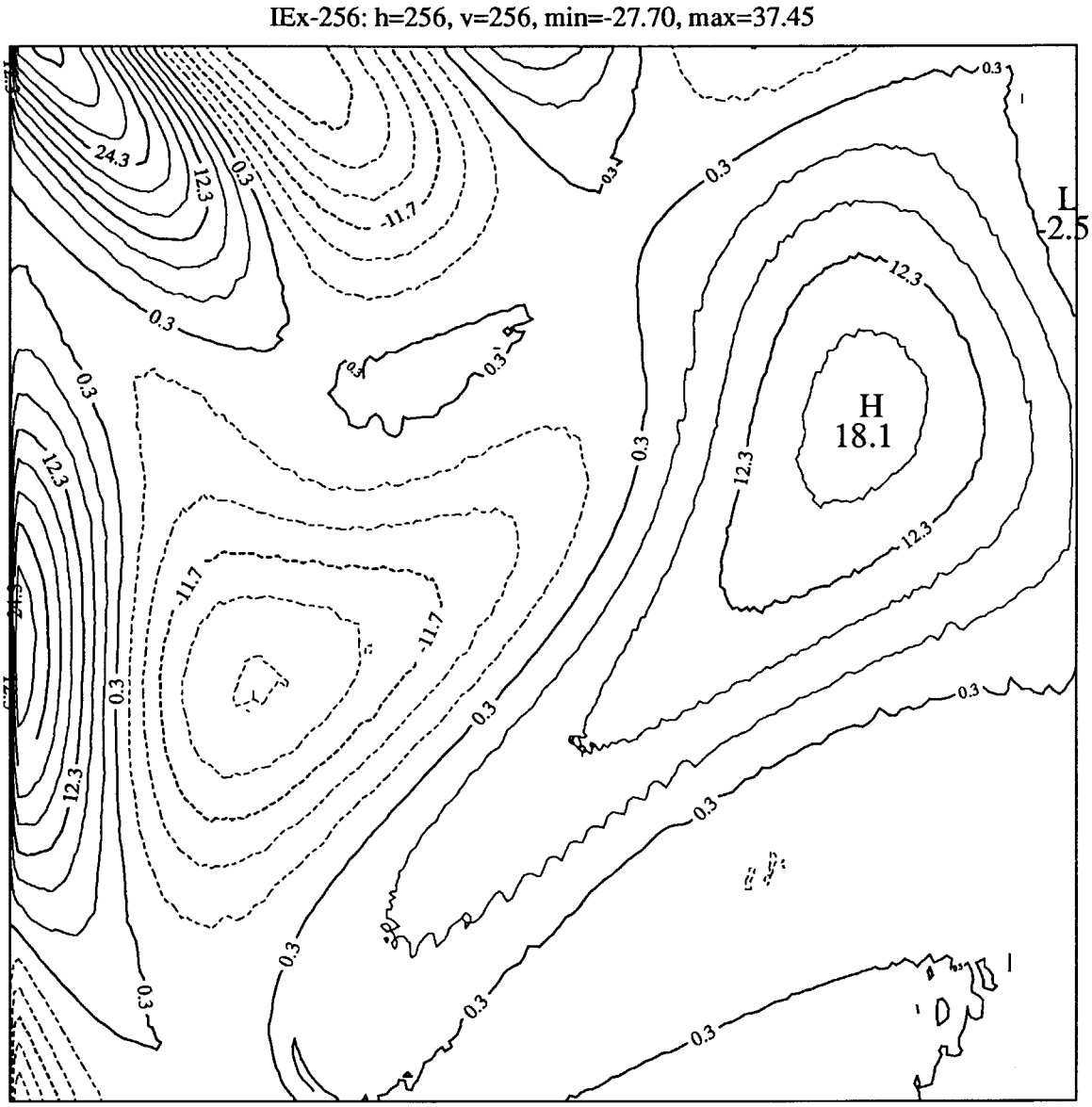

Figure 4.3: Contour plot of the estimated horizontal gradients of the lower right quadrant of the 8–bit quantized image shown in figure 4.1(a). The plot is the contour plot of the lower right quadrant of the horizontal gradients depicted in figure 4.1(c)

Camera parameters: F = 60mm, pixel size 1/60mm Distance focal plane to object: Z_0 = 1250mm									
	Rotation in degrees			Translation in pixels			Normal		
Measured parameters	.0	.0	.5	1.0	−1.0	.0	.0	.0	1.0
Final estimates	.031	−.081	.590	.682	−.695	.0814	.0542	-.034	1.0
Absolute rel. error (%)	3.1	8.1	18.0	31.8	30.5	8.1	5.4	3.4	0.0
Corrected estimates	.031	−.081	.59	.859	−.836	.0814	.0542	−.034	1.0
Absolute rel. error (%)	3.1	8.1	18.0	14.1	16.4	4.1	5.4	3.4	0.0

Table 4.6: Final and adjusted estimates and corresponding relative errors of the motion and structure parameters using the multiframe DFU FICE on the real irradiance sequence shown in figure 4.4.

4.2.1.2 Real Data

The last set of experiments that were performed for the distant source case involves real data. The object used in these experments is a plane covered with a highly textured, matte piece of wall paper illuminated by a diffuse light source. Due to the primitive experimental set–up, only two degrees of freedom are available, a translation along the optical axis and a rotation around the optical axis. Two more degrees of freedom are generated, two translations parallel to the image plane, by synthetically shifting the irradiance images by an integer number of pixels in the x and y directions.

The results of the experiment need to be taken with a grain of salt since all the measurements were done *directly* on the experimental set–up, and the rotation was performed manually and measured with a protractor. The real purpose of the experiment is to show that plausible results can be obtain with real images using the multi–frame DFU algorithm. Figure 4.4(a) shows the irradiance data, figure 4.4(b) the temporal gradient, and figure 4.4(c–d) the spatial gradients used in the experiment. The gradients are computed using the DFU method for the approximate motion parameters shown in table 4.6. Table 4.6 summarizes the measured parameters of the experiment and the computed motion and structure parameters.

As expected, the computed and measured parameters are fairly different. Several factors can explain the discrepancies. The planar surface being imaged was not rigorously perpendicular to the optical axis, resulting in some extraneous, nonzero $\mathbf{n}_x$ and $\mathbf{n}_y$ components as well as

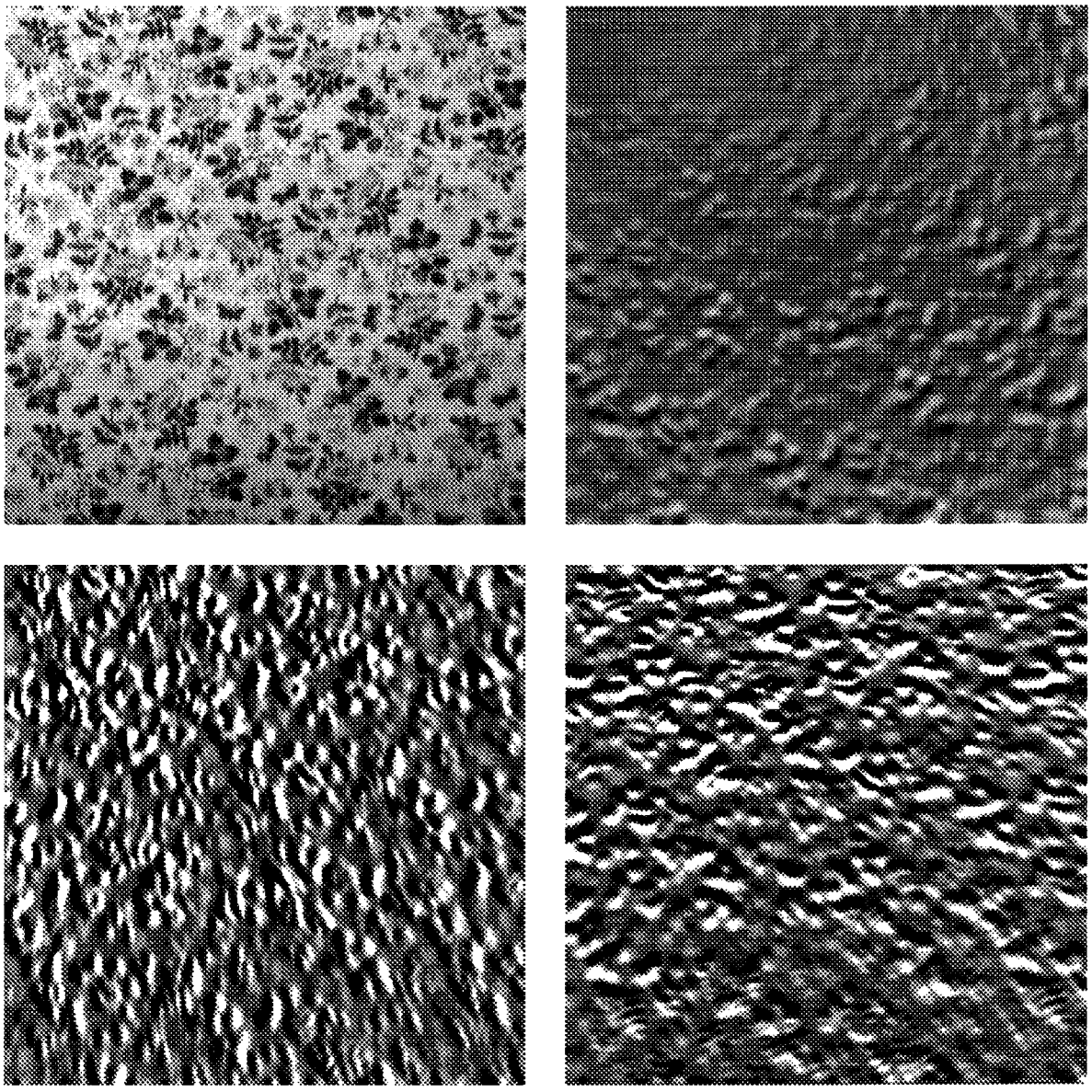

Figure 4.4: Real data of matte piece of wallpaper on a frontal plane. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (middle left and right) the x– and y–gradients.

spurious ω_x and ω_y components. The value of the rotation around the optical axis is reasonable considering the crude measured effective rotation. On the other hand, the estimate of the translation is rather poor both in terms of the relative size of the each component and in terms of the overall scale factor that is determined by the measured distance Z_0 and the effective focal distance F of the camera. However, Z_0 is subject to measurement errors and the advertised value of the lens focal length (60 mm) was used; it is known that this differs substantially from the effective focal length. The anomalies in the x and y components of the translation vector can be explained by the fact that the center of rotation of the planar surface was not exactly coincident with the intersection of the plane with the optical axis; therefore, the measured translation is in fact the sum of the true translation, i.e. the translation obtained when the optical axis and the axis of rotation are coincident and two translation components in the plane perpendicular to the optical axis. This assumption was verified experimentally by running the algorithm on a set of data obtained from the same set–up but with only a rotation around the optical axis and no translation. The true center of rotation was estimated from the resulting optical flow and found to be approximately at the coordinates (13, 9) from the intersection of the optical axis with the plane. This position was confirmed by the measurement of the focus of expansion location on a sequence, where the motion was a pure translation along the optical axis. Another set of experiments were run with only synthetic translations in a plane parallel to the image plane in order to estimate the overall scale factor for the translation vector; the correction was found to be 1.08. The final estimates obtained during the original experiment were adjusted to take into account the induced translation provoked by the off–center center of rotation and the overall translation scale factor and the corrected values are also shown in table 4.6. The compensated values are far better than the raw estimates and are very respectable.

4.2.2 Nearby Source

The case of a planar Lambertian surface illuminated by a nearby source is richer than the previous, distant source situation but is also much more complex to implement due to the analytical complexity of the expressions for the shading and temporal shading variations (equations (3.1) and (3.2)). The spatial irradiance gradients are produced in part by the spatial variations of

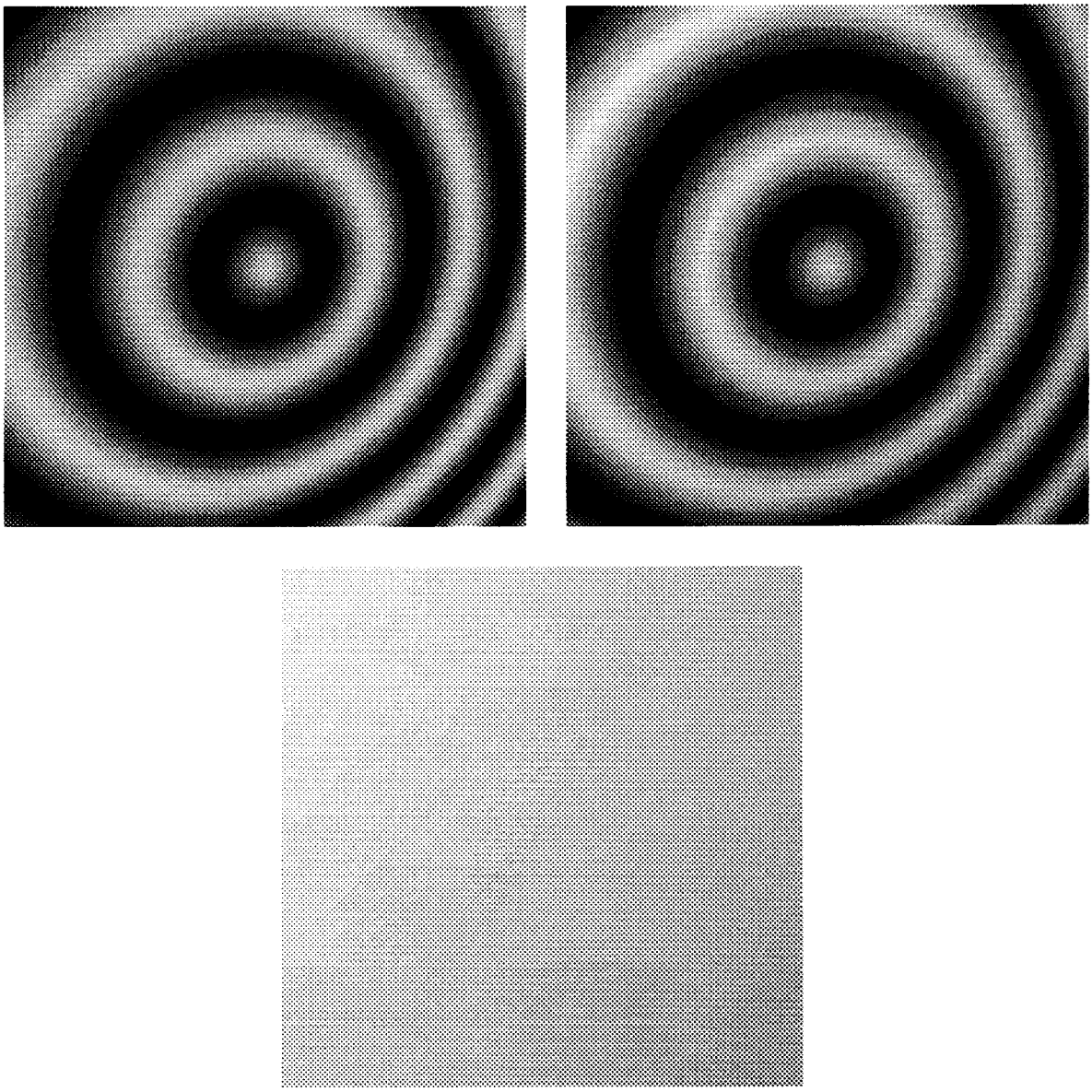

Figure 4.5: Exponentially damped cosine on a slanted plane. (a) (top left) irradiance image with far–away source, (b) (top right) irradiance image for close–up source, (c) (bottom) irradiance image for textureless surface.

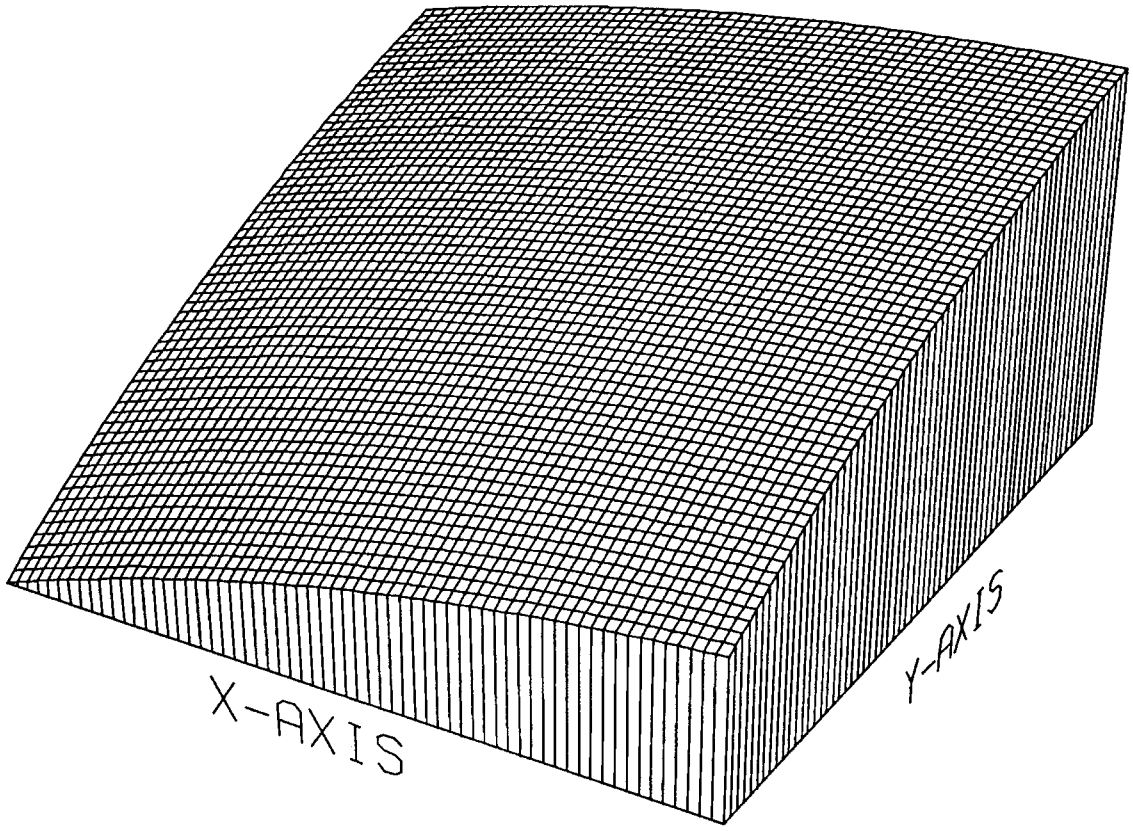

Figure 4.6: Surface plot of the irradiance image of textureless surface shown in figure 4.5(a).

the texture and in part by the spatial variations of the shading across the surface as opposed to the distant case where the spatial irradiance gradients are only due to the texture. The spatial shading variations are induced by the proximity of the light source that is seen from a slightly different spatial position by each point on the surface. Figure 4.5(a) shows an exponentially damped zone plate texture illuminated by a distant source in direction $\hat{\mathbf{l}}$, and figure 4.5(b) displays the same textured plane illuminated by a nearby source in the same direction. The difference between the two cases is best demonstrated by figure 4.5(c) that represents a textureless plane, with the same geometry and position as before, illuminated with the same nearby source. Figure 4.5(c) is in fact the ratio of the irradiance of the planar surface illuminated by the nearby source by the irradiance of the same planar surface illuminated by the distant source. The perspective plot (figure 4.6) of the irradiance function depicted in figure 4.5(c) provides a more striking picture of the spatial shading variations across the surface. The reference level of the plot is the flat irradiance level that occurs in the distant source case. More specifically, the plot shows exactly the additive shading variations, across the surface, that are present in the nearby source case as opposed to the distant source case, i.e. it represents the difference between the irradiance produced by the nearby and distant sources.

It can be observed that the spatial irradiance variations are very smooth and, as expected, the small spatial irradiance variations generate, in turn, extremely smooth spatial gradients (see figures 4.7(a) and 4.7(b) for the horizontal and vertical gradients respectively) with practically no contrast. These very small spatial gradients cause a lot of difficulty in determining the structure and motion parameters. The very smooth gradients result in a very slow convergence of the algorithm and can provoke instabilities.

Table 4.7 shows the results of a typical run for a textureless planar surface with the structure and motion parameters described by table 4.1. These results were obtained for "near–perfect" data, i.e. the irradiance sequence was kept as a floating point array and the gradients were computed directly from the floating point sequence. Under these favorable conditions it takes about 16000 iterations to converge from a frontal plane initial estimate of the normal ($\mathbf{n}$ = (0 0 1)) and the convergence is excruciately slow as the figures in the tables show.

Moderate success were obtained for the 8–bit quantized data with an arbitrary initial estimate of the normal. Convergence with relative error of 3–10% between the estimated parameters

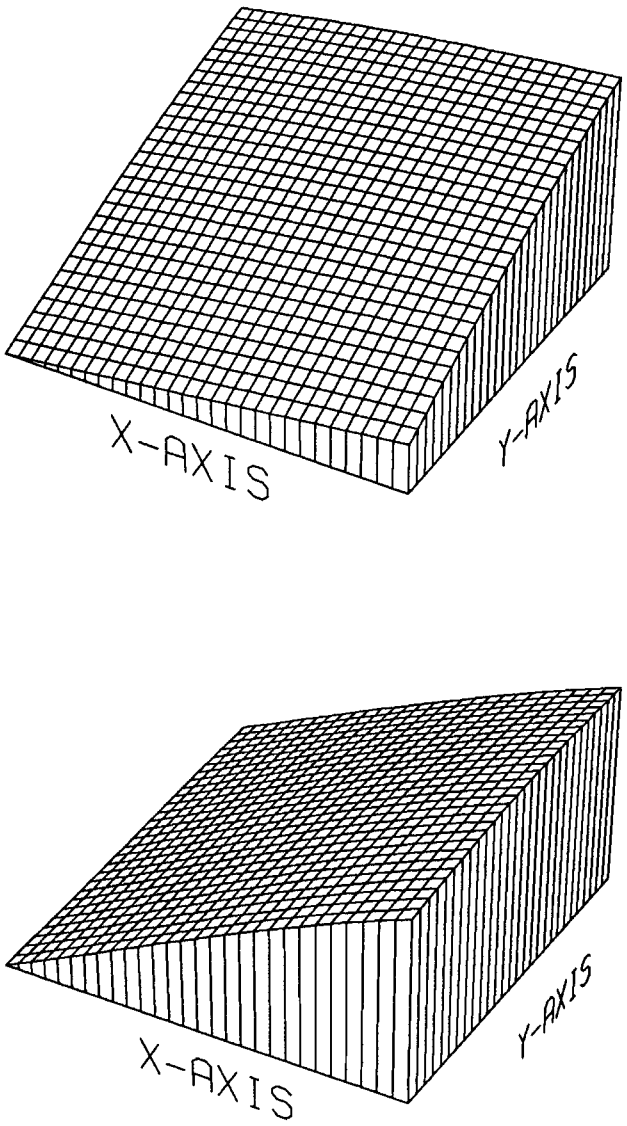

Figure 4.7: Surface plot of the spatial irradiance gradients of the textureless surface shown in figure 4.5(a). (a) (top) horizontal gradient, (b) (bottom) vertical gradient.

Iter	Rotation			Translation			Normal		
No	ω_1	ω_2	ω_3	t_1	t_2	t_3	n_1	n_2	n_3
1	.01080	.00062	.01445	−.00338	.01583	−.01379	.00596	−.00477	1.0
2	.01077	.00057	.01454	−.00365	.01633	−.01378	.00890	−.00717	1.0
3	.01073	.00052	.01463	−.00389	.01680	−.01378	.01179	−.00955	1.0
4	.01070	.00048	.01470	−.00412	.01723	−.01377	.01462	−.01190	1.0
5	.01067	.00043	.01477	−.00431	.01762	−.01377	.01737	−.01420	1.0
10	.01053	.00023	.01495	−.00487	.01887	−.01376	.02975	−.02462	1.0
25	.01028	−.00015	.01477	−.00443	.01888	−.01372	.05351	−.04490	1.0
50	.01006	−.00040	.01433	−.00330	.01746	−.01370	.07045	−.06258	1.0
100	.00975	−.00057	.01392	−.00230	.01593	−.01372	.08060	−.08389	1.0
200	.00932	−.00063	.01362	−.00166	.01463	−.01381	.08243	−.11251	1.0
300	.00899	−.00064	.01347	−.00140	.01390	−.01390	.08074	−.13369	1.0
400	.00872	−.00063	.01338	−.00123	.01336	−.01396	.07910	−.15084	1.0
500	.00849	−.00063	.01330	−.00109	.01293	−.01401	.07795	−.16535	1.0
1000	.00768	−.00067	.01303	−.00063	.01148	−.01416	.07796	−.21711	1.0
1500	.00714	−.00078	.01283	−.00028	.01055	−.01422	.08368	−.25237	1.0
2000	.00672	−.00093	.01265	.00007	.00984	−.01425	.09377	−.28042	1.0
3000	.00602	−.00151	.01218	.00099	.00859	−.01422	.13367	−.32902	1.0
4000	.00535	−.00287	.01122	.00279	.00722	−.01392	.23157	−.38022	1.0
5000	.00516	−.00433	.01003	.00477	.00622	−.01330	.34746	−.40355	1.0
10000	.00663	−.00517	.00889	.00730	.00503	−.01201	.45027	−.31904	1.0
11000	.00675	−.00520	.00883	.00748	.00492	−.01191	.45587	−.31125	1.0
12000	.00683	−.00521	.00879	.00759	.00484	−.01185	.45954	−.30592	1.0
13000	.00688	−.00522	.00877	.00767	.00479	−.01180	.46194	−.30234	1.0
14000	.00692	−.00523	.00875	.00772	.00475	−.01177	.46349	−.29998	1.0
15000	.00694	−.00523	.00874	.00775	.00473	−.01175	.46449	−.29844	1.0
16000	.00695	−.00523	.00874	.00777	.00471	−.01174	.46513	−.29743	1.0
16631	.00696	−.00523	.00874	.00778	.00471	−.01174	.46542	−.29699	1.0

Table 4.7: Evolution of the motion and structure parameter estimates for a textureless planar surface using the FICE and a Powell hybrid implementation of the nonlinear equation. Motion and structure parameters are as specified in table 4.1.

and the true parameters was obtained for initial estimates relatively close to the true value of the normal, but no systematic way of determining the initial estimate was found. A classical CE implementation cannot be used, as in the previous section, to obtain a crude initial estimate, because such an implementation cannot deal with a textureless surface where all the variations are due to shading.

The main reason for the near failure in determining the structure and motion parameters, in the textureless planar surface case, is that the shading variations are too weak, even for a very close light source, to generate sizable spatial gradients. The problem is easier, in this respect, for quadratic patches because the shading variations are much more pronounced due to the curvature of the surface and the spatial gradients have significantly more contrast. Unfortunately, quadratic patches are far more complex to deal with, and only the distant source case can be reasonably dealt with. Chapter 5 presents an example of a textureless surface for a distant light source.

4.2.3 Attenuated Lambertian Model

The attenuated Lambertian model is closely related to the previously described nearby source model, with the addition that the irradiance is now proportional to the reciprocal of the square of the distance between the source and the planar patch. Such a model is usable for the estimation of the motion and structure parameters of a textureless surface, because the shading across the surface is not uniform. By comparison to the previous model, the spatial variations of the shading are stronger, and the spatial gradients have higher contrast because of the additional attenuation term that modulates the irradiance at each point. In fact, each point on the surface sees a light source in a slightly different spatial position and with a slightly different intensity. Figure 4.8(a) presents a planar surface with an attenuated Lambertian reflectance function and illuminated by a nearby source; figure 4.8(b) displays the same surface illuminated by the same light source but with a regular, nonattenuated, Lambertian surface and figure 4.8(c) shows the ratio of the irradiance of the attenuated Lambertian surface of figure 4.8(a) and the irradiance of the Lambertian surface of figure 4.8(b). As such, figure 4.8(c) corresponds to the irradiance of a textureless surface with an attenuated Lambertian reflectance and illuminated by the same nearby source. As expected, the spatial shading variations over the surface are more pronounced

than in the nearby case with a nonattenuated Lambertian reflectance (see figure 4.5(c) and figure 4.8(c)); the estimation should to be easier to carry out, i.e. the convergence rate should be higher and the choice of the initial estimate for the surface normal less critical than before.

The general attenuated case is virtually intractable without the source–viewer approximation, presented in section 3.2.2, resulting in the simplified system of equations (4.12), (4.13) and (4.14). Table 4.8 presents the results of a typical run for a textureless planar surface with the structure and motion parameters described by table 4.1. The experiment was run with "near–perfect" data, i.e. with synthetic floating point irradiance data. The convergence is about eight time faster than in the nearby case although still slower than comparable runs for a textured surface. The estimates of the solution tend to approach the neighborhood of the solution fairly fast and then the rate of convergence drops dramatically. These experiments suggest that faster convergence could be obtained with an implementation that switches numerical method once the estimated solution is close to the true solution.

Table 4.9 displays the final results and relative errors for 8–bit quantized irradiance data. It was not necessary to provide an initial estimate that was very close to the true solution for these experiments, unlike the nonattenuated Lambertian nearby case, because the spatial gradients due to shading were strong enough to allow the algorithm to lock on the correct solution. The relative errors ranged from about .5% to 3% for the various components of the parameters.

4.3 Summary of Results and Comments

This chapter presented several implementations for different types of Lambertian surfaces and illumination conditions (nonattenuated, attenuated), different types of textures (gratings, constant, real data) and for various numerical methods in the planar patch case. It was found that the FICE performs better than its conventional CE counterpart in cases where the CE has been traditionally used (Lambertian surface illuminated by an infinitely distant punctual light source or hemispherical extended light source), and that the FICE implementation can recover the structure and motion parameters in the case of a textureless surface.

The results of the simulations demonstrated the advantages of the FICE in providing extra accuracy in computing the parameters and in extending the range of cases that can be handled by the algorithm. Specifically, the FICE formulation can deal with weak surface markings

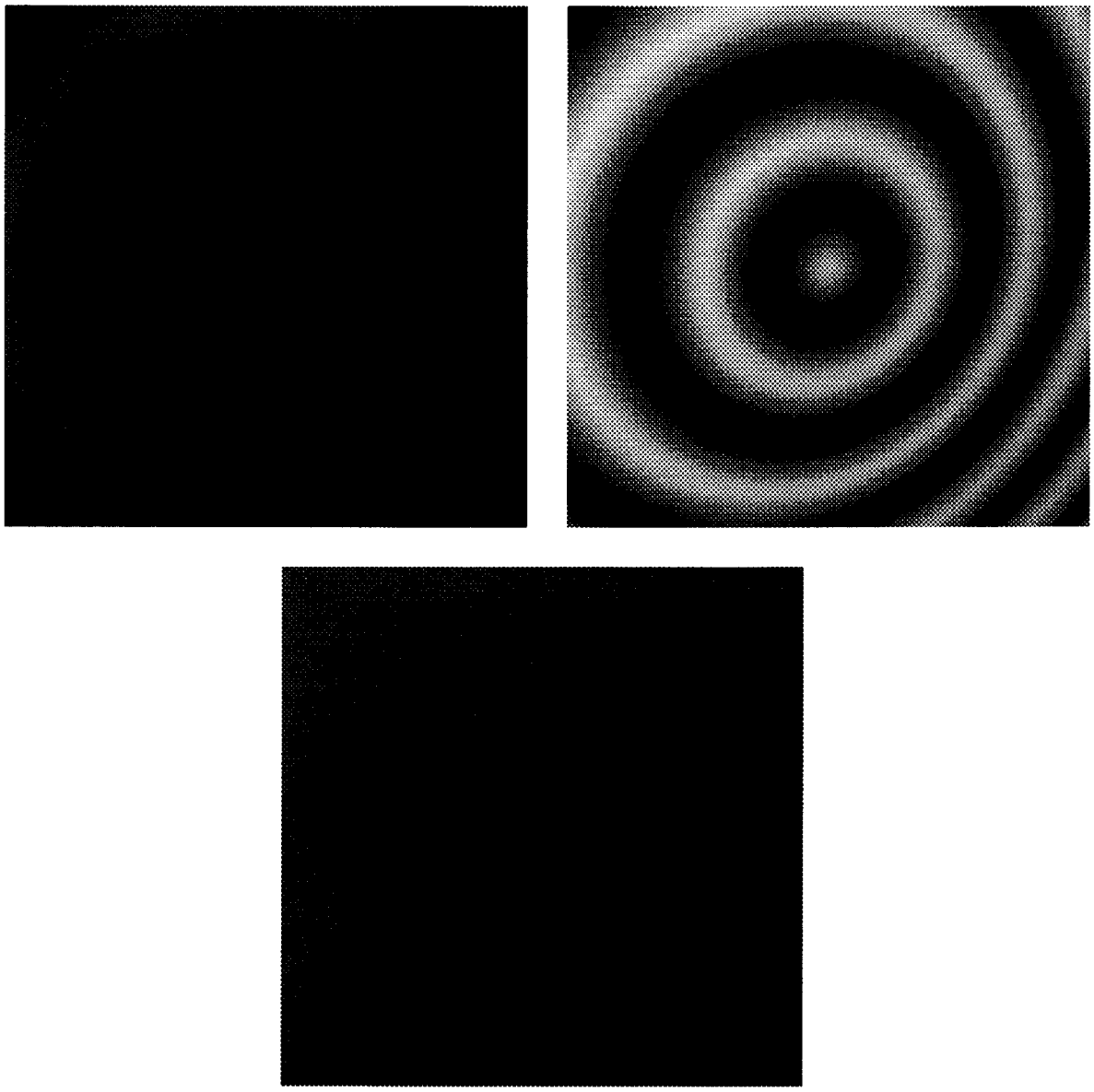

Figure 4.8: Exponentially damped cosine on a slanted plane. (a) (top left) irradiance image with close–up source and attenuated Lambertian reflectance function, (b) (top right) irradiance image for close–up source and regular Lambertian surface, (c) (bottom) irradiance image for textureless attenuated Lambertian surface.

Iter	Rotation			Translation			Normal		
No	ω_1	ω_2	ω_3	t_1	t_2	t_3	n_1	n_2	n_3
1	.01117	.00100	.01442	−.00107	.01251	−.01378	.01210	−.00769	1.0
2	.01112	.00091	.01447	−.00110	.01269	−.01376	.01784	−.01119	1.0
3	.01108	.00081	.01451	−.00112	.01284	−.01375	.02335	−.01446	1.0
4	.01104	.00073	.01455	−.00112	.01298	−.01374	.02864	−.01752	1.0
5	.01101	.00064	.01457	−.00112	.01309	−.01372	.03370	−.02036	1.0
10	.01088	.00029	.01458	−.00096	.01338	−.01365	.05569	−.03188	1.0
50	.01045	−.00077	.01372	.00066	.01231	−.01340	.12339	−.07127	1.0
100	.00997	−.00105	.01327	.00129	.01131	−.01343	.13901	−.10291	1.0
200	.00922	−.00134	.01283	.00187	.01013	−.01353	.15403	−.14948	1.0
300	.00870	−.00163	.01247	.00235	.00930	−.01355	.17161	−.18277	1.0
400	.00829	−.00196	.01213	.00286	.00862	−.01351	.19346	−.20874	1.0
500	.00797	−.00234	.01175	.00344	.00800	−.01343	.22020	−.23015	1.0
1000	.00707	−.00461	.00947	.00680	.00538	−.01227	.40387	−.29016	1.0
1500	.00698	−.00520	.00878	.00774	.00473	−.01176	.46224	−.29587	1.0
2000	.00698	−.00523	.00873	.00781	.00469	−.01172	.46607	−.29565	1.0
2047	.00698	−.00523	.00873	.00781	.00469	−.01172	.46615	−.29563	1.0

Table 4.8: Evolution of the motion and structure parameter estimates for a textureless planar surface with an attenuated Lambertian reflectance function using the FICE and a Powell hybrid implementation of nonlinear equation. Motion and structure parameters are as specified in table 4.1.

	Rotation			Translation			Normal		
True para.	.00698	−.00524	.00873	.00781	.00469	−.01172	.4663	−.2956	1.0
Last est.	.00694	−.00508	.00877	.00757	.00472	−.01184	.4798	−.2861	1.0
Rel err (%)	.6	3.0	.5	3.1	.6	1.0	2.9	3.2	.0

Table 4.9: Final estimates and relative errors of the motion and structure parameters using the FICE on the synthetic quantized irradiance sequence depicted in figure 4.5(a). All the parameters are specified by table 4.1.

and textureless surfaces *provided* the spatial variations of the shading are strong enough to generate sufficient spatial gradients as in the nearby source and attenuated Lambertian cases. The major drawback of the FICE algorithm is its very high computational cost as well as its numerical implementation complexity. It is clear that such a formulation is not recommended in applications were only a reasonable estimate is sought since a much simpler CE implementation can provide the required accuracy in cases where the data are reasonably approximated by the constraint equation (see section 3.3.1). On the other hand, if the accuracy of the solution is the prime concern, the FICE implementation is a legitimate candidate.

In addition to the use of the FICE, this chapter showed the advantage of the DFU incremental method to compute the motion parameters. Such a method enabled us to run the algorithm on real data acquired by a video camera and can be implemented very efficiently for rigid motion and optical flow applications independently of the type of constraint equation used.

Chapter 5

Quadratic Patch Estimation

This chapter discusses the specific problems and forms of the general minimization equations, developed in chapter 2, when applied to quadratic patches. Specific implementations with specific shading models are derived and results of the algorithms on synthetic data presented.

Geometrically, a quadratic patch is only slightly more complicated than a planar patch and is fully defined by the normal $\hat{\mathbf{n}}_0$ at a given point $\mathbf{Z}_0$ and its associated principal curvatures d_{p_1} and d_{p_2}, along the principal axes $\mathbf{p}_1$ and $\mathbf{p}_2$; alternatively, the curvature tensor is represented, for convenience, by the vector $\mathbf{d}_0$ at $\mathbf{Z}_0$ (see figure 5.1). The implementation of the quadratic patch case for the CE is only marginally harder than the planar patch situation, and Negahdaripour (1986) proposed an implementation for the direct motion case. One of the major differences with the planar case is that no closed–form solution exists and the structure and motion parameters are estimated by an iterative procedure similar to the semilinear iterative procedure used for the planar patch case. However, in the FICE formulation, the task of estimating the parameters is an order of magnitude more complex, even in the case of the simplest distant punctual source. The complexity stems from the need to determine the normal $\mathbf{n}$ and/or unit normal $\hat{\mathbf{n}}$ at every point on the surface as a function of the normal $\hat{\mathbf{n}}_0$ and curvature vector $\mathbf{d}_0$, and to evaluate the shading at each point. Contrary to the planar case, spatial shading variations always occur, even in the distant source case, and its spatiotemporal variations need to be tracked in the FICE formulation. Figure 5.2(a) and figure 5.2(b) show two examples of shading caused by a distant punctual source for an elliptic hyperboloid and an ellipsoid respectively. The complexity of the formulation and therefore of the implementation

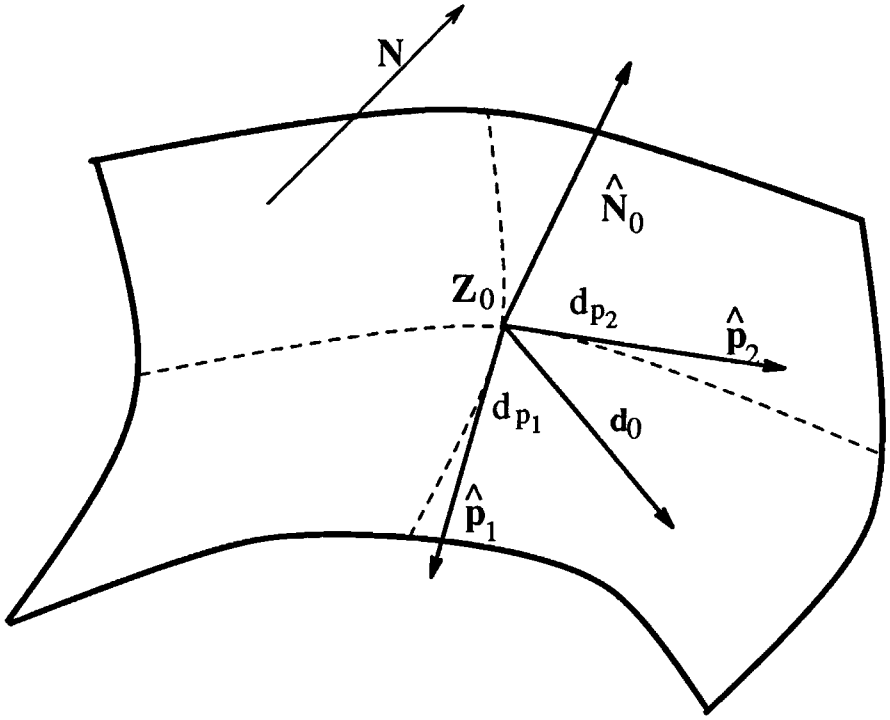

Figure 5.1: Quadratic patch geometry.

limit the study of the quadratic patch case to the distant punctual light source case which can be solved for *both* textured and textureless surfaces.

The next section presents the analytical formulation of the minimization equations in the distant source case and is followed by examples of runs on synthetic data.

5.1 Implementation of the Quadratic Patch Case

A quadratic patch is fully defined by two vectors, a normal $\mathbf{n_0}$ and a curvature(tensor) $\mathbf{d_0}$ at a single point on the surface, and the reciprocal of the depth Z can be expressed in terms of the image coordinates, $\mathbf{r} = (x, y, F)$, the normal $\mathbf{n_0}$ and the curvature $\mathbf{d_0}$ at $\mathbf{Z_0}$. Using a second order Taylor series of the reciprocal of the depth around the point $\mathbf{Z_0}$ (see equation 2.22), $1/Z$ can be approximated by

$$\frac{1}{Z} = \frac{1}{Z_0}(\mathbf{r} \cdot \mathbf{n_0}) + (\mathbf{q} \cdot \mathbf{d_0}) \tag{5.1}$$

where $\mathbf{q} = (\frac{1}{2}x^2, xy, \frac{1}{2}y^2)^T$ is a quadratic coordinate vector, and $\mathbf{d_0} = -(Z_{xx}, Z_{xy}, Z_{yy})^T$ is the curvature vector at the point $\mathbf{Z_0}$. It should be noted that, in the quadratic patch and higher

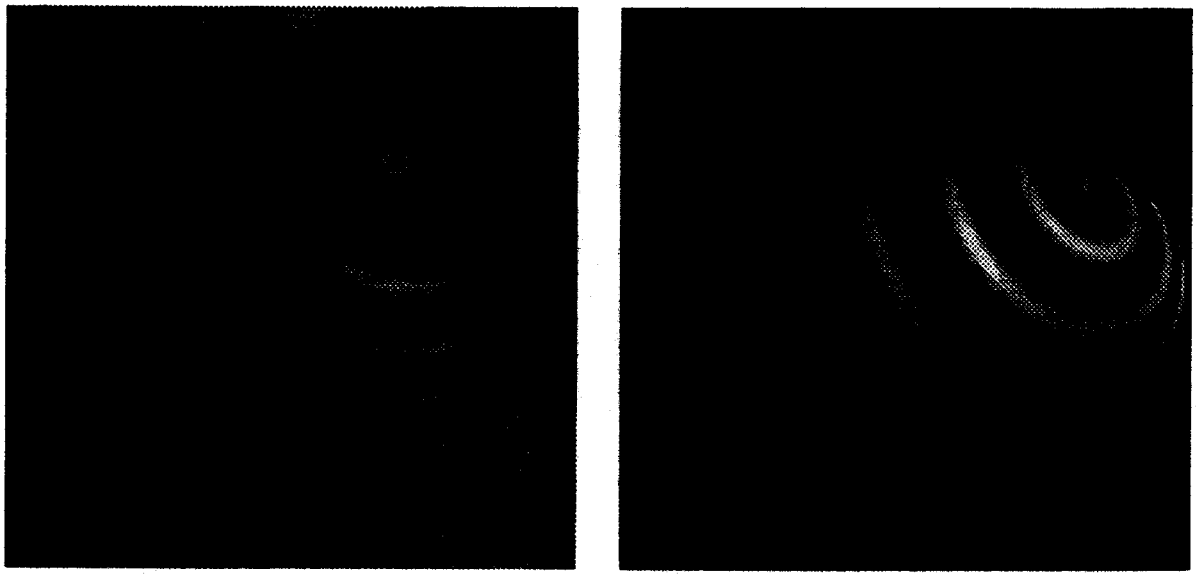

Figure 5.2: Irradiance image of an elliptic hyperboloid (a)(left) of an and ellipsoid (b)(right) illuminated by a distant punctual light source.

dimensional patch cases, the reciprocal of the depth is only given by an approximate equation at a given order, unlike the planar case where $1/Z$ can be expressed *exactly* in terms of the surface coordinates. Equation (5.1) is a second–order approximation.

The normal $\mathbf{n}$ and unit normal $\hat{\mathbf{n}}$ at image point $\mathbf{r}$ can be expressed, to second order, by

$$\mathbf{n} = \mathbf{n}_0 + Z_0\big(1 - (\tilde{\mathbf{r}} \cdot \mathbf{n}_0)\big)\mathbf{H}\mathbf{r} \tag{5.2}$$

and

$$\hat{\mathbf{n}} = \hat{\mathbf{n}}_0 + \frac{Z_0}{\|\mathbf{n}_0\|}\big(1 - \tilde{\mathbf{r}} \cdot \mathbf{n}_0\big)\big(\mathbf{I}_3 - \hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T\big)\mathbf{H}\mathbf{r} - \frac{1}{2}\frac{Z_0^2}{\|\mathbf{n}_0\|^2}\Big(\mathbf{r}^T\tilde{\mathbf{H}}\mathbf{r}\hat{\mathbf{n}}_0 + 2(\hat{\mathbf{n}}_0 \cdot \mathbf{H}\mathbf{r})\mathbf{H}\mathbf{r}\Big) \tag{5.3}$$

where $\tilde{\mathbf{r}} = (x\ y\ 0)^T$, $\tilde{\mathbf{H}} = \mathbf{H}^2 - 3\mathbf{H}\hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T\mathbf{H}$ and

$$\mathbf{H} = \begin{pmatrix} -Z_{xx} & -Z_{xy} & 0 \\ -Z_{xy} & -Z_{yy} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

The details of the derivation that leads to the equations (5.2) and (5.3) can be found in appendix E. The next section derives the quadratic patch minimization equations for a distant punctual source model.

5.1.1 General Equation for the Far–away Punctual Source Case

Let us consider a distant punctual source in the direction $\hat{\mathbf{l}}$ and examine the shading equation for a quadratic patch. Using the second–order Taylor series expansion of the unit normal $\hat{\mathbf{n}}$ (equation 5.3), the distant shading equation takes the form

$$E(\mathbf{r},t) = \rho\Big((\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}_0) + \widetilde{Z_0}\mathbf{l}_\perp^{\hat{\mathbf{n}}_0}\mathbf{H}\mathbf{r} - Z_0(\mathbf{r}\cdot\hat{\mathbf{n}}_0)\mathbf{l}_\perp^{\hat{\mathbf{n}}_0} - \frac{1}{2}\widetilde{Z_0}^2\mathbf{r}^T\big((\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}_0)\widetilde{\mathbf{H}} + 2(\mathbf{H}^T\hat{\mathbf{n}}_0\mathbf{l}_\perp^{\hat{\mathbf{n}}_0})\mathbf{H}\big)\mathbf{r}\Big) \quad (5.4)$$

where $\widetilde{Z_0} = Z_0/\|\mathbf{n}_0\|$ and $\mathbf{l}_\perp^{\hat{\mathbf{n}}_0} = \hat{\mathbf{l}} - (\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}_0)\hat{\mathbf{n}}_0$. Equation 5.4 shows that the irradiance is a quadratic functional of the image coordinates $\mathbf{r}$ and includes constant and linear terms. If the light source direction $\hat{\mathbf{l}}$ is parallel to the normal $\hat{\mathbf{n}}_0$, the previous equation takes the much simplified form

$$E(\mathbf{r},t) = \rho\Big(1 - \frac{1}{2}\widetilde{Z_0}^2\mathbf{r}^T\big(\mathbf{H}^2 + (\mathbf{H}\hat{\mathbf{n}}_0)(\mathbf{H}\hat{\mathbf{n}}_0)^T\big)\mathbf{r}\Big)$$

that is, the spatial irradiance variations are purely quadratic without any linear component, and the spatial gradients $\nabla_{\mathbf{r}}$ are linear,

$$\nabla E_{\mathbf{r}} = -\rho\widetilde{Z_0}^2\big(\mathbf{H}^2 + (\mathbf{H}\hat{\mathbf{n}}_0)(\mathbf{H}\hat{\mathbf{n}}_0)^T\big)\mathbf{r}.$$

Figure 5.3 shows the irradiance, horizontal and vertical gradients of an elliptic hyperboloid with a light source $\hat{\mathbf{l}}$ parallel to the normal $\hat{\mathbf{n}}_0$ of the tangent plane at the origin and figure 5.4 shows the irradiance, horizontal and vertical gradients of an elliptic hyperboloid with a light source not parallel to the normal $\hat{\mathbf{n}}_0$. The smoothness of the spatial shading gradients suggests that the convergence of the algorithm for a textureless surface is going to be very slow and that the global behavior of the algorithm is going to be similar to the nearby source, textureless, planar patch one.

Let $F(\boldsymbol{\omega},\mathbf{t},\hat{\mathbf{n}}_0,\mathbf{d}_0) = E_t - (\mathbf{v}\cdot\boldsymbol{\omega}) - (\mathbf{r}\cdot\hat{\mathbf{n}}_0)(\mathbf{s}\cdot\mathbf{t}) - (\mathbf{q}\cdot\mathbf{d}_0)(\mathbf{s}\cdot\mathbf{t}) - \rho(\mathbf{r})[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}(\mathbf{r})]$, where $\hat{\mathbf{n}}(\mathbf{r})$ is specified by equation (5.3). The unconstrained minimization equation can be written in the form

$$C = \min\iint_\sigma\Big(F^2 + \mu(\mathbf{r})\big(E(\mathbf{r},t) - \rho(\mathbf{r})(\hat{\mathbf{l}}\cdot\hat{\mathbf{n}}(\mathbf{r}))\big)\Big)d\mathbf{r} + \lambda\big(\|\hat{\mathbf{n}}_0\|^2 - 1\big).$$

At an extremum of C, the derivatives of C with respect to the motion parameters $\boldsymbol{\omega}$, $\mathbf{t}$ and structure parameters $\hat{\mathbf{n}}_0$, $\mathbf{d}_0$ are zero, i.e.

$$\frac{\partial C}{\partial\boldsymbol{\omega}} = 0, \quad \frac{\partial C}{\partial\mathbf{t}} = 0, \quad \frac{\partial C}{\partial\hat{\mathbf{n}}_0} = 0 \quad \text{and} \quad \frac{\partial C}{\partial\mathbf{d}_0} = 0.$$

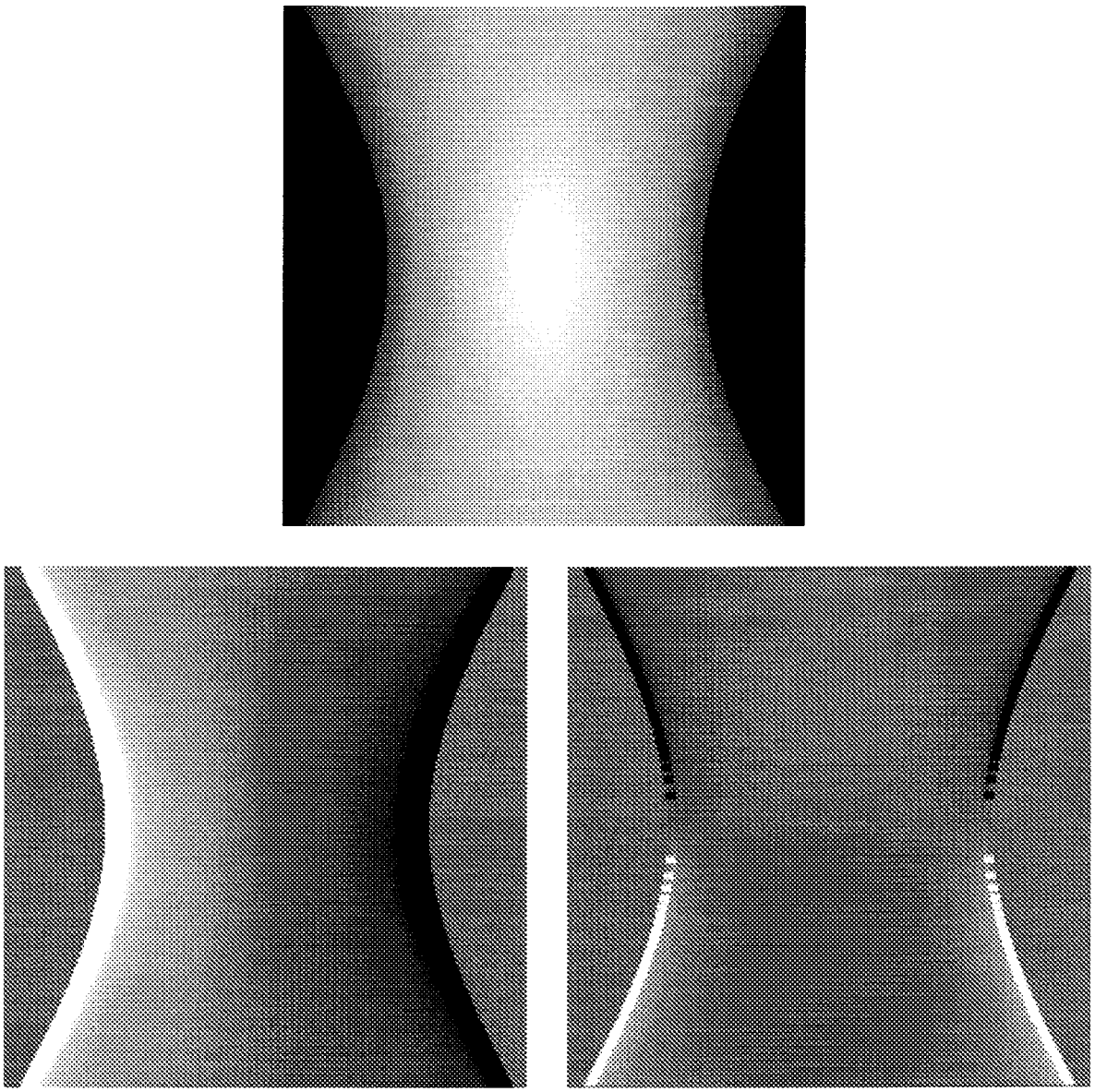

Figure 5.3: Elliptic hyperboloid illuminated by a light source $\hat{\mathbf{l}}$ parallel to the normal $\hat{\mathbf{n}}_0$ of the tangent plane at the origin. (a) (top) irradiance image, (b) (bottom left) horizontal gradient, (c) (bottom right) vertical gradients.

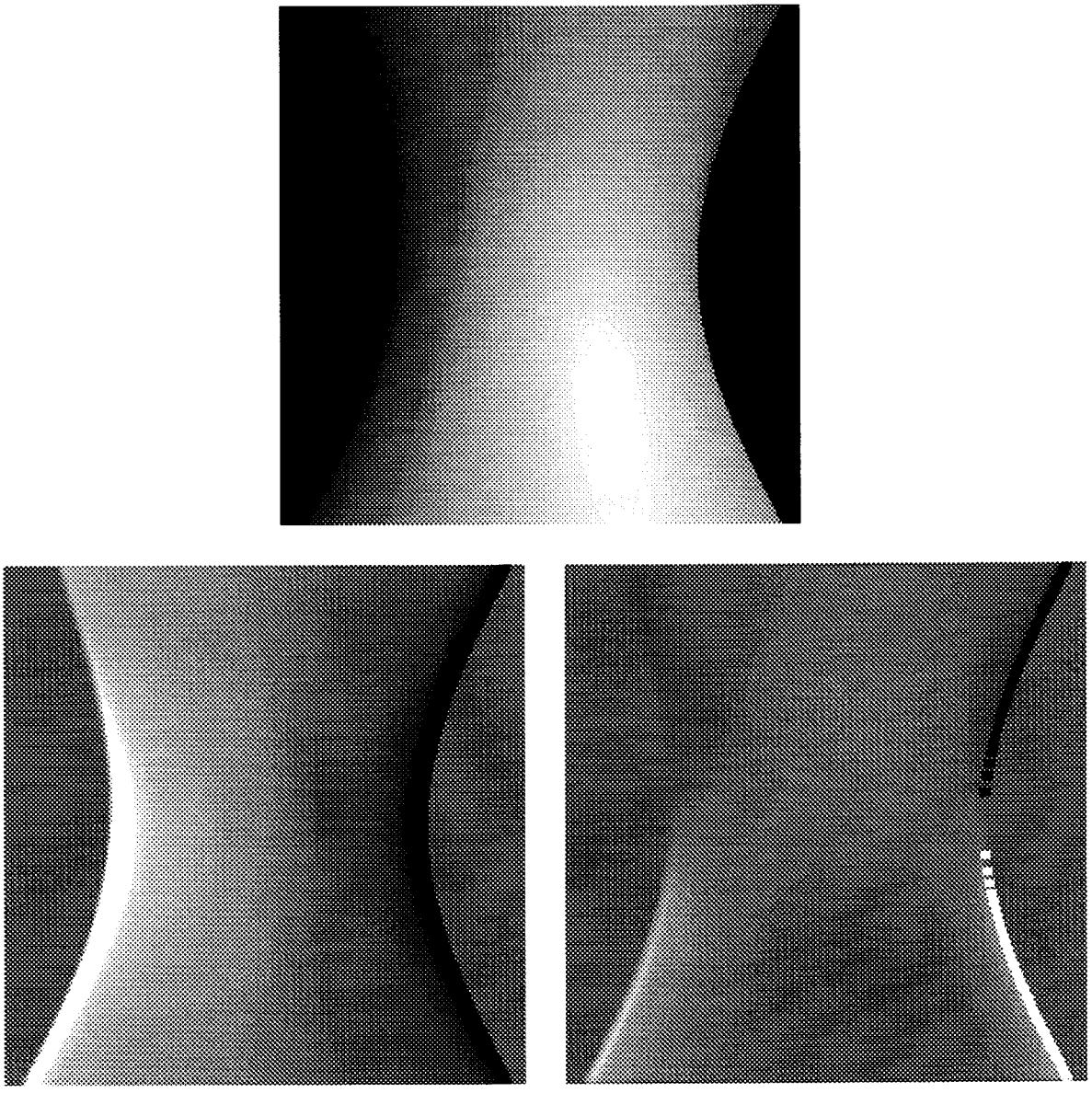

Figure 5.4: Elliptic hyperboloid illuminated by a light source $\hat{\mathbf{l}}$ not parallel to the normal $\hat{\mathbf{n}}_0$ of the tangent plane at the origin. (a) (top) irradiance image, (b) (bottom left) horizontal gradient, (c) (bottom right) vertical gradients.

The resulting system of vectorial nonlinear equations, that defines the parameters $\mathbf{t}$, $\boldsymbol{\omega}$, $\hat{\mathbf{n}}_0$ and $\mathbf{d}_0$, takes the form

$$\iint_\sigma F\big(-\mathbf{v} - \rho(\mathbf{r})(\hat{\mathbf{n}}_0 \times \hat{\mathbf{l}})\big)d\mathbf{r} = 0 \qquad (5.5)$$

$$\iint_\sigma F\big(-\mathbf{s}(\mathbf{r}\cdot\hat{\mathbf{n}}_0) - \mathbf{s}(\mathbf{q}\cdot\mathbf{d}_0)\big)d\mathbf{r} = 0 \qquad (5.6)$$

$$\iint_\sigma F\big(-\mathbf{q}(\mathbf{s}\cdot\mathbf{t})\big)d\mathbf{r} = 0 \qquad (5.7)$$

$$\iint_\sigma \left(F\big(-(\mathbf{s}\cdot\mathbf{t})\mathbf{r} - \rho(\mathbf{r})\left(\frac{\partial\hat{\mathbf{n}}}{\partial\hat{\mathbf{n}}_0}\right)^T(\boldsymbol{\omega}\times\hat{\mathbf{l}})\big) - \mu(\mathbf{r})\rho(\mathbf{r})\left(\frac{\partial\hat{\mathbf{n}}}{\partial\hat{\mathbf{n}}_0}\right)^T\hat{\mathbf{l}}\right)d\mathbf{r} + \lambda\hat{\mathbf{n}}_0 = 0 \qquad (5.8)$$

where $\dfrac{\partial\hat{\mathbf{n}}}{\partial\hat{\mathbf{n}}_0}$ is computed using (5.3). The expression of this partial derivative is complex and can be found in appendix E along with its derivation. The Lagrange multiplier λ and the Lagrange multiplier function $\mu(\mathbf{r})$ are eliminated by taking the dot product of equation (5.8) with the vectors $\hat{\mathbf{n}}_0$ and $\hat{\mathbf{l}}$ and solving the resulting scalar linear system in the unknowns λ and $\iint_\sigma \mu(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$. The expressions for the multipliers are very cumbersome, due to the $\dfrac{\partial\hat{\mathbf{n}}}{\partial\hat{\mathbf{n}}_0}$ term, and are omitted here. Once the multipliers are determined, their value is plugged back in (5.8) to obtain a system of four nonlinear vectorial equations in the unknowns $\boldsymbol{\omega}$, $\mathbf{t}$, $\hat{\mathbf{n}}_0$ and $\mathbf{d}_0$.

It should be noted that the previous vectorial equations are, formally, fairly similar to those of the system $\mathcal{S}$ obtained in the distant planar patch case. Equations (5.5) and (4.2) are formally identical, the new parameter $\mathbf{d}_0$ adds an extra term in (5.6) and gives rise to the new equation (5.7). However, the similarity between the two systems of equations in the planar and quadratic cases is deceptive. In fact, the unit normal $\hat{\mathbf{n}} = \hat{\mathbf{n}}(\mathbf{r})$ is no more a constant vector but depends on the image coordinates $\mathbf{r}$ and the previous system of nonlinear equations is far more complex to implement than its planar patch counterpart. In particular, equation (5.8) is extremely hard to implement numerically and the scalar expressions are messy due, in part, to the lack of privileged axes of projection, found in the planar case.

5.1.2 Solution of the Global Nonlinear System

The previous system can be solved either as a global nonlinear system or as a semilinear system. The global nonlinear system was not implemented because the simpler planar case was already exhibiting fairly poor convergence properties, compared to semilinear implementation, and the

quadratic case is even more nonlinear and of higher dimension and its behavior can only be worse.

The first equation is linear in ω, $\mathbf{t}$ and $\mathbf{d}_0$, the second in ω and $\mathbf{t}$, the third in ω and $\mathbf{d}_0$ and the last in $\mathbf{d}_0$ only. Two semilinear implementations can be considered. The global system can either be broken into a linear system in the unknowns ω and $\mathbf{t}$ with the first two equations and a nonlinear system in $\mathbf{d}_0$ and $\mathbf{n}_0$ with the last two equations or into a linear system in ω and $\mathbf{d}_0$ with the first and third equations and a nonlinear system in $\mathbf{t}$ and $\mathbf{n}$ with the second and fourth equations. The former implementation was chosen for two distinct reasons: its implementation is fairly close to the linear patch implementation, and it segregates the parameter estimation into motion parameters obtained from the linear system and structure parameters obtained from the nonlinear system. The linear system can be written in the form

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{N}_2 \\ \mathbf{N}_2^T & \mathbf{N}_4 \end{pmatrix} \begin{pmatrix} \omega \\ \mathbf{t} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{f}_2 \end{pmatrix}, \tag{5.9}$$

where $\mathbf{M}_1$ and $\mathbf{e}_1$ are the same as in the planar case, with $\hat{\mathbf{n}}$ replaced by $\hat{\mathbf{n}}_0$, and

$$\begin{aligned} \mathbf{N}_2 &= \iint_\sigma \Big(\mathbf{r}\cdot\hat{\mathbf{n}}_0 + \mathbf{q}\cdot\mathbf{d}_0\Big)\Big((\mathbf{v} + \rho(\mathbf{r})(\hat{\mathbf{n}}_0 \times \hat{\mathbf{l}}))\mathbf{s}^T\Big)d\mathbf{r} \\ \mathbf{N}_4 &= \iint_\sigma \Big(\mathbf{r}\cdot\hat{\mathbf{n}}_0 + \mathbf{q}\cdot\mathbf{d}_0\Big)^2 \mathbf{s}\mathbf{s}^T d\mathbf{r} \\ \mathbf{f}_2 &= \iint_\sigma E_t\Big(\mathbf{r}\cdot\hat{\mathbf{n}}_0 + \mathbf{q}\cdot\mathbf{d}_0\Big)\,\mathbf{s}\,d\mathbf{r} \end{aligned}$$

and the solution to the system (5.9) is given by

$$\begin{cases} \mathbf{t} &= (\mathbf{N}_4 - \mathbf{N}_2^T\mathbf{M}_1^{-1}\mathbf{N}_2)^{-1}(\mathbf{N}_2^T\mathbf{M}_1^{-1}\mathbf{e}_1 - \mathbf{f}_2) \\ \omega &= -\mathbf{M}_1^{-1}(\mathbf{e}_1 + \mathbf{N}_2\mathbf{t}) \end{cases}. \tag{5.10}$$

The nonlinear part of the system was implemented with Powell's hybrid method.

The next section presents two examples that illustrate the performance of the algorithm and compares the results to those given by a classical CE implementation in the case of a textured surface.

5.2 Examples

Two examples are presented in this section, a saddle surface with an exponentially damped zone plate synthetic texture and the same surface with a constant albedo (textureless surface).

Rotation in radians:	$\omega_1 = -.01047$	$\omega_2 = .00698$	$\omega_3 = .00785$
Rotation in degrees:	$\omega_1 = -.6$	$\omega_2 = .4$	$\omega_3 = .45$
Translation:	$t_1 = .00468$	$t_2 = -.00625$	$t_3 = .00895$
Translation in pixels:	$t_1 = .6$	$t_2 = -.8$	$t_3 = 1.1$
Normal to tangent plane at origin:	$n_1 = .3640$	$n_2 = -.2851$	$n_3 = 1.0$
Orientation of tangent plane:	Slant = 15°	tilt = 20°	
Curvatures at origin:	$d_1 = -1.0$	$d_2 = 0.0$	$d_3 = 1.0$

Table 5.1: Motion and structure parameter values used in the quadratic experiments with synthetic data.

Figure 5.1 lists the motion and structure parameters used in the quadratic patch experiments. A large field of view of 90° is used in the experiments so that the second order terms are significant and the saddle surface does not appear similar to its tangent plane at the origin. The goal of these examples is to show the increased accuracy of the FICE algorithm over the traditional CE algorithms, in the case of textured surfaces, and to demonstrate the ability of the algorithm in recovering the motion and structure parameters for textureless quadratic surfaces. The data that are used are synthetic floating point data, i.e. the irradiance surface were raytraced as floating point arrays, and the spatial gradients are estimated directly from these arrays instead of the eight bit quantized irradiance images. Such data were used to assess the performance of the algorithm with respect to the convergence speed and the initial estimates choice without having to worry about the errors introduced by the quantization process. These experiments are meant to show the maximum performance that can be obtained with nonquantized data.

5.2.1 Textured Surface

As it was argued in section 5.1.2, only a semilinear implementation was used in the quadratic patch case with ω and $\mathbf{t}$ computed from a linear system of equations and $\hat{\mathbf{n}}_0$ and $\mathbf{d}_0$ from a nonlinear system of equations. In many cases, the algorithm failed to converge to a solution or to the correct solution for an arbitrary initial estimate of the structure parameters when the tangent plane of surface at the origin was highly slanted (τ or $\sigma \geq 25^\circ$) or the curvatures were high. In the example used in this section, the tangent plane slant and tilt were small enough ($\tau = 15^\circ$ and $\sigma = 20^\circ$) so that an initial estimate of $\mathbf{n}_0 = (0\ 0\ 1)$ (frontal tangent plane) was

Init normal Init curvat	$n_1 = 0.0$ $d_1 = 0.0$			$n_2 = 0.0$ $d_2 = 0.0$			$n_3 = 1.0$ $d_3 = 0.0$		
FICE	True parameters			Last estimates (1648)			Relative error (%)		
Rotation	−.00470	.00698	.00785	−.00476	.00716	.00778	1.34	2.61	0.84
Translation	.00468	−.00625	.00859	.00464	−.00537	.00869	0.85	1.95	1.21
Normal	.36400	−.28510	1.00000	.37450	−.27760	1.00000	2.91	2.61	
Curvature	−1.00000	.00000	1.00000	−1.01840	.05100	.09784	1.84	0.51	2.16
CE	True parameters			Last estimates (1841)			Relative error (%)		
Rotation	−.00470	.00698	.00785	−.00523	.00667	.00882	11.4	14.5	12.3
Translation	.00468	−.00625	.00859	.00411	−.00711	.00982	12.1	13.8	14.4
Normal	.36400	−.28510	1.00000	.30460	−.32800	1.00000	16.7	15.4	
Curvature	−1.00000	.00000	1.00000	−1.17800	.12300	.83500	17.8	12.3	16.5

Table 5.2: Final estimates and relative errors of the motion and structure parameters using the FICE and CE on the synthetic, floating point irradiance sequence depicted in figure 5.5. All the parameters are specified by table 5.1.

adequate for all types of textures used. On the other hand, an initial null estimate for the curvature (planar surface) only works for sufficiently high contrast textures. Weak textures require a more accurate initial estimate for the curvature and $\mathbf{d}_0 = (-.5\ X\ .5)$ was found to be adequate for any texture mapped onto the surface specified by the parameters of table 5.1. Figure 5.5 shows the irradiance and spatio-temporal gradients of the quadratic patch, defined by table 5.1, mapped by an exponentially damped zone plate texture. Table 5.2 shows the real values, the initial and final estimates of the motion and structure parameters as well as the corresponding relative errors for two experiments using the irradiance data of figure 5.5(a). The two sets of results that appear in table 5.2 are for an implementation of the classical CE and for an implementation of the FICE.

It does not come as a surprise that the number of iterations (1648) required to converge to the right solution is larger than the equivalent planar patch case since the problem is of higher dimension, 11 parameters to recover instead of 8, and the nonlinear part of the system is numerically far more complex. In addition, the cost of solving the nonlinear system for the structure parameters, $\hat{\mathbf{n}}_0$ and $\mathbf{d}_0$ is about 14 times the cost of solving the nonlinear equation for the parameter $\hat{\mathbf{n}}_0$ in the planar patch case.

The accuracy of the estimates for the FICE experiment, less than 3% error, is very good

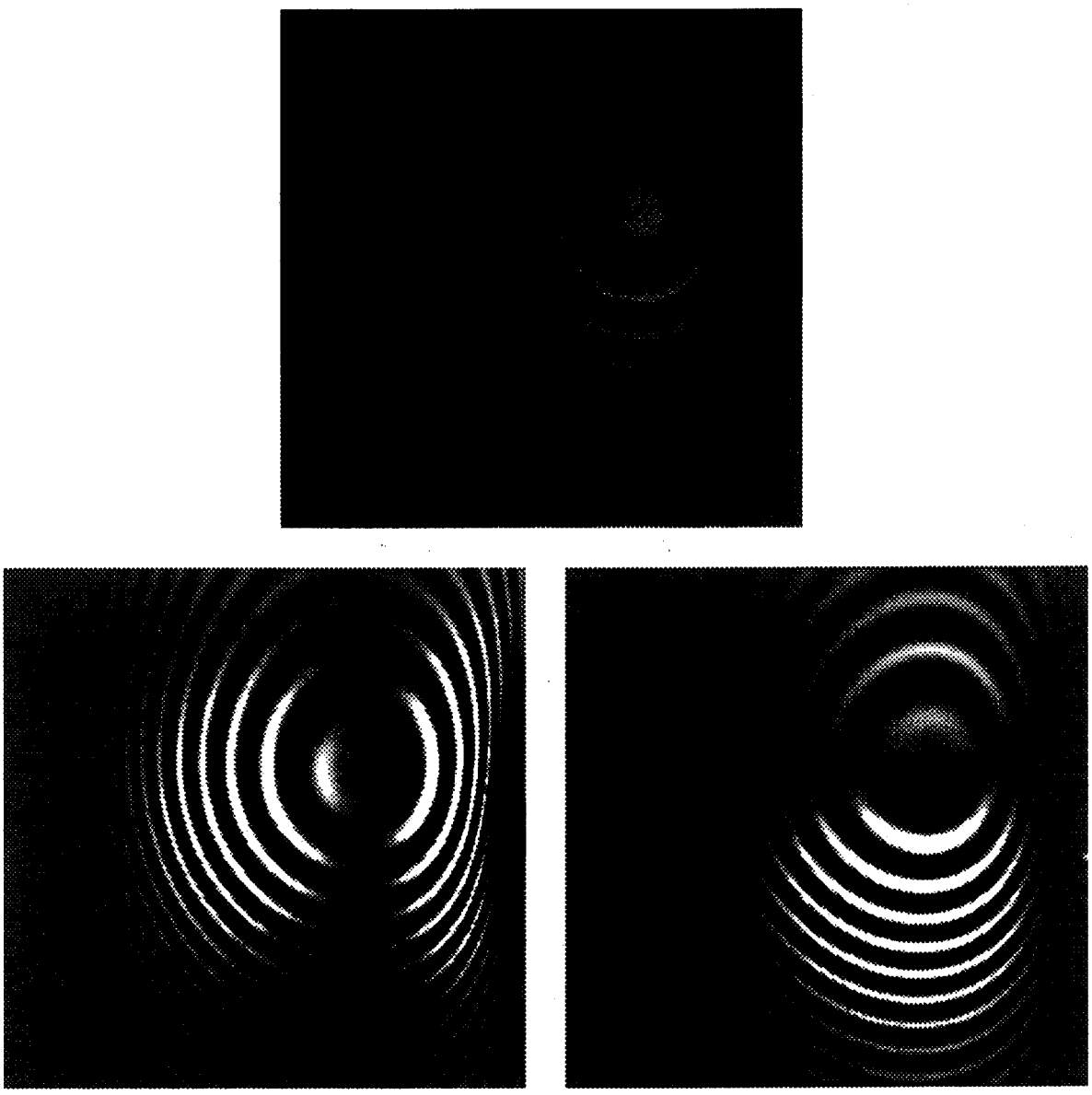

Figure 5.5: Exponentially damped zone plate texture for a saddle surface defined in table 5.1. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (bottom left and right) the x– and y–gradients.

considering that only the irradiance data were synthesized (in floating point accuracy) and the spatiotemporal gradients were estimated from these irradiance data. The results for the CE implementation have a much higher bias than in the planar patch case and the relative errors range from 11 to 18%. These results are not really surprising because the irradiance data for a quadratic patch exhibits a strong spatial shading, induced by the distant source, across its surface and the temporal shading variations are far more substantial than in the planar patch case. The strong shading effects are best seen on figure 5.6(a) for a textureless surface.

Experiments with eight bit quantized data were not very impressive because of the strenuous requirement of providing a fairly good initial structure estimate in order to converge to the correct solution. In the case of the quadratic patch with a distant light source and quantized data, the CE implementation did not provide sufficiently good estimates, i.e. within 20 to 25% of the real values, except in the case of very strongly marked surfaces. Similarly, experiments with real data were inconclusive because the uncertainties on the true measured motion and structure parameters were so large, that the relative errors between the final estimates and the measured values of the parameters were totally meaningless.

5.2.2 Textureless Surface

In this experiment the same geometric saddle surface, defined by the parameters of table 5.1, is used but the albedo is now constant on the surface. Figure 5.6 shows the irradiance and the spatiotemporal gradients for the textureless surface. It can be observed that the shading spatial gradients are stronger for the textureless quadratic patch case than for the textureless planar patch case. This situation should stimulate a faster convergence to the correct solution and increase the region of convergence i.e. the algorithm for the quadratic patch case has the potential of being less sensitive to the initial estimate than the corresponding planar patch algorithm. Unfortunately, the example in the previous section demonstrates the need for strong spatial gradients to insure convergence for arbitrary initial values for the structure parameters and a textureless quadratic surface does not provide gradients with enough contrast. The example of figure 5.6 did not converge to the correct solution for the canonical initial values ($\mathbf{n}_0 = (0\ 0\ 1)$ and $\mathbf{d}_0 = \mathbf{0}$) and a new initial estimate was selected. Table 5.3 displays the true parameters, the initial values and the final estimates and provides the relative errors between

Init normal Init curvat	$n_1 = 0.0$ $d_1 = -0.5$			$n_2 = 0.0$ $d_2 = 0.0$			$n_3 = 1.0$ $d_3 = 0.5$		
FICE	True parameters			Last estimates (5410)			Relative error (%)		
Rotation	−.00470	.00698	.00785	−.00478	.00674	.00794	1.78	3.41	1.15
Translation	.00468	−.00625	.00859	.00462	−.00726	.00877	1.31	1.62	2.12
Normal	.36400	−.28510	1.00000	.37520	−.29950	1.00000	3.10	2.98	
Curvature	−1.00000	.00000	1.00000	−1.02410	−.00112	1.02680	2.41	1.12	2.68

Table 5.3: Final estimates and relative errors of the motion and structure parameters using the FICE on the textureless synthetic, floating point irradiance sequence depicted in figure 5.6. All the parameters are specified by table 5.1.

the true parameters and the final estimates. The errors are more or less identical to those obtained in the textured case but the number of iterations is more than three times the number of iterations required in the textured quadratic patch case. A strict comparison in number of iterations required by the planar and quadratic patch case, for a textureless surface, as opposed to a textured surface, is impossible because, in the former case, the same initial estimate is used in the textured and textureless surface situations and, in the quadratic case, a change of initial estimates was required.

The results of the experiments with quantized data were very similar to the results of the experiments in the planar patch case: very good initial estimates are required to attain convergence to the correct solution. The problem of systematically finding the initial values is very difficult in the textureless case because the CE algorithm cannot handle textureless surfaces and is, therefore, unable to provide any useful solution.

5.3 Conclusions

The overall implementation of the quadratic patch case in the simplest possible shading case, the distant punctual source, turned out to be extremely hard and the convergence properties rather poor for a random initial estimate of the surface normal and quadrature vector and an arbitrary quadratic patch geometry. The simple, canonical initial estimate of a frontal plane i.e. $\mathbf{n_0} = (0\ 0\ 1)$ and $\mathbf{d_0} = \mathbf{0}$ failed in many cases unless the curvatures and inclination of the tangent plane at the origin were small or the surface markings very strong. The convergence

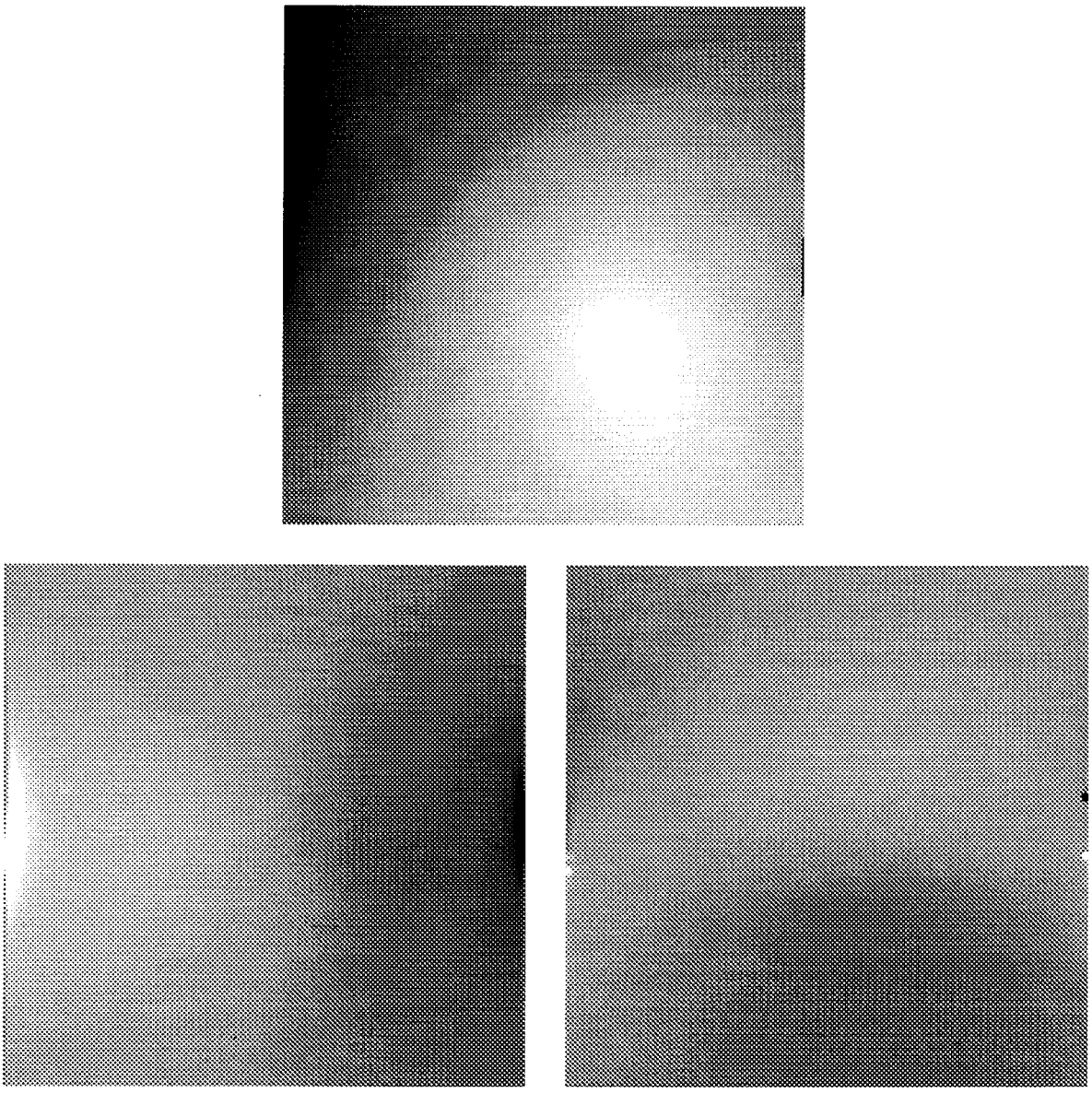

Figure 5.6: Textureless saddle surface defined by table 5.1. (a) (top left) is the irradiance image, (b) (top right) the temporal gradient, (c–d) (bottom left and right) the x– and y–gradients.

properties are worse than those of the planar patch case because the overall system is far more nonlinear and the number of unknows is higher. The nonlinear portion of the global system is no more a relatively simple nonlinear vectorial equation, but is now a full system of two vectorial equations, one of which (5.8) is very complex. The only guaranteed way of converging to the correct solution is to provide a fairly good initial estimate of the normal and curvature vectors. Negahdaripour's iterative scheme (Negahdaripour 1986), which solves the problem for the classical constraint equation, was used to obtain an initial estimate in cases where the texture gradients were strong and reasonable estimates could be computed using the CE scheme. On the other hand, very good results were obtained for both textured and textureless surfaces. In the textured cases, the FICE performances were much superior to the performances of the CE implementation and the relative error was two to three times smaller. Very nice results were also obtained in the textureless case, a case that cannot be handled by CE based algorithms.

These implementation difficulties demonstrate that the quadratic patch case is really the most complex case that can be solved using this formalism. As it was hinted before, the implementation complexity grows an order of magnitude faster than the degree of the surface patch, leading very early on to intractable systems. The main source of complexity is due to difficulty in expressing analytically the unit normal at each point of the surface and in computing its derivatives with respect to the fixed parameters of the patch (e.g. $\hat{\mathbf{n}}_0$ and $\mathbf{d}_0$). On the other hand, it is possible to recover the structure and motion parameters for a textureless surface and the parameters obtained in the textured surface case are more accurate than those determined by an algorithm that relies on the CE.

Chapter 6

Multiframe Formulation

This chapter discusses the multiframe implementation of the DFU constraint equation introduced in chapter 2 and compares it to two alternate algorithms that rely on a more classical use of the regular or full irradiance constraint equation. The performances of the various multiframe algorithms are examined with special emphasis on the relationship between the number of frames used, the amount of noise present in the irradiance sequence and the accuracy of the estimated results. The performances are evaluated for synthetic data, where noise of known variance is added and systematic comparison of performance is possible, and for a sequence of real data acquired with a video camera.

The multiframe techniques described in this chapter do not rely on the use of shading information and can be utilized with any differential technique that uses a constraint equation. Moreover, shading information and multiple frames are two different techniques that can be used separately or together in a given implementation and the performance of each one does not affect the other one. The multiple–frame schemes that are presented are not only applicable to algorithms that compute the rigid body motion and structure but also to algorithms that estimate dense optical flow. The results are more dramatic in the former case because the highly overconstrained problem provides a better noise smoothing capability than in the case of the optical flow estimation that is a far less overconstrained problem.

In this chapter, the minimization equations are presented for the full irradiance constraint equation with a distant light source model; the algorithms are run on data compatible with these assumptions. The real sequence, processed by the multiframe DFU incremental FICE,

was used in chapter 4 in the experiments with real data for a planar patch. The example will be further studied in this chapter to show how the accuracy of the estimated parameters varies when the number of frames used in the multiframe DFU algorithm is changed.

6.1 Multiframe Algorithm Implementations

The global goal in using multiple frames is to reduce the sensitivity of the algorithm to noise by using a larger amount of correlated data in the least–squares formulation. The key to this type of formulation is the derivation of a constraint equation, or of a series of constraint equations, that link the various frames so that a global parameter estimation can be performed on the augmented spatiotemporal block of data.

Three different schemes are presented in this chapter: a central frame algorithm, an incremental constraint equation (ICE) scheme and the DFU algorithm that was introduced in chapter 2 in the two frames case. All three algorithms share the assumption that the rigid body motion is constant within the temporal analysis window. More specifically, the translational velocity $\mathbf{t}$, the rate of rotation $\|\boldsymbol{\omega}\|$ and the direction and position of axis of rotation are constant. Under this assumption the rotational and translational displacements of the rigid object in frame i, *with respect* to the initial frame 0, are expressed, for a unit time step, by $\boldsymbol{\omega}^{(i)} = i\boldsymbol{\omega}$ and $\mathbf{t}^{(i)} = i\mathbf{t}$, respectively[1]. This assumption seems very restrictive but is in fact fairly benign for temporal windows of three to seven frames, in the case of ordinary motions. It will be shown later that the assumption can be somewhat relaxed to allow slowly varying motions from frame to frame.

The next three sections present the three multiframe algorithms for recovering the motion and structure parameters and are followed by a study of the performance of some of the algorithms with respect to noise. In order to simplify the mathematical expressions, the algorithms which are derived for the planar patch case, instead of the generic depth map case, use the full irradiance with the distant source model, and assume that the surface is textureless. The algorithms can be adapted to the different light source and surface models that were presented in the previous chapters at the cost of a more complex formulation. In particular, textured surface algorithm implementations are obtained by the introduction of the shading model constraint

[1] In this chapter the parenthesized superscripts refer to the frame number.

via a Lagrange multiplier function that is eliminated in a manner similar to the one employed in section 4.1.1.1.

6.1.1 Central Frame Algorithm

The central frame algorithm is a straightforward extension of the regular two–frame algorithm. Let us assume that $2n+1$ frames, indexed from time t_{-n} to t_n, are used and let us consider the $2n$ individual constraint equations CE_i, $i \in [-n,n]-\{0\}$, between the reference frame 0 and the frame i. Each constraint equation CE_i relates the same reference frame to a given frame i in the sequence and the collection of constraint equations is globally minimized, or more specifically, the integral of the sum of the squares of all the $2n$ constraint equations is minimized. The choice of the reference frame is arbitrary and any of the $2n+1$ frames could be chosen. The advantages of the central frame as reference frame is that the symmetry introduces some simplifications in the expressions and the constant motion assumption is a better approximation since the maximal temporal distance between the reference frame and any other frame in the sequence within the temporal window is n. The minimization problem can be formulated as

$$\min\left(\iint_\sigma \sum_{i=-n}^{n}\left(E_t^{(i)} - (\mathbf{v}\cdot\boldsymbol{\omega}^{(i)}) - (\mathbf{r}\cdot\hat{\mathbf{n}})(\mathbf{s}\cdot\mathbf{t}^{(i)}) - \rho[\hat{\mathbf{l}},\boldsymbol{\omega}^{(i)},\hat{\mathbf{n}}]\right)^2 d\mathbf{r} + \lambda\left(\|\hat{\mathbf{n}}\|^2 - 1\right)\right). \quad (6.1)$$

To facilitate the expression of the summation, the $k=0$ term is included because $E_t^{(0)} = 0$ and $\boldsymbol{\omega}^{(0)} = \mathbf{t}^{(0)} = \mathbf{0}$. Under the constant rigid body motion assumption, $\boldsymbol{\omega}^{(i)} = i\boldsymbol{\omega}$ and $\mathbf{t}^{(i)} = i\mathbf{t}$, and after some simple algebraic manipulations, it is apparent that this formulation and the resulting equations, obtained by differentiation of equation (6.1) with respect to the parameters $\mathbf{t}$, $\boldsymbol{\omega}$ and $\hat{\mathbf{n}}$, are similar to the one developed in section 4.1.1. For example, the equivalent linear system takes the form

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

where $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_4$ are identical to the quantity of the same name in the system (4.6) and

$$\tilde{\mathbf{e}}_1 = \iint_\sigma \left(\sum_{i=-n}^{n} iE_t^{(i)} \Big/ \sum_{i=-n}^{n} i^2\right)(\mathbf{v} + \rho(\hat{\mathbf{n}}\times\hat{\mathbf{l}}))d\mathbf{r}; \quad \tilde{\mathbf{e}}_2 = \iint_\sigma \left(\sum_{i=-n}^{n} iE_t^{(i)} \Big/ \sum_{i=-n}^{n} i^2\right)(\mathbf{r}\cdot\hat{\mathbf{n}})\mathbf{s}\,d\mathbf{r}.$$

The overall effect of the central frame minimization is the averaging of the temporal gradients over the length of the temporal window. This result is not surprising because, by design, only

the spatial gradients of the central frames are used and all the other quantities, besides the temporal gradients, are independent of the frame indices.

The only advantage of this formulation is its simplicity. The multiframe implementation is nearly identical to the two–frame implementation and the two–frame implementation is in fact the multiframe implementation for two frames. The drawbacks are numerous and serious. The formulation underuses the available data, the spatial gradients are only computed from the reference frame, and the rest of the sequence is merely used to compute the temporal gradients. This method only solves the noise problem in a weak sense because it only provides noise reduction by smoothing the gradient data in the temporal direction and it is totally ineffective in dealing with the noise present in the spatial gradients. In addition, for large motion, the smoothing of the temporal gradients can be detrimental to the convergence of the algorithm and worse results can be obtained in the multiframe case than in the two–frame case. This method was presented with the unique goal of showing how some quick improvements, over any two–frame algorithm, can be gained at the cost of very few minor changes to the two–frame implementation.

The next section describes an algorithm, called the incremental constraint equations algorithm, that fully uses the spatiotemporal gradients of each frame in the sequence. However, the algorithm leads to a system of highly nonlinear equations that are very delicate to implement.

6.1.2 Incremental Constraint Equation Algorithm

Given $n+1$ consecutive frames in a sequence, the ICE algorithm minimizes the integral of the sum of the squares of n constraint equations that link two *adjacent* frames at a time. Each constraint equation is a two–frame constraint equation, that relates the spatiotemporal gradients of two successive frames, and is, consequently, different from the constraint equation at neighboring pair of frames. Under the assumption of constant rigid body motion, the motion parameters and the intrinsic structure parameters (e.g. $\mathbf{d}_0$) are identical in each CE, but the normal to the planar patch, or to the tangent plane at the origin for higher order surfaces, is different. More specifically, in the distant source case, the minimization problem can be written

as

$$\min_{I}\left(C=\iint_{\sigma}\sum_{i=1}^{n}\left(E_t^{(i)}-(\mathbf{v}^{(i)}\cdot\boldsymbol{\omega})-(\mathbf{r}\cdot\hat{\mathbf{n}}^{(i)})(\mathbf{s}^{(i)}\cdot\mathbf{t})-\rho[\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}^{(i)}]\right)^2 d\mathbf{r}-\lambda\left(\|\hat{\mathbf{n}}\|^2-1\right)\right), \tag{6.2}$$

where $\hat{\mathbf{n}}^{(i)}=\hat{\mathbf{n}}+i(\boldsymbol{\omega}\times\hat{\mathbf{n}})$ and $\hat{\mathbf{n}}=\hat{\mathbf{n}}^{(0)}$ denotes the unit normal in the starting frame. In addition, the vector fields $\mathbf{v}^{(i)}$ and $\mathbf{s}^{(i)}$ are different in each frame i and are computed using the i^{th} frame spatial gradients.

Under the assumption of constant rigid body motion, this scheme amounts to performing n simultaneous two–frame minimizations to recover a common set of motion and structure parameters (e.g. $\boldsymbol{\omega}$, $\mathbf{t}$ and $\hat{\mathbf{n}}$ in the planar patch case). Expanding the normal $\hat{\mathbf{n}}^{(i)}$ in the minimization equation (6.2) and differentiating C with respect to the structure and motion parameters leads to the following system of three nonlinear equations

$$\begin{aligned}
\iint_{\sigma}F^{(i)}\Big(-\mathbf{v}^{(i)}+i(\mathbf{r}\times\hat{\mathbf{n}})(\mathbf{s}^{(i)}\cdot\mathbf{t})-\rho\big((\hat{\mathbf{n}}\times\hat{\mathbf{l}})+i\hat{\mathbf{n}}\times(\hat{\mathbf{l}}\times\boldsymbol{\omega})-i\hat{\mathbf{l}}\times(\boldsymbol{\omega}\times\hat{\mathbf{n}})\big)\Big)d\mathbf{r} &= 0\\
\iint_{\sigma}F^{(i)}\Big(-\mathbf{s}^{(i)}\big((\mathbf{r}\cdot\hat{\mathbf{n}})+[\mathbf{r},\boldsymbol{\omega},\hat{\mathbf{n}}]\big)\Big)d\mathbf{r} &= 0\\
\iint_{\sigma}\Big(F^{(i)}\Big(-(\mathbf{s}^{(i)}\cdot\mathbf{t})\big(\mathbf{r}+i(\mathbf{r}\times\boldsymbol{\omega})\big)-\rho\big((\hat{\mathbf{l}}\times\boldsymbol{\omega})-i(\hat{\mathbf{l}}\times\boldsymbol{\omega})\times\boldsymbol{\omega}\big)\Big)\Big)d\mathbf{r}+\lambda\hat{\mathbf{n}} &= 0,
\end{aligned}$$

where $F^{(i)}(\boldsymbol{\omega},\mathbf{t},\hat{\mathbf{n}})=E_t^{(i)}-(\mathbf{v}^{(i)}\cdot\boldsymbol{\omega})-\big((\mathbf{r}\cdot\hat{\mathbf{n}})+i[\mathbf{r},\boldsymbol{\omega},\hat{\mathbf{n}}]\big)(\mathbf{s}^{(i)}\cdot\mathbf{t})-\rho\big([\hat{\mathbf{l}},\boldsymbol{\omega},\hat{\mathbf{n}}]+i(\hat{\mathbf{l}}\times\boldsymbol{\omega})(\hat{\mathbf{n}}\times\boldsymbol{\omega})\big)$.

It can immediately be noticed that the previous equations are no longer linear in the motion parameter $\boldsymbol{\omega}$, resulting in the loss of the semilinearity of the overall system that can now only be solved by a general nonlinear global method. As a result, it should be expected that the convergence of the nonlinear system is far from being guaranteed for an arbitrary set of initial conditions and an accurate initial condition is required. However, when restricted to two frames, the system becomes semilinear, its convergence properties are improved, and the solution can generally be found for an arbitrary initial condition. Therefore, the problem is first solved in the two–frame case and its solution is used as an initial estimate to the full multiframe system, resulting in a fast convergence to a more accurate solution. The convergence is fast because the initial value is usually a very good estimate of the final solution.

The ICE algorithm fully uses all the irradiance data available in the sequence and provides a systematic way of computing the constant motion parameters and the structure parameters from a given set of frames in a sequence. Its drawbacks are twofold: the algorithm leads to a

fully nonlinear system of vectorial equations and the approach segments the temporal data into a set of n separate two–frame problems that are solved simultaneously instead of solving the $(n+1)$–frame problem directly. In this respect, the ICE algorithm methodology is similar to the one used by some nonlinear system solvers that decompose the structure of the system by considering each equation separately, and minimize the sum of the squares of the residuals of each equation instead of solving directly the initial system.

The next section presents a method that offers more flexibility, globally uses all the data, and leads to a semilinear implementation of the system of nonlinear equations.

6.1.3 Dynamical Frame Unwarping Constraint Equation

Section 2.3.3 introduced the concept of the DFU constraint equation in the case of one virtual frame and two real frames. The DFU constraint equation algorithm relies on a dynamical CE that uses a displaced frame difference, computed for the current motion estimate, in lieu of the temporal gradients, and that minimizes the square of the residuals of the DFU constraint equation over the full image. The major difference with the classical method is that the constraint equation, which is minimized at each iteration, is not static, but depends on the current estimate of the motion parameters; the spatial gradients are dynamically computed from the unwarped irradiance fields at each iteration. In effect, the DFU process takes the two irradiance fields and spatially distorts them so that a given point in the image plane is the image of the same object point at times t_1 and t_2. The unwarping process effectively undoes the motion to temporally align each object point so that it projects on the same image point throughout the sequence.

It does not take much of a change to extend the unwarping process to an arbitrary number of frames. Let us define the global displaced frames difference, $D_G(\mathbf{x}_i, t, \tau, n, \hat{\mathbf{d}}_t(\mathbf{x}_i, t))$, as the sum of the $2n$ elementary DFDs formed by the n pairs of symmetric frames around the virtual frame[2] . Using the notations that were first introduced in section 2.3.3, the global DFD, D_G, is given by

$$D_G\big(\mathbf{x}_i, t, \tau, n, \hat{\mathbf{d}}_t(\mathbf{x}_i, t)\big) = \sum_{k=0}^{n} \big(\tilde{u}(\mathbf{x}_i + (1-k\tau)\hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_+^{(k)}) - \tilde{u}(\mathbf{x}_i - k\tau\hat{\mathbf{d}}_t(\mathbf{x}_i, t), t_-^{(k)})\big)$$

[2]In practice, the central frame is a virtual frame when the number of frames is an even number but is middle, real frame for odd number of frames.

where $t_-^{(k)} = t - k\tau T_{\mathbf{r}}^t$ and $t_+^{(k)} = t - (1-k\tau)T_{\mathbf{r}}^t$, and is used to derive the multiframe DFU–FICE expressed by

$$D_G(\mathbf{x}_i, t, \tau, n, \hat{\mathbf{d}}^{(n)}) - (\hat{\mathbf{d}}^{(n)} - \hat{\mathbf{d}}_t) \sum_{k=0}^{n} \left(\mathbf{I}_3 - k\tau \frac{\partial \hat{\mathbf{d}}^{(n)}}{\partial \mathbf{x}_i} \right) \left(\nabla_{\mathbf{x}} \tilde{u}(\mathbf{x}_i + (1-k\tau)\hat{\mathbf{d}}^{(n)}, t_+^{(k)}) = \dot{E} \right. \quad (6.3)$$

in the optical flow case. Equation (6.3) is the multiframe equivalent of equation (2.30) that was derived in the two–frame case. In the rigid body motion formulation, and in the simplified case where the motion field is locally constant, translational or slowly varying, equation (6.3) takes the simpler form

$$D_G(\mathbf{x}_i, t, \tau, n, \hat{\mathbf{d}}^{(n)}) - \mathbf{V}_G(\mathbf{x}_i, t_+, \tau, n, \mathbf{d}^{(n)})(\boldsymbol{\omega}^{(n)} - \boldsymbol{\omega}) + (\mathbf{r} \cdot \hat{\mathbf{n}}) \mathbf{S}_G(\mathbf{x}_i, t_+, \tau, n, \mathbf{d}^{(n)})(\mathbf{t}^{(n)} - \mathbf{t}) = \dot{E} \quad (6.4)$$

where

$$\begin{cases} \mathbf{V}_G(\mathbf{x}_i, t_+, \tau, n, \mathbf{d}^{(n)}) &= \displaystyle\sum_{k=0}^{n} \mathbf{v}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+^{(k)}) \\ \mathbf{S}_G(\mathbf{x}_i, t_+, \tau, n, \mathbf{d}^{(n)}) &= \displaystyle\sum_{k=0}^{n} \mathbf{s}(\mathbf{x}_i, (1-\tau)\mathbf{d}^{(n)}, t_+^{(k)}) \end{cases} . \quad (6.5)$$

The multiframe equation (6.4) is identical to the two–frame equation (2.33), where the DFD is replaced by a global DFD, that is the sum of the individual DFD of a pair of frames symmetric about the virtual frame, and the $\mathbf{v}$ and $\mathbf{s}$ vector fields are replaced by the global $\mathbf{V}_G$ and $\mathbf{S}_G$ fields that are the sum of the individual $\mathbf{v}$ and $\mathbf{s}$ fields. In this respect, the multiframe formulation is formally identical to the two–frame algorithm and all the results and implementations described in the preceding chapters are directly applicable.

The DFU constraint equation provides a uniform way of dealing with sequences of frames by dynamically unwarping the individual frames of the sequence to temporally align them, stack them into a spatiotemporal block of data and estimate, in one step, the motion and structure parameters of the full block of data. The elegance of the DFU algorithm is that the two–frame case is effectively the general case, and only modest modifications are required to deal with an arbitrary number of frames. The main drawback of the multiframe method is its higher computational cost, caused in part by the computation of the unwarped gradients at each iteration. Section 2.3.3 provided a more thorough discussion of the computational cost of the method.

In the practical implementation of the multiframe algorithm, it is advisable to start the iterations with only two frames and increase the number of frames only when a decent estimate of the motion parameters is available. In fact, if no a priori estimate of the motion is known, and the iterations are started by assuming a zero motion (the most probable alternative), it is a mistake to stack several raw frames at once, especially if the motion is large. The computation of the spatial gradients that uses the full spatio-temporal analysis window is then heavily distorted; the computed gradients are grossly inaccurate and can inhibit the convergence of the algorithm to the correct solution. In practice, the multiframe algorithm is run as a two–frame algorithm up to a point where the difference between two successive estimates of the parameters is below a predefined threshold, and then the full sequence, within the temporal window, is unwarped and the full multiframe algorithm enabled[3].

6.1.4 Slowly Changing Motion

All the algorithms described in the previous sections were derived under the *constant* rigid body motion assumption and a unique set of motion parameters are computed for the set of frames belonging the temporal analysis window. This assumption can be somewhat relaxed and the algorithms modified to take into account slowly varying motion where the translation and rotation vectors can be thought of as made out of a root term, which represents the velocity at time t, and a first–order update term, which represents the incremental portion of the velocity vector between the frame at time t and the frame at time $t + \delta t$.

At a *given* time t the motion parameters are computed, using the multiframe algorithm, by assuming that the motion is constant within the spatiotemporal window of analysis. At the next time step, $t+\delta t$, the spatiotemporal window is moved by δt, and the new motion parameters are computed using the previous estimate as initial estimate. This process is particularly well suited to the DFU incremental FICE, because the incremental formulation only computes the update term and the computation can be performed very efficiently for small increments. In summary, slowly varying motion parameters are accommodated by assuming that the parameters are constant within the analysis window at a given time, and are changing slightly as the analysis

[3]In reality, as it was pointed out earlier, the unwarped sequence is never computed but the indices of each temporal arrays are shifted by the amount of estimated motion in each frame and the unwarping process is a mere computation of offset indices in the various arrays.

window is displaced temporally; an update of the motion parameters is computed at the next time frame.

The next section presents synthetic and real data examples of multiframe parameter estimations. The emphasis of the section is the study of the change in the accuracy of the estimated parameters as the number of frames and the noise are varied.

6.2 Examples

Multiframe processing is a way to increase the robustness and accuracy of the solution by providing additional redundancy to the algorithm. Under the assumption that the noise in the irradiance sequence is a zero–mean, additive process uncorrelated with the signal, the accuracy of the estimated parameters is a monotonous increasing function of the number of frames; the error between the true and estimated parameters approaches zero as the number of frames increases (at least in the FICE case, where no bias is present due to the use of the constraint equation on data which do not obey it).

These results are confirmed for synthetic images where the rigid body motion is truly constant and the noise is moderate (10 to 15 %). For high noise level, the computation of the gradients is too inaccurate and the algorithm can either diverge or converge to the wrong solution. The mode of failure and susceptibility to noise is different for the ICE and DFU algorithms. The ICE algorithm is very sensitive to noise, because of its high degree of nonlinearity, and small amounts of noise are enough to prevent convergence to the correct solution. The DFU algorithm is much more robust in the presence of noise. However, the two–frame algorithm is fired first and a high amount of noise can make it diverge or prevent it from locking on the correct solution, in which case the multiframe algorithm cannot recover from the initial incorrect unwarping of the sequence and does not converge either. It is paradoxical that the multiframe algorithm behavior in the presence of noise is conditioned by the behavior in the two–frame algorithm, but the multiframe DFU algorithm can rarely be started directly, except in cases where the motion is moderate, because the stacking of many nontemporally aligned frames can prevent it from converging to the right solution.

In the case of real data, the improvements in accuracy are only observed up to a certain point, generally five to seven frames for regular motions, and then the performance degrades.

The problem with real data is that the assumption of constant motion, within the temporal analysis window, is only a reasonable approximation for a small number of frames; when too many frames are used, the assumption breaks down and the results rapidly get worse. It was found experimentally that the maximum reasonable length of a centered temporal window is about seven frames, i.e. three frames before and after the frame under consideration. However, most of the time, five frames were enough to converge to the true solution with sufficient accuracy.

Three experiments are presented in the next sections: the first one uses the central frame scheme, the last two the DFU algorithm. The first experiment demonstrates the limits of applicability of the central frame algorithm; the second presents the results of the DFU algorithm for a synthetically generated sequence, in terms of the number of frames and of the amount of noise, and the last experiment shows the behavior of the DFU algorithm for real data.

6.2.1 Central Frame Algorithm Example

By design, this very simple scheme can only improve the accuracy of the estimated parameters on sequences where the main source of noise is in the temporal gradients and little can be achieved in terms of lowering the error in the estimates, caused by noisy spatial gradients, by using more frames. The noise in the spatial gradients is not averaged by this algorithm because only the central frame is used to compute the spatial gradients. In fact, when the noise is concentrated on the spatial gradients, the error in the estimates is approximately constant and independent of the number of frames used. The bias in the solution is caused by the noise in $E_{\mathbf{r}}$. When both the spatial and temporal gradients are noisy, the accuracy of estimates increases with the number of frames and eventually levels off at a level that represents the bias introduced by the noisy spatial gradients.

The data in this experiment are totally synthetic: the spatial gradient of a planar patch mapped with a cosine grating with a sinusoidal phase variation (see figure 4.1) is analytically generated and the temporal gradients are computed directly from the distant source FICE. The true motion parameters are $\boldsymbol{\omega} = (1.3, -.1, -1.2)10^{-2}$ radians, and $\mathbf{t} = (.25, .5, 1.25)10^{-2}$, and the unit normal is $\hat{\mathbf{n}} = (.0371, -.0371, .9278)$. These ideal gradient data are selectively corrupted by 5% additive noise in three different ways — the temporal irradiance gradients E_t

are noisy and the spatial gradients $E_{\mathbf{r}}$ noise–free — the temporal gradients are noiseless and $E_{\mathbf{r}}$ noisy — both E_t and $E_{\mathbf{r}}$ are noisy. The spatiotemporal gradients are directly input to the central frame algorithm.

Table 6.1 displays the relative errors, expressed in percentages, between the true and the estimated parameters for multiframe runs ranging from two to six frames. As expected, the accuracy of the estimates increases with the number of frames in the case of noisy temporal gradients; the relative error reaches about 10^{-3} for all the parameters when six frames are used. The performance of the algorithm is independent of the number of frames in the case where only the spatial gradients are noisy, and the computed solution exhibits a relative bias of about .5% at convergence. In the case of noisy spatiotemporal gradients, the accuracy of estimates initially increases with the number of frames then reaches a constant level, the bias level of the previous case, in about five frames.

The analysis of these results clearly shows the strong limitation of the algorithm: only temporal gradient noise can be reduced by the central frame algorithm; spatial noise is not averaged but is translated into a bias in the final estimates of the motion and structure parameters. On the other hand, the results clearly indicate the gains that are achieved by the multiframe formulation and prove that substantial performance improvement is available from these multiple–frame techniques.

6.2.2 DFU Algorithm: Synthetic Data

This experiment focuses on the measure of the accuracy of the estimates as a function of the number of frames and of the noise level. The irradiance data (see figure 4.1) are generated in floating point, then zero–mean, noncorrelated noise is added and the noisy data are quantized to eight bits. The synthetic sequence is generated from a perfect constant rigid body motion, that is given in table 4.1, along with the structure parameters. The irradiance sequence is the only input to the DFU algorithm and the spatial gradients are dynamically estimated from the unwarped, quantized irradiance frames.

In this experiment, the multiframe algorithm is started as a two-frame algorithm and runs as such until the relative error between two successive estimates is less than 20%, at which point it is switched to the multiframe mode. The 20% threshold is rather arbitrary. It was determined

5% noise in E_t, 0% noise in $E_{\mathbf{r}}$									
# frames	Rel. rotation error			Rel. translation error			Rel. normal error		
2	2.96	2.94	1.82	1.02	7.53	4.95	3.17	3.07	7.01
3	0.92	0.92	1.10	1.02	2.51	1.28	0.94	0.94	1.65
4	0.53	0.53	0.16	0.70	1.54	0.69	0.54	0.54	1.41
5	0.18	0.18	0.07	0.08	0.39	0.32	0.18	0.18	1.58
6	0.09	0.09	0.07	0.00	0.17	0.19	0.09	0.09	1.43

0% noise in E_t, 5% noise in $E_{\mathbf{r}}$									
# frames	Rel. rotation error			Rel. translation error			Rel. normal error		
2	0.22	0.22	0.00	0.21	0.65	0.33	0.23	0.24	1.68
3	0.44	0.44	0.02	0.65	1.43	0.53	0.44	0.42	1.64
4	0.56	0.56	0.00	0.65	1.43	0.78	0.55	0.54	1.63
5	0.21	0.21	0.00	0.20	0.52	0.31	0.20	0.10	1.61
6	0.38	0.38	0.01	0.48	1.01	0.50	0.37	0.36	1.60

5% noise in E_t, 5% noise in $E_{\mathbf{r}}$									
# frames	Rel. rotation error			Rel. translation error			Rel. normal error		
2	2.58	2.58	2.52	1.21	2.4	5.4	7.83	7.86	7.30
3	1.27	1.27	1.41	1.03	1.43	1.64	1.43	1.43	1.59
4	0.64	0.64	0.54	0.63	1.01	0.64	0.94	0.95	1.40
5	0.33	0.33	0.36	0.45	0.34	0.32	0.54	0.53	1.52
6	0.32	0.33	0.27	0.51	0.98	0.38	0.41	0.42	1.57

Table 6.1: Relative errors, expressed in percentage, of the motion and structure parameters as a function of the number of frames for the noisy synthetic irradiance sequence depicted in figure 4.1 processed by the central frame algorithm.

by trial and error and represents a safe level at which the unwarping of the full sequence by the current motion estimate does not prevent the algorithm converging to the correct solution. Three levels of noise were selected—1%, 5% and 10%. Table 6.2 reports the results, which are expressed in percentages of relative error between the final estimates and the true value of the parameters, for the three noise levels and for a variable number of frames (2–7). Globally, it can be observed that the accuracy steadily increases with the number of frames and that this increase is particularly dramatic for the noisiest sequence, where the initial gain is the largest. In general, the gain in accuracy obtained from switching from two frames to three is much bigger than the incremental gain achieved by adding additional frames. The results suggest that, for a large number of frames, the errors is going to zero. Unfortunately, in most cases, the motion constancy assumption is met for at most seven frames; it should not be expected that the error will be zero for seven frames in the case of moderately noisy data.

This experiment shows that the overall multiframe DFU algorithm is very robust against noise and that excellent performance can be achieved in cases where the rigid body motion is constant. However, we should be aware that the results on the accuracy of the estimated parameters obtained with these synthetic data represents an upper bound on the gain in performances that can be achieved for noisy quantized data. Although the DFU algorithm was run on the synthetic data in a realistic way, i.e. the noisy, quantized irradiance data were fed directly to the algorithm, these data are still ideal. In particular, the motion is rigorously constant, the shading is analytically computed and the added noise is very controlled. Real motion is rarely constant; even for a short temporal analysis window, the shading equation is, at best, a decent approximation of the real situation and the noise process is neither truly zero–mean, nor purely white and additive and worst results should be expected in the real data case.

The next section presents the results of the same algorithm applied to real data acquired by a video camera.

6.2.3 DFU Algorithm: Real Data

The data in this experiment are those used and described in section 4.2.1.2. In particular, it should be recalled that the measured parameters from the set–up were not very accurate and that the final estimates, presented in table 4.6, were ultimately corrected for errors in the focal

1% noise in irradiance image								
# frames	Rel. rotation error			Rel. translation error			Rel. normal error	
2	0.81	2.62	0.45	2.41	0.73	1.13	4.61	4.72
3	0.32	1.21	0.38	1.13	0.26	0.68	1.99	2.02
4	0.11	0.46	0.38	1.13	0.26	0.68	1.99	2.02
5	0.06	0.21	0.05	0.25	0.06	0.11	0.69	0.54
6	0.01	0.09	0.00	0.11	0.02	0.07	0.19	0.23
7	0.00	0.01	0.00	0.03	0.00	0.01	0.06	0.11

5% noise in irradiance image								
# frames	Rel. rotation error			Rel. translation error			Rel. normal error	
2	1.41	3.99	1.31	3.67	1.84	2.45	6.51	6.49
3	0.62	1.42	0.51	1.45	0.72	1.04	2.42	2.36
4	0.29	0.61	0.32	0.84	0.29	0.67	1.03	1.07
5	0.14	0.39	0.21	0.52	0.12	0.41	0.67	0.69
6	0.08	0.21	0.13	0.34	0.09	0.29	0.31	0.32
7	0.06	0.10	0.08	0.15	0.07	0.19	0.17	0.18

10% noise in irradiance image								
# frames	Rel. rotation error			Rel. translation error			Rel. normal error	
2	2.64	5.75	2.47	6.12	3.42	4.89	8.69	8.71
3	1.04	2.12	1.09	2.90	1.42	2.17	3.12	3.10
4	0.63	1.31	0.67	1.72	0.80	1.19	2.15	2.13
5	0.41	0.77	0.41	0.85	0.57	0.69	1.41	1.42
6	0.30	0.48	0.27	0.49	0.30	0.37	0.83	0.84
7	0.19	0.28	0.15	0.28	0.17	0.17	0.41	0.41

Table 6.2: Relative errors, expressed in percentages, of the motion and structure parameters as a function of the number of frames for the noisy synthetic irradiance sequence depicted in figure 4.1 processed by the DFU incremental FICE algorithm. The true motion and structure parameters are given by table 4.1.

# frames	Rel. rotation error			Rel. translation error			Rel. normal error		Rel. avg. error
2	11.4	26.6	46.1	39.7	43.5	13.2	18.1	16.2	26.9
3	6.2	14.1	25.9	21.3	22.7	7.1	11.3	9.4	14.7
4	4.1	12.7	22.7	18.8	19.1	5.9	7.7	5.2	12.0
5	3.1	8.1	18.0	14.1	16.4	4.1	5.4	3.4	9.1

Table 6.3: Adjusted relative errors, expressed in percentages, of the motion and structure parameters as a function of the number of frames for the real irradiance sequence shown in figure 4.4, processed by the DFU incremental FICE algorithm. The true motion and structure parameters are described in table 4.6.

length of the camera, the distance between the projection center and the object and the position of center of rotation. As such, less attention should be focussed on the actual individual values of relative error and more on the global trend that is given by the average relative size of the errors as a function of the number of frames. In the case of real data, the exact relative error figures for the various parameters are nearly meaningless because the true parameters are not known with enough precision to perform a direct comparison with the actual parameters, but the performance of the algorithm can still be judged by the average relative accuracy of the algorithm with respect to the number of frames used in the estimation.

Table 6.3 presents the adjusted relative errors between the measured and estimated parameters, i.e. the relative error in the parameters once the correction to these final estimates is computed. The improvement in performance is dramatic and the average relative error drops from 27% in the two–frame case to about 9% in the five frame case. This example illustrates the necessity of a multiframe algorithm when dealing with real data as opposed to synthetic data or synthetic gradients data and clearly demonstrates the substantial gains in accuracy that can be achieved. It is fair to state that most two–frame algorithms that are described in the literature and only tested for synthetic gradient data could not be used directly for real data.

6.3 Conclusions

This chapter presented several multiframes techniques that are, intrinsically, independent of the type of constraint equation used (CE or FICE) and are applicable to rigid body motion

and structure estimation or to optical flow computation problems.

The most attractive method was found to be the DFU incremental algorithm; because it can estimate all the motion and structure parameters of a full block of spatiotemporal data at once, it leads to an efficient, semilinear implementation; its complexity is independent of the number of frames and it can very efficiently compute the incremental motion update terms to deal with slowly varying motion parameters.

The most severe restriction in these algorithms is the assumption of constant rigid body motion within the temporal analysis window; in practice, this assumption limits the useful number of simultaneous frames to five or seven. Experimental results on synthetic data suggest that, for moderate amount of noise, more than five to seven frames are needed fro the algorithm to converge to the exact solution. The underlying limitation of these multiple–frame algorithms is that they use a small number of frames at a time, i.e. the memory of the system is limited in practice to five to seven frames. A state variable approach, like the Kalman filtering approach, does not have these restrictions and might be more appropriate for dealing systematically with multiple frames.

Chapter 7

Summary and Conclusions

This thesis addresses the problem of recovering the shape and motion of patches moving with respect to a fixed camera and fixed light source. The primary goal is to recover the motion and structure parameters of a moving patch directly from the time–varying image sequence using motion and shading cues without either computing the optical flow or establishing correspondences between features in the image sequence. The primary goal consists of three subgoals, the use of shading information, the relevance of shading information, and the use of multiple frames.

This thesis establishes a theoretical and computational framework for incorporating a priori knowledge of shading conditions and for using it, in conjunction with motion cues, to compute the motion and structure parameters of a generic patch. More specifically, a class of constraint equations that link the spatiotemporal gradients of the irradiance of a sequence of images to the temporal variations of shading is derived for arbitrary depth functions. The generic surfaces are specialized to polynomial patches in the minimization equations in order to express the depth functions in terms of a few structure parameters that fully define the patches. Further specializations to planar and quadratic patches are carried out later in the implementation examples.

The advantage of the new class of constraint equations is twofold: it provides a more accurate solution to problems where surface irradiance changes are due to motion, and it provides a solution for textureless surfaces. Under the classical CE, which assumes that the irradiance of a a given point does not change with motion, only highly textured surfaces can be used, and

the solution is only approximate in cases where the classical CE is not a good approximation.

The minimization equations are derived in a wide variety of cases and a specific implementable set of equations is given for different combinations of albedo (arbitrary and constant), source types (collimated and extended), source positions (distant or nearby), surface geometry (planar and quadratic) and irradiance models (general Lambertian and attenuated Lambertian models). The minimization equations form a system of three or more vectorial nonlinear equations in the motion and surface parameters. Several semilinear and globally nonlinear procedures are described and their relative performances discussed. Perfect results, in terms of the accuracy, are obtained in the cases of ideal synthetic data for both textured and textureless surfaces. These results also demonstrate that the new CE performs substantially better than implementations based on the classical CE, in cases where the classical CE is only an approximation, and that the new method is able to recover the motion and structure parameters for textureless surfaces, a case that cannot be solved by implementations of the classical CE. Very good results are obtained for quantized synthetic images from which the spatiotemporal gradients are estimated, and promising performance was achieved for real data, although it was impossible to rigorously assess the accuracy of the results due to the uncertainty in measurements of the experimental parameters.

The new results are not obtained at a trivial computational cost. The formulation is over an order of magnitude more complicated than the classical CE case and no closed–form solution exists even in the planar case. The convergence is typically very slow, requiring as many as 16000 iterations for textureless surfaces, and the system does not automatically converge to the correct solution given an arbitrary initial value. However, those characteristics are shared with iterative implementations of the CE, although the tolerance is tighter in the FICE case due to the high order of the nonlinearities. As a result, and as a consequence of the much higher complexity of the implementation and much higher computational cost, the full FICE implementation should only be used in cases where it is critical to determine the most accurate estimate possible or when the surface markings are too weak for reliable use of a less expensive method. In particular, this method is inappropriate in a feedback loop system, where only an approximate solution is required, and the additional computational burden is not justified.

Secondly, the thesis discusses, qualitatively and quantitatively, the importance and relevance

of the shading information in the case of an object moving with respect to a light source. In particular, the CE is evaluated analytically to determine its validity and the error as the texture scale and the magnitude of rigid motion vary. The approximation was found to worsen as the rotational component of the motion increased, even when the surface markings are strong and there is high contrast at high frequencies. The classical CE is a very good approximation for highly textured surfaces in the passive navigation case, because the objects are fixed with respect to the light sources and there is no motion induced shading. On the other hand, it is, at best, only a fair approximation in cases where the object moves with respect to the camera, especially for motion where the rotational motion is substantial.

Thirdly, the thesis proposes an efficient multiple–frame algorithm that allows the global estimation of the motion and structure from a block of frames. The task is achieved by considering a constraint equation which directly links a block of frames by stacking the unwarped frames into a single spatiotemporal block of data and by globally minimizing the data. The method assumes that the rigid motion is constant within the spatiotemporal block, but this assumption can be relaxed to allow for situations where the motion is varying slowly. The DFU multiframe implementation of the FICE is found to be very efficient, because a lot of the quantities required to estimate the spatiotemporal gradients can be precomputed for each frame and the unwarping operation simply amounts to temporally summing the offset frames of coefficients (the offsetting or unwarping operations are carried out directly on the arrays, by index calculations, at the time of the summation). The experiments show that a substantial gain in performance is obtained by the multiple–frame algorithm when dealing with noisy irradiance sequences and further demonstrate that only a multiframe method is viable for estimating the motion and structure parameters of real data. This highly overconstrained nature of the problem allows for a much more robust solution and provides for the required noise reduction.

A key assumption in the multiframe algorithm is the assumption of constant motion within the spatiotemporal analysis window. For all practical purposes, this assumption limits the number of frames to perhaps five or seven which may at times not be enough for very noisy sequences in order to sufficiently neutralize the ill effects of the noise. An alternate and more desirable solution is to summarize the entire past of the sequence by state vectors that are updated when a new frame is added to the sequence. The Kalman filtering approach provides

such a framework and is under investigation by other researchers.

One important issue that has not been addressed in this thesis is the problem of solution uniqueness. It has been shown (Negahdaripour 1986, Negahdaripour 1987) that a motion vision problem can have at most three solutions; only planar surfaces and quadratic surfaces with negative or zero Gaussian curvature can give rise to ambiguity, and then only under very special circumstances. The introduction of the shading component does not affect the multiple solution findings and was therefore not discussed in this thesis.

Another issue that was left aside is the image segmentation problem. In many situations, the scene consists of both stationary and independently moving objects and it is necessary to segment the image into regions of *similar* velocity before attempting to recover the motion and structure parameters. The segmentation and motion estimation processes can be simultaneous, as in the procedure adopted by Adiv (1985), or it can be two separate stages in which the segmentation and motion and structure parameters processes are independent. This issue is totally independent of the shading problem and is found, in exactly the same terms, in traditional motion estimation that does rely on the classical CE. Since this thesis concentrates on developing a new class of algorithms, problems common with earlier approaches were not examined in detail.

In conclusion, we develop a new generalized multiframe constraint equation that combines shading and motion and provides implementation for various cases. The solutions that are obtained are more accurate than the solutions based on the use of the classical CE and the new algorithms allow the computation of the motion and structure parameters for textured *and* textureless surfaces. The study made apparent the complexity of formulations that use both the shading and motion cues and that lead to systems of vectorial nonlinear equations which are difficult to implement. This complexity is the price of more general and more accurate solutions. In addition to new algorithms, the thesis carefully examines the classical CE approximation and improves the understanding of the tradeoffs in its use. The practical choice might be to stay with less expensive, less accurate methods, but it is important to be able to pick the best algorithms given a desired accuracy and computational complexity.

Appendix A

Finite Motion and Instantaneous Motion Optical Flow Equations

This appendix derives the optical flow equations for the finite motion equation and compares them to the instantaneous optical flow equations shown in section 2.2.2.

Let $\mathcal{R}$ denote a rotation matrix i.e. $\mathcal{R}\mathcal{R}^T = \mathbf{I}_3$ and $\mathbf{T}$ a translation vector. The finite motion of a point $\mathbf{R}$ of rigid body is defined by the equation

$$\mathbf{R}' = \mathcal{R}\mathbf{R} + \mathbf{T} \tag{A.1}$$

Let $\mathbf{r}$ and $\mathbf{r}'$ represent the perspective projections of the points $\mathbf{R}$ and $\mathbf{R}'$ onto the image plane, then

$$\begin{aligned} \delta\mathbf{r} = \mathbf{r}' - \mathbf{r} &= F\left(\frac{X'}{Z'} - \frac{X}{Z}\right) \\ &= F\left(\frac{Z}{Z'}\mathcal{R} - \mathbf{I}_3\right)\mathbf{r} + \frac{F}{Z'}\mathbf{T} \end{aligned}$$

but

$$Z' = \left((\mathcal{R}\mathbf{R} + \mathbf{T}) \cdot \hat{\mathbf{z}}\right) = \frac{1}{Z}(Z(\mathcal{R}\mathbf{r} \cdot \hat{\mathbf{z}}) + F\mathbf{T}_z)$$

hence

$$\delta\mathbf{r} = \frac{FZ}{Z\mathcal{R}\mathbf{r} \cdot \hat{\mathbf{z}} + F\mathbf{T}_z}\left(\left(\mathcal{R} - \frac{(\mathcal{R}\mathbf{r} \cdot \hat{\mathbf{z}})}{F}\mathbf{I}_3\right)\mathbf{r} + \frac{1}{Z}(F\mathbf{T} - \mathbf{r}\mathbf{T}_z)\right)$$

or

$$\delta\mathbf{r} = \frac{1}{Z(\mathcal{R}\mathbf{r} \cdot \hat{\mathbf{z}})/F + \mathbf{T}_z/Z}\left(\left(\mathbf{I}_3 - \frac{\mathbf{r}\hat{\mathbf{z}}^T}{F}\right)\mathcal{R}\mathbf{r} + \frac{1}{Z}(F\mathbf{T} - \mathbf{r}\mathbf{T}_z)\right). \tag{A.2}$$

Equation A.2 represents the finite optical flow equation derived from the finite rigid motion equation A.1. The instantaneous optical flow equation can be inferred from the finite motion optical flow equation A.2 by assuming that the rotation matrix has small rotation angles i.e. $\mathcal{R} \approx \mathbf{I}_3 + \mathbf{\Omega}\delta t$ (assuming a constant rate of rotation) and by assuming that the component of the translation along the optical axis is small relative to the distance of the object from the camera i.e. ($\mathbf{T}_z \ll 1$). Under these assumptions

$$\frac{(\mathcal{R}\mathbf{r} \cdot \hat{\mathbf{z}})}{F} + \frac{\mathbf{T}_z}{Z} \approx 1 - \frac{1}{F}(x\omega_y + y\omega_x)\delta t$$

and

$$\left(\mathbf{I}_3 - \frac{\mathbf{r}\hat{\mathbf{z}}^T}{F}\right)\mathcal{R}\mathbf{r} \approx (\mathbf{I}_3 - \frac{\mathbf{r}\hat{\mathbf{z}}^T}{F})\mathbf{\Omega}\mathbf{r}\delta t$$

With these approximations, the finite motion optical flow equation A.2 becomes

$$\frac{\delta \mathbf{r}}{\delta t} = \left(\mathbf{I}_3 - \frac{\mathbf{r}\hat{\mathbf{z}}^T}{F}\right)\mathbf{\Omega}\mathbf{r} + \frac{1}{Z}(F\mathbf{V} - \mathbf{r}\mathbf{V}_z)$$

where $\mathbf{V}$ represents the translational velocity ($\mathbf{T} = \mathbf{V}\delta t$ for a constant velocity $\mathbf{V}$). As δt decreases to zero, $\frac{\delta \mathbf{r}}{\delta t}$ tends to $\dot{\mathbf{r}}$, the optical flow, and equations A.2 and 2.9 are identical.

Appendix B

Gradients and Hessians at Neighboring Points

In this section, we will derive the relationship between the spatial gradients (and Hessian) of two corresponding points in two frames. In particular, we show that the gradients (and Hessians) are approximately equal in the two fields under some conditions.

Lets us denote by $\mathbf{d}(\mathbf{x},t) = (u(t)\,, v(t))^T$ the displacement field at the point $\mathbf{x}$ between the irradiance field $E(\mathbf{x},t_1) = E_1(\mathbf{x})$ at time t_1 and the irradiance field $E(\mathbf{x},t_2) = E_2(\mathbf{x})$ at time t_2. The displacement field between the two irradiance frames over the domain D is defined by the equation:

$$E(\mathbf{x}+\mathbf{d}(\mathbf{x},t_1),t_1) = E(\mathbf{x},t_2) \qquad \text{for} \quad \forall \mathbf{x} \in D \tag{B.1}$$

If the displacement field is constant over a neighborhood N of $\mathbf{x}$, we have the relationships

$$\begin{aligned} \nabla_{\mathbf{x}} E_1(\mathbf{x}+\mathbf{d}) &= \nabla_{\mathbf{x}} E_2(\mathbf{x}) \\ \nabla\nabla_{\mathbf{x}} E_1(\mathbf{x}+\mathbf{d}) &= \nabla\nabla_{\mathbf{x}} E_2(\mathbf{x}) \end{aligned} \tag{B.2}$$

for all $\mathbf{x}$ in the neighborhood N and we can expect a similar approximate relationship for a "nearly" constant or slowly varying displacement field in the neighborhood N.

Proposition B.1 *Let $E_1(\mathbf{x})$ and $E_2(\mathbf{x})$ be two vector fields, twice differentiable, and $\mathbf{d}(\mathbf{x})$ a vectorial disparity field, also twice differentiable, defined by the relation $E_1(\mathbf{x}+\mathbf{d}(\mathbf{x})) = E_2(\mathbf{x})$.*

The spatial gradients and Hessian are given by

$$\nabla_{\mathbf{x}} E_2(\mathbf{x}) = \nabla_{\mathbf{x}} E_1(\mathbf{x}) + \left(\frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{x}}\right)^T \nabla_{\mathbf{x}} E_1(\mathbf{x}) \tag{B.3}$$

and

$$\nabla\nabla_{\mathbf{x}} E_2(\mathbf{x}) = \nabla\nabla_{\mathbf{x}} E_1(\mathbf{x}) + \left(\frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{x}}\right)^T \nabla_{\mathbf{x}} E_1(\mathbf{x}) + \nabla\nabla_{\mathbf{x}} u \frac{\partial E_1(\mathbf{x})}{\partial x} + \nabla\nabla_{\mathbf{x}} v \frac{\partial E_1(\mathbf{x})}{\partial y} \tag{B.4}$$

Proof: Taking the partial derivatives of equation B.1 and applying the chain rule yields

$$\begin{aligned} \frac{\partial E_2}{\partial x} &= \frac{\partial E_1}{\partial x}(1+u_x) + \frac{\partial E_1}{\partial y} v_x \\ \frac{\partial E_2}{\partial y} &= \frac{\partial E_1}{\partial x} u_y + \frac{\partial E_1}{\partial y}(1+v_y) \end{aligned} \tag{B.5}$$

where u_x and u_y denote the partial derivative of the scalar u with respect to x and y, and similarly for v. Rearranging the terms and using the gradient vector gives, i.e.

$$\nabla_{\mathbf{x}} E_2 = \nabla_{\mathbf{x}} E_1 + \begin{pmatrix} u_x & v_x \\ u_y & v_y \end{pmatrix} \nabla_{\mathbf{x}} E_1,$$

that can be rewritten in full matrix notation as equation B.3.

Differentiating equation B.5 with respect to x and y we get

$$\begin{aligned} \frac{\partial^2 E_2}{\partial x^2} &= \frac{\partial^2 E_1}{\partial x^2}(1+u_x) + \frac{\partial E_1}{\partial x} u_{xx} + \frac{\partial^2 E_1}{\partial xy} v_x + \frac{\partial E_1}{\partial y} v_{xx} \\ \frac{\partial^2 E_2}{\partial xy} &= \frac{\partial^2 E_1}{\partial xy}(1+u_x) + \frac{\partial E_1}{\partial x} u_{xy} + \frac{\partial^2 E_1}{\partial y^2} v_x + \frac{\partial E_1}{\partial y} v_{xy} \\ \frac{\partial^2 E_2}{\partial xy} &= \frac{\partial^2 E_1}{\partial x^2} u_y + \frac{\partial E_1}{\partial x} u_{xy} + \frac{\partial^2 E_1}{\partial xy}(1+v_y) + \frac{\partial E_1}{\partial y} v_{xy} \\ \frac{\partial^2 E_2}{\partial y^2} &= \frac{\partial^2 E_1}{\partial x^2} u_y + \frac{\partial E_1}{\partial x} u_{yy} + \frac{\partial^2 E_1}{\partial xy}(1+v_y) + \frac{\partial E_1}{\partial y} v_{yy} \end{aligned} \tag{B.6}$$

Equation B.6 can be written in vector form,

$$\nabla\nabla_{\mathbf{x}} E_2 = \begin{pmatrix} 1+u_x & v_x \\ u_y & 1+v_y \end{pmatrix} \nabla\nabla_{\mathbf{x}} E_1 + \nabla\nabla_{\mathbf{x}} u \frac{\partial E_1}{\partial x} + \nabla\nabla_{\mathbf{x}} v \frac{\partial E_1}{\partial y} \tag{B.7}$$

which is equivalent to equation B.4.

If we assume that $\|\nabla_{\mathbf{x}} u\|$ and $\|\nabla_{\mathbf{x}} v\|$ are small, i.e. $|u_x|$, $|u_y|$, $|v_x|$, $|v_y| \ll 1$, then equation B.3 simplifies to $\nabla_{\mathbf{x}} E_1(\mathbf{x}) \approx \nabla_{\mathbf{x}} E_2(\mathbf{x})$. If, in addition, if we assume that $\|\nabla\nabla_{\mathbf{x}} u\|$ and $\|\nabla\nabla_{\mathbf{x}} v\|$ are small, i.e. all the second–order components of the disparity field are small, then equation B.4 simplies to $\nabla\nabla_{\mathbf{x}} E_1(\mathbf{x}) \approx \nabla\nabla_{\mathbf{x}} E_2(\mathbf{x})$. The previous conditions on the disparity field $\mathbf{d}(\mathbf{x})$ are met when $\mathbf{d}(\mathbf{x})$ is locally constant, translational or slowly varying.

Appendix C

Matrix A and Stencil Computation

This appendix symbolically computes the matrix **A** in terms of the orthogonal basis functions for a general spatiotemporal window, and presents simplified expressions in the cases of symmetric spatial windows and symmetric spatiotemporal windows.

The symmetric matrix **A** is defined by $\mathbf{A} = \sum_{W} \Psi\Psi^T$ where Ψ is the vector of orthogonal basis functions, $\Psi^T = (1 \;\; x \;\; y \;\; t \;\; xt \;\; yt \;\; xy \;\; x^2 \;\; y^2)$ and W represents the spatio-temporal analysis window. The explicit summation indices will be omitted in the following development in order to simplify the equations.

For a general window W, the symmetric matrix **A** is given by

	1	x	y	t	x^2	y^2	xy	xt	yt
1	$\sum 1$								
x	$\sum x$	$\sum x^2$							
y	$\sum y$	$\sum xy$	$\sum y^2$						
t	$\sum t$	$\sum xt$	$\sum yt$	$\sum t^2$					
x^2	$\sum x^2$	$\sum x^3$	$\sum x^2y$	$\sum x^2t$	$\sum x^4$				
y^2	$\sum y^2$	$\sum xy^2$	$\sum y^3$	$\sum y^2t$	$\sum x^2y^2$	$\sum y^4$			
xy	$\sum xy$	$\sum x^2y$	$\sum xy^2$	$\sum xyt$	$\sum x^2y$	$\sum xy^3$	$\sum x^2y^2$		
xt	$\sum xt$	$\sum x^2t$	$\sum xyt$	$\sum xt^2$	$\sum x^3t$	$\sum xy^2t$	$\sum x^2yt$	$\sum x^2t^2$	
yt	$\sum yt$	$\sum xyt$	$\sum y^2t$	$\sum yt^2$	$\sum x^2yt$	$\sum y^3t$	$\sum xy^2t$	$\sum xyt^2$	$\sum y^2t^2$

Since most windows are spatially symmetric around the central point, the matrix **A** is sparse

and is given by

	1	x	y	t	x^2	y^2	xy	xt	yt
1	$\sum 1$								
x	0	$\sum x^2$							
y	0	0	$\sum y^2$						
t	$\sum t$	0	0	$\sum t^2$					
x^2	$\sum x^2$	0	0	$\sum x^2 t$	$\sum x^4$				
y^2	$\sum y^2$	0	0	$\sum y^2 t$	$\sum x^2 y^2$	$\sum y^4$			
xy	0	0	0	0	0	0	$\sum x^2 y^2$		
xt	0	$\sum x^2 t$	0	0	0	0	0	$\sum x^2 t^2$	
yt	0	0	$\sum y^2 t$	0	0	0	0	0	$\sum y^2 t^2$

If the three windows (horizontal, vertical and temporal) are symmetric around the central point, and the spatial extend of the two spatial windows is equal to N and the length of the temporal window is T, the representation of the matrix **A** can be further simplified to

	1	x	y	t	x^2	y^2	xy	xt	yt
1	N^2T								
x	0	$N'\sum x^2$							
y	0	0	$N'\sum x^2$						
t	$\sum t$	0	0	$\sum t^2$					
x^2	$N'\sum x^2$	0	0	$N'\sum x^2 t$	$N'\sum x^4$				
y^2	$N'\sum x^2$	0	0	$N'\sum x^2 t$	$M\sum x^4$	$N'\sum x^4$			
xy	0	0	0	0	0	0	$M\sum x^4$		
xt	0	$N'\sum x^2 t$	0	0	0	0	0	$N'\sum x^2 t^2$	
yt	0	0	$N'\sum x^2 t$	0	0	0	0	0	$N'\sum x^2 t^2$

(C.1)

where $N' = 4N$ and $M = 4(N/2+1)$. It should be noted that the summations in equation C.1 are *one–dimensional summations* along the horizontal or temporal axis, as appropriate, unlike the summations in the previous equations that were three–dimensional. These sparse symmetric matrices have also sparse symmetric inverses that can be computed easily.

Equation C.1 can be used to compute the matrices for different symmetric windows. The

numerical values of the $\mathbf{A}$ and $\mathbf{A}^{-1}$ are given below for the $3 \times 3 \times 2$ and $5 \times 5 \times 2$ symmetric windows cases.

$$\mathbf{A}_{3\times3\times2} = \begin{pmatrix} 18 & 0 & 0 & 9 & 12 & 12 & 0 & 0 & 0 \\ 0 & 12 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 6 \\ 9 & 0 & 0 & 9 & 6 & 6 & 0 & 0 & 0 \\ 12 & 0 & 0 & 6 & 12 & 8 & 0 & 0 & 0 \\ 12 & 0 & 0 & 6 & 8 & 12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix}$$

$$\mathbf{A}^{-1}_{3\times3\times2} = \begin{pmatrix} 1/3 & 0 & 0 & -1/9 & 1/6 & -1/6 & 0 & 0 & 0 \\ 0 & 1/6 & 0 & 0 & 0 & 0 & 0 & -1/6 & 0 \\ 0 & 0 & 1/6 & 0 & 0 & 0 & 0 & 0 & -1/6 \\ -1/9 & 0 & 0 & 2/9 & 0 & 0 & 0 & 0 & 0 \\ -1/6 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ -1/6 & 0 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/8 & 0 & 0 \\ 0 & -1/6 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & -1/6 & 0 & 0 & 0 & 0 & 0 & 1/3 \end{pmatrix}$$

$$\mathbf{A}_{5\times5\times2} = \begin{pmatrix} 50 & 0 & 0 & 25 & 100 & 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 50 \\ 25 & 0 & 0 & 25 & 50 & 50 & 0 & 0 & 0 \\ 100 & 0 & 0 & 50 & 340 & 200 & 0 & 0 & 0 \\ 100 & 0 & 0 & 50 & 200 & 340 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 200 & 0 & 0 \\ 0 & 50 & 0 & 0 & 0 & 0 & 0 & 50 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 & 0 & 0 & 50 \end{pmatrix}$$

$$A^{-1}_{5\times5\times2} = \begin{pmatrix} 17/175 & 0 & 0 & -1/25 & 1/70 & -1/70 & 0 & 0 & 0 \\ 0 & 1/50 & 0 & 0 & 0 & 0 & 0 & -1/50 & 0 \\ 0 & 0 & 1/50 & 0 & 0 & 0 & 0 & 0 & -1/50 \\ -1/25 & 0 & 0 & 2/50 & 0 & 0 & 0 & 0 & 0 \\ -1/70 & 0 & 0 & 0 & 1/140 & 0 & 0 & 0 & 0 \\ -1/70 & 0 & 0 & 0 & 0 & 1/140 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/200 & 0 & 0 \\ 0 & -1/50 & 0 & 0 & 0 & 0 & 0 & 1/25 & 0 \\ 0 & 0 & -1/50 & 0 & 0 & 0 & 0 & 0 & 1/25 \end{pmatrix}$$

Appendix D

Temporal Derivative of Shading Models

D.1 Temporal Derivative of General Lambertian Model

This section derives equation 3.2, the temporal derivative of the general Lambertian shading model described by equation 3.1. Let us consider a constant light source of intensity L_0 and position $\mathbf{l}$. Denote by $\mathbf{L}$ the light source as seen by the patch at point $\mathbf{R}$

$$\hat{\mathbf{L}} = \frac{\mathbf{L}}{\|\mathbf{L}\|} = \frac{\mathbf{l}-\mathbf{R}}{\|\mathbf{l}-\mathbf{R}\|}$$

were $\mathbf{L} = \mathbf{l} - \mathbf{R}$ is the illumination vector from the surface to the source (see figure D.1)

The irradiance $E(\mathbf{r},t)$ is proportional to $(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}})$ (K constant of proportionality) i.e.

$$E(\mathbf{r},t) = K(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}}) \Longleftrightarrow E(\mathbf{r},t)\,\|\mathbf{l}-\mathbf{R}\| = K\big((\mathbf{l}-\mathbf{R}) \cdot \hat{\mathbf{n}}\big) \tag{D.1}$$

If we assume that $\mathbf{l}$ is fixed and differentiate equation D.1 with respect to time, we obtain

$$\begin{aligned}
& \dot{E}\,\|\mathbf{l}-\mathbf{R}\| - E\frac{\big((\mathbf{l}-\mathbf{R})\cdot\dot{\mathbf{R}}\big)}{\|\mathbf{l}-\mathbf{R}\|} = K\Big(-\dot{\mathbf{R}}\cdot\hat{\mathbf{n}} + \big((\mathbf{l}-\mathbf{R})\cdot\dot{\hat{\mathbf{n}}}\big)\Big) \\
\Longleftrightarrow\quad & \dot{E}\,\|\mathbf{L}\| = K\Big(\big(-\hat{\mathbf{n}}\cdot(\mathbf{t}+\boldsymbol{\omega}\times\mathbf{R})\big) + \big(\mathbf{L}\cdot(\boldsymbol{\omega}\times\hat{\mathbf{n}})\big)\Big) + E\big(\hat{\mathbf{L}}\cdot(\mathbf{t}+\boldsymbol{\omega}\times\mathbf{R})\big) \\
\Longleftrightarrow\quad & \dot{E} = \frac{K}{\|\mathbf{L}\|}\Big[\big(-\hat{\mathbf{n}}\cdot(\mathbf{t}+\boldsymbol{\omega}\times\mathbf{R})\big) + \big(\mathbf{L}\cdot(\boldsymbol{\omega}\times\hat{\mathbf{n}})\big) + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})\big(\hat{\mathbf{L}}\cdot(\mathbf{t}+\boldsymbol{\omega}\times\mathbf{R})\big)\Big]
\end{aligned}$$

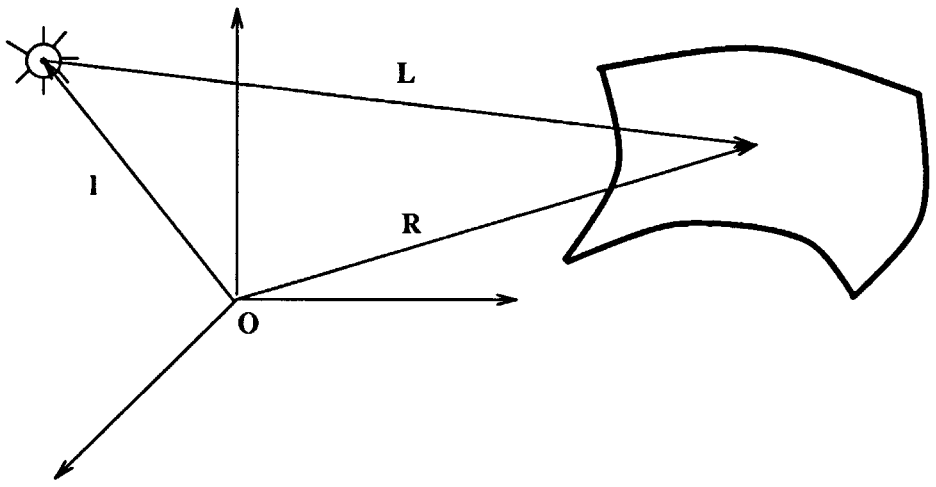

Figure D.1: Light source, camera and patch geometry

Replacing $\mathbf{L}$ by its value, we obtain

$$\begin{aligned}\dot{E} &= \frac{K}{\|\mathbf{L}\|}\Big([\mathbf{l},\boldsymbol{\omega},\hat{\mathbf{n}}] - (\mathbf{t}\cdot\hat{\mathbf{n}}) + ((\boldsymbol{\omega}\times\mathbf{R}+\mathbf{t})\cdot\hat{\mathbf{L}})(\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})\Big)\\ &= \frac{K}{\|\mathbf{L}\|}\Big([\mathbf{l},\boldsymbol{\omega},\hat{\mathbf{n}}] + (\hat{\mathbf{L}}\cdot\hat{\mathbf{n}})[\hat{\mathbf{L}},\boldsymbol{\omega},\mathbf{R}] - \Big((\mathbf{t}-(\mathbf{t}\cdot\hat{\mathbf{L}})\hat{\mathbf{L}})\cdot\hat{\mathbf{n}}\Big)\Big)\end{aligned}$$

D.2 First–Order Approximation of Lambertian Model

This section derives equation 3.3, the first–order approximation of the temporal shading variations of the general Lambertian model described by equation 3.2.

The square of the norm of the light source–patch vector $\mathbf{L} = \mathbf{l} - \mathbf{R}$ is given by

$$\begin{aligned}\|\mathbf{L}\|^2 &= \|\mathbf{l}\|^2 - 2(\hat{\mathbf{l}}\cdot\hat{\mathbf{R}})\,\|\mathbf{R}\|\,\|\mathbf{l}\| + \|\mathbf{R}\|^2\\ &= \|\mathbf{l}\|^2\left(1 - 2(\hat{\mathbf{l}}\cdot\hat{\mathbf{R}})\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} + \left(\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right)^2\right).\end{aligned}$$

The first–order Taylor series of the reciprocal of the norm of $\mathbf{L}$ is directly computed from the previous equation and is expressed by

$$\|\mathbf{L}\|^{-1} = \|\mathbf{l}\|^{-1}\left(1 + (\hat{\mathbf{l}}\cdot\hat{\mathbf{R}})\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right) + \mathcal{O}\left(\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right). \tag{D.2}$$

The first–order expansion of the unit vector $\hat{\mathbf{L}}$ is inferred from equation D.2 and can be written, after regrouping the zeroth– and first–order terms, as

$$\hat{\mathbf{L}} = \hat{\mathbf{l}} - (\hat{\mathbf{R}} - (\hat{\mathbf{l}}\cdot\hat{\mathbf{R}}))\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} + \mathcal{O}\left(\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right) \tag{D.3}$$

The $\hat{\mathbf{L}}$–orthogonal translation vector $\mathbf{t}_{\perp}^{\hat{\mathbf{L}}} = \mathbf{t} - (\mathbf{t} \cdot \hat{\mathbf{L}})\hat{\mathbf{L}}$ can similarly be approximated by a first–order expression. If we denote by $\mathbf{t}_{\perp}^{\hat{\mathbf{l}}} = \mathbf{t} - (\mathbf{t} \cdot \hat{\mathbf{l}})\hat{\mathbf{l}}$ the $\hat{\mathbf{l}}$–orthogonal translation vector and use the equation D.3, the first–order Taylor series of $\mathbf{t}_{\perp}^{\hat{\mathbf{L}}}$ is given by

$$\mathbf{t}_{\perp}^{\hat{\mathbf{L}}} = \mathbf{t}_{\perp}^{\hat{\mathbf{l}}} + \Big((\mathbf{t} \cdot \hat{\mathbf{l}})\mathbf{R} + (\mathbf{t} \cdot \hat{\mathbf{R}})\hat{\mathbf{l}} - 2(\mathbf{t} \cdot \hat{\mathbf{l}})(\hat{\mathbf{l}} \cdot \hat{\mathbf{R}})\hat{\mathbf{l}}\Big)\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} + \mathcal{O}\left(\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right) \tag{D.4}$$

The first–order approximation of the term $(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}})[\hat{\mathbf{L}}, \boldsymbol{\omega}, \mathbf{R}]$ is determined similarly by using equation D.3, multiplying out the terms and gathering the zeroth– and first–order terms—

$$(\hat{\mathbf{L}} \cdot \hat{\mathbf{n}})[\hat{\mathbf{L}}, \boldsymbol{\omega}, \mathbf{R}] = [\hat{\mathbf{l}}, \boldsymbol{\omega}, \mathbf{R}]\Big(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}} - \hat{\mathbf{R}} \cdot \hat{\mathbf{n}} - 2(\hat{\mathbf{l}} \cdot \mathbf{r})(\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})\Big)\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} + \mathcal{O}\left(\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|}\right) \tag{D.5}$$

The first–order Taylor series (equation 3.3) is finally determined by plugging back the equations D.2, D.3, D.4 and D.5 into the original equation 3.2, multiplying out all the terms and collating the zeroth– and first order terms to yield

$$\dot{E} = \rho_{\lambda}(\alpha, \beta) L_0 \left([\hat{\mathbf{l}}, \boldsymbol{\omega}, \hat{\mathbf{n}}] + (\hat{\mathbf{l}} \cdot \hat{\mathbf{n}})[\hat{\mathbf{l}}, \boldsymbol{\omega}, \hat{\mathbf{R}}]\frac{\|\mathbf{R}\|}{\|\mathbf{l}\|} - \frac{(\mathbf{t}_{\perp}^{\hat{\mathbf{l}}} \cdot \hat{\mathbf{n}})}{\|\mathbf{l}\|} \right).$$

Appendix E

Quadratic Case Shading Equation

This appendix presents the derivations of the expressions of the generic normal $\mathbf{n}$ and unit normal $\hat{\mathbf{n}}$ in terms of the unit normal $\hat{\mathbf{n}}_0$ and curvatures at $\mathbf{Z}_0$, the point of expansion of the Taylor series expressing the reciprocal of the depth in terms of the image coordinates.

A quadratic patch is expressed by

$$Z(X,Y) = Z_0 + Z_X X + Z_Y Y + \frac{1}{2} Z_{XX} X^2 + Z_{XY} XY + \frac{1}{2} Z_{YY} Y^2$$

or

$$Z_0 = \mathbf{R} \cdot \mathbf{n}_0 + \mathbf{Q} \cdot \mathbf{d}$$

where $\mathbf{n}_0 = (-Z_X \;\; -Z_Y \;\; 1)^T$ represents the normal at the point $\mathbf{Z}_0$, $\mathbf{Q} = (\frac{1}{2}X^2 \;\; XY \;\; \frac{1}{2}Y^2)^T$ is a quadratic world coordinates vector and $\mathbf{d} = (-Z_{XX} \;\; -Z_{YY} \;\; -Z_{XY})^T$ is the "curvature" vector at the point $\mathbf{Z}_0$. The reciprocal of the depth can immediately be computed in terms of the image coordinates by a second-order Taylor series of $1/Z$,

$$\frac{1}{Z} = \frac{1}{Z_O}(\mathbf{r} \cdot \hat{\mathbf{n}}_0) + \mathbf{q} \cdot \mathbf{d} + \mathcal{O}(\|\mathbf{r}\|^2) \tag{E.1}$$

where $\mathbf{q} = (\frac{1}{2}x^2 \;\; xy \;\; \frac{1}{2}y^2)^T$ is a quadratic image coordinate vector.

The generic normal $\mathbf{n}$ is defined by

$$\mathbf{n} = \begin{pmatrix} -\dfrac{\partial Z(X,Y)}{\partial X} \\ -\dfrac{\partial Z(X,Y)}{\partial Y} \\ 1 \end{pmatrix} = \begin{pmatrix} -Z_X \\ -Z_Y \\ 1 \end{pmatrix} + \begin{pmatrix} -Z_{XX}X - Z_{XY}Y \\ -Z_{XY}X - Z_{YY} \\ 0 \end{pmatrix}$$

and can be rewritten, in vectorial form,

$$\mathbf{n} = \mathbf{n_0} + \mathbf{HR} = \mathbf{n_0} + Z\mathbf{Hr} \tag{E.2}$$

where the matrix $\mathbf{H}$ is defined by

$$\mathbf{H} = \begin{pmatrix} -Z_{XX} & -Z_{XY} & 0 \\ -Z_{XY} & -Z_{YY} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The depth Z can be computed in terms of the image coordinates by a second–order Taylor series; from equation E.1 we get

$$\begin{aligned} Z &= Z_0\big(\mathbf{r}\cdot\mathbf{n_0} + Z_0(\mathbf{q}\cdot\mathbf{d})\big)^{-1} \\ &= Z_0\big(1 + \tilde{\mathbf{r}}\cdot\mathbf{n_0} + Z_0(\mathbf{q}\cdot\mathbf{d})\big)^{-1} \\ &= Z_0\big(1 - \tilde{\mathbf{r}}\cdot\mathbf{n_0}\big) - Z_0(\mathbf{q}\cdot\mathbf{d}) + (\tilde{\mathbf{r}}\cdot\mathbf{n_0})^2 + \mathcal{O}(\|\mathbf{r}\|^2). \end{aligned} \tag{E.3}$$

Plugging the value of the depth Z, defined by equation E.3, into equation E.2, and collecting the terms up to the second order yields

$$\mathbf{n} = \mathbf{n_0} + Z_0\big(1 - \tilde{\mathbf{r}}\cdot\mathbf{n_0}\big)\mathbf{Hr} + \mathcal{O}(\|\mathbf{r}\|^2). \tag{E.4}$$

In order to compute the unit normal, we need to determine the value of $\|\mathbf{n}\|^{-1}$. The square of the norm of $\mathbf{n}$ is given by

$$\begin{aligned} \|\mathbf{n}\|^2 &= \|\mathbf{n_0}\|^2 + 2Z_0\big(1 - \tilde{\mathbf{r}}\cdot\mathbf{n_0}\big)(\mathbf{n_0}\cdot\mathbf{Hr}) + Z_0^2\mathbf{r}^T\mathbf{H}^2\mathbf{r} + \mathcal{O}(\|\mathbf{r}\|^2) \\ &= \|\mathbf{n_0}\|^2\left(1 + 2\frac{Z_0}{\|\mathbf{n_0}\|}\big(1 - \tilde{\mathbf{r}}\cdot\mathbf{n_0}\big)(\hat{\mathbf{n}}_0\cdot\mathbf{Hr}) + \frac{Z_0^2}{\|\hat{\mathbf{n}}_0\|^2}\mathbf{r}^T\mathbf{H}^2\mathbf{r}\right) + \mathcal{O}(\|\mathbf{r}\|^2). \end{aligned} \tag{E.5}$$

From equation E.5 we can compute a second–order Taylor series of $\|\hat{\mathbf{n}}\|^{-1}$:

$$\begin{aligned} \frac{1}{\|\mathbf{n}\|} &= \frac{1}{\|\mathbf{n_0}\|}\left(1 - \frac{Z_0}{\|\hat{\mathbf{n}}_0\|}\big(1 - \tilde{\mathbf{r}}\cdot\hat{\mathbf{n}}_0\big)(\hat{\mathbf{n}}_0\cdot\mathbf{Hr}) - \frac{1}{2}\frac{Z_0^2}{\|\hat{\mathbf{n}}_0\|^2}\mathbf{r}^T\mathbf{H}^2\mathbf{r} + \right. \\ &\qquad \left. \frac{3}{8}\left(\frac{4Z_0^2}{\|\hat{\mathbf{n}}_0\|^2}\big(1 - \tilde{\mathbf{r}}\cdot\hat{\mathbf{n}}_0\big)^2\mathbf{r}^T\mathbf{H}\hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T\mathbf{Hr}\right)\right) + \mathcal{O}(\|\mathbf{r}\|^2) \\ &= \frac{1}{\|\mathbf{n_0}\|}\left(1 - \frac{Z_0}{\|\hat{\mathbf{n}}_0\|}\big(1 - \tilde{\mathbf{r}}\cdot\hat{\mathbf{n}}_0\big)(\hat{\mathbf{n}}_0\cdot\mathbf{Hr}) - \frac{1}{2}\frac{Z_0^2}{\|\hat{\mathbf{n}}_0\|^2}\mathbf{r}^T\big(\mathbf{H}^2\mathbf{r} - 3\mathbf{H}\hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T\mathbf{H}\big)\mathbf{r}\right) + \mathcal{O}(\|\mathbf{r}\|^2) \end{aligned} \tag{E.6}$$

The second–order Taylor series of $\hat{\mathbf{n}}$ is obtained by multiplying out the Taylor series of $\mathbf{n}$ (equation E.4) and $\|\mathbf{n}\|^{-1}$ (equation E.6) and collecting all the terms up to the second order to finally obtain

$$\hat{\mathbf{n}} = \hat{\mathbf{n}}_0 + \frac{Z_0}{\|\mathbf{n}_0\|}(1 - \tilde{\mathbf{r}} \cdot \mathbf{n}_0)(\mathbf{I}_3 - \hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T)\mathbf{H}\mathbf{r} - \frac{1}{2}\frac{Z_0^2}{\|\mathbf{n}_0\|^2}\left(\mathbf{r}^T\tilde{\mathbf{H}}\mathbf{r}\hat{\mathbf{n}}_0 + 2(\hat{\mathbf{n}}_0 \cdot \mathbf{H}\mathbf{r})\mathbf{H}\mathbf{r}\right) + \mathcal{O}(\|\mathbf{r}\|^3)$$

where $\tilde{\mathbf{H}} = \mathbf{H}^3 - 3\mathbf{H}\hat{\mathbf{n}}_0\hat{\mathbf{n}}_0^T\mathbf{H}$.

Appendix F

Implementation Issues

In the following expressions, we will use the summation convention of tensor calculus, i.e. there is an implicit summation over any index that appears twice in an expression.

$$\begin{aligned}
\{\mathbf{M}_1\}_{ij} &= \underbrace{\iint_\sigma v_i v_j d\mathbf{r}}_{\mathcal{E}_1(9\mapsto 6)} + \underbrace{\left(\iint_\sigma v_i d\mathbf{r}\right)}_{\mathcal{E}_2(3\mapsto 3)} \{\hat{\mathbf{n}}\times\mathbf{k}\}_j + \underbrace{\left(\iint_\sigma v_j d\mathbf{r}\right)}_{\mathcal{E}_2(3\mapsto 3)} \{\hat{\mathbf{n}}\times\mathbf{k}\}_i \\
&\qquad + \underbrace{\left(\iint_\sigma d\mathbf{r}\right)}_{\mathcal{E}_3(1\mapsto 1)} \{\hat{\mathbf{n}}\times\mathbf{k}\}_i \{\hat{\mathbf{n}}\times\mathbf{k}\}_j \\
\{\mathbf{M}_2\}_{ij} &= \underbrace{\left(\iint_\sigma v_i s_j r_k d\mathbf{r}\right)}_{\mathcal{E}_4(27\mapsto 27)} n_k + \{\hat{\mathbf{n}}\times\mathbf{k}\}_i \underbrace{\left(\iint_\sigma s_j r_k d\mathbf{r}\right)}_{\mathcal{E}_5(9\mapsto 9)} n_k \\
\{\mathbf{M}_4\}_{ij} &= \underbrace{\left(\iint_\sigma s_i s_j r_k r_l d\mathbf{r}\right)}_{\mathcal{E}_6(81\mapsto 36)} n_k n_l \\
\{\mathbf{e}_1\}_i &= \underbrace{\left(\iint_\sigma E_t v_i d\mathbf{r}\right)}_{\mathcal{E}_7(3\mapsto 3)} + \{\hat{\mathbf{n}}\times\mathbf{k}\}_i \underbrace{\left(\iint_\sigma E_t d\mathbf{r}\right)}_{\mathcal{E}_8(1\mapsto 1)} \\
\{\mathbf{e}_2\}_i &= \underbrace{\left(\iint_\sigma E_t s_i r_k d\mathbf{r}\right)}_{\mathcal{E}_9(9\mapsto 9)} n_k
\end{aligned}$$

In the previous expressions, $\mathcal{E}_i(j \mapsto k)$ represents the i^{th} precomputed term has j elements from which only k are distinct (symmetry).. All the $\mathcal{E}_i$, i.e. 95 numbers, depend only on $\mathbf{r}$, $E_\mathbf{r}$ and E_t, and represent all the image information required to solve the problem.

Bibliography

Adiv, G. 1985. Determining three–dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 7(4):384–401.

Aloimonos, J., and A. Bandopadhay. 1987. Active vision. In *Proc. Int. Conf. Computer Vision*, 35–54, London, England, 8–11 June.

Aloimonos, Y. 1987. Combining sources of information in vision: I. computing shape from shading and motion. *Perception* 738–741.

Ballard, D., and O. Kimball. 1983. Rigid body motion from depth and optical flow. *Comput. Vision Graph. Image Process.* 22:95–115.

Bier, E., and K. Sloan. 1986. Two–part texture mappings. *IEEE Comput. Graph. Applic.* 40–53.

Bolles, R., and H. Baker. 1985. Epipolar–plane image analysis: a technique for analyzing motion sequences. In *Proc. Workshop Comput. Vision: Representation and Control*, 168–178, Bellaire, MI, 13–16 October.

Bolles, R., H. Baker, and D. Marimont. 1987. Epipolar–plane image analysis: an approach to determining structure from motion. *Int. J. Comput. Vision* 1(1):7–55.

Broida, T., and R. Chellappa. 1985. Estimation of object motion parameters from noisy images. In *Proc. Conf. Comput. Vision Pattern Recognit.*, 82–88, San Francisco, CA, June.

Broida, T., and R. Chellappa. 1986. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-8(1):90–99.

Brooks, M., and B. Horn. 1985. Shape and source from shading. AI Memo 820, Artif. Intell. Lab., MIT, January.

Brown, C., D. Ballard, and O. Kimball. 1982. Constraint interaction in shape from shading algorithms. In *Proc. Image Understanding Workshop*, 79–89.

Bruss, A., and B. Horn. 1983. Passive navigation. *Comput. Vision Graph. Image Process.* 21:3–20.

Bui-Tong, P. 1975. Illumination for computer–generated images. *Comm. ACM* 18(6):311–317.

Cafforio, C., and F. Rocca. 1983. The differential method for image motion estimation. In T. Huang (Ed.), *Image Sequence Processing and Dynamic Scene Analysis*, 104–124. Springer–Verlag.

Chow, S., J. Mallet-Paret, and J. Yorke. 1978. Finding zeros of maps: Homotopy methods that are constructive with probability one. *Math. Comput.* 32:887–899.

Davidĕko, D. 1953. On a new method of numerical solution of systems of non–linear equations. *Math. Reviews* 14:906.

Deist, F., and L. Sefor. 1967. Solution of systems of nonlinear equations by parameters variation. *Comput. J.* 10(1):78–82.

Fletcher, R., and M. Powell. 1963. A rapidly convergent descent method for minimization. *Comput. J.* 6:163–168.

Grzywacz, N., and E. Hildreth. 1987. Incremental rigidity scheme for recovering structure from motion: Position–based versus velocity–based formulations. *J. Opt. Soc. Am.* 4(845):503–518.

Horn, B. 1986. *Robot Vision.* MIT Press.

Horn, B. 1987. Relative orientation. AI Memo 994, Artif. Intell. Lab., MIT, September.

Horn, B., and B. Schunck. 1981. Determining optical flow. *Artif. Intell.* 17:185–203.

Horn, B., and R. Sjoberg. 1979. Calculating the reflectance map. *Appl. Opt.* 18(11).

Horn, B., and E. Weldon. 1988. Direct methods for recovering motion. *Int. J. Comput. Vision* 2(2):51–76.

Huang, T., J. Weng, and N. Ahuja. 1986. 3–D motion from image sequences: Modeling, understanding and prediction. In *Proc. Workshop Motion: Representation and Analysis*, 125–130, Charlestown, SC, 7–9 May.

Keys, R. 1981. Cubic convolution for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* ASSP-29(6).

Krause, E. 1987. *Motion Estimation for Frame–Rate Conversion.* PhD thesis, Elect. Eng. and Comput. Sci. Dept., MIT, June.

Lee, C., and A. Rosenfeld. 1985. Improved methods of estimating shape from shading using the light source coordinate systems. *Artif. Intell.* 26:125–143.

Limb, J., and J. Murphy. 1975. Estimating the velocity of moving images from television signals. *Comput. Graph. Image Process.* 4(2):311–327.

Longuet-Higgins, H., and K. Prazdny. 1980. The interpretation of a moving retinal image. *Proc. R. Soc. London, Ser. B* 208:385–397.

Martinez, D., and J. Lim. 1986. Implicit motion compensated noise reduction of motion video scenes. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Process.*, 375–378, Tampa, FL, 26 March.

Moffit, F., and E. Mikhail. 1980. *Photogrammetry.* New York, NY: Harper and Row.

More, J. 1977. The Levenberg–Marquardt algorithms: Implementation and theory. In G. Watson (Ed.), *lectures Notes in Mathematics 630.* Springer–Verlag.

Nagel, H. 1989. On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-11(1):13–30.

Negahdaripour, S. 1986. *Direct Methods for Structure from Motion.* PhD thesis, MIT.

Negahdaripour, S. 1987. Ambiguities of a motion field. In *Proc. Int. Conf. Computer Vision*, 607–611, London, England, 8–11 June.

Negahdaripour, S., and B. Horn. 1987. Direct passive navigation. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-9(1):168–176.

Netravali, A., and J. Salz. 1985. Algorithms for estimation of 3–D motion. *AT&T Tech. J.* 64(2):335–346.

Nicodemus, F., J. Richmond, J. Hsia, I. Ginsberg, and T. Limperis. 1977. Geometrical considerations and nomenclature for reflectance. NBS Monograph 160, Nat. Bureau of Standards, Washington, DC, October.

Ortega, J., and W. Rheinboldt. 1970. *Iterative Solution of Nonlinear Equations in Several Variables.* New York, NY: Academic Press.

Pentland, A. 1984. Local shading analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-6(2):170–187.

Powell, M. 1970. A hybrid method for nonlinear equations. In P. Rabinowitz (Ed.), *Numerical Methods for Nonlinear Algebraic Equations*, 87–114. London–Paris–New York: Gordon and Breach Science.

Roach, J., and J. Aggarwal. 1980. Determining the movement of objects from a sequence of images. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-2(6):554–562.

Schalkoff, R., and E. McVey. 1982. A model and tracking algorithm for a class of video targets. *IEEE Trans. Pattern Anal. Mach. Intell.* PAMI-4(1):2–10.

Schunck, B. 1985. Image flow: Fundamentals and future research. In *Proc. Conf. Comput. Vision Pattern Recognit.*, 560–571, San Francisco, CA, June.

Shariat, H., and K. Price. 1986. How to use more than two frames to estimate motion. In *Proc. Workshop Motion: Representation and Analysis*, 89–94, Charlestown, SC, 7–9 May.

Torrance, K., and E. Sparrow. 1967. Theory for off–specular reflection from roughened surfaces. *J. Opt. Soc. Am.* 57(9).

Tsai, R., and T. Huang. 1981. Estimating three–dimensional motion parameters of a rigid planar patch. *IEEE Trans. Acoust. Speech Signal Process.* ASSP-29(6):1147–1152.

Tsai, R., and T. Huang. 1984. Estimating three–dimensional motion parameters of a rigid planar patch, III: Finite point correspondences and the three–view problem. *IEEE Trans. Acoust. Speech Signal Process.* ASSP-32(2):213–220.

Ullman, S. 1983. Maximizing rigidity: The incremental recovery of 3–D structure from rigid and rubbery motion. AI Memo 721, Artif. Intell. Lab., MIT, June.

Watson, L., and D. Fenner. 1980. Algorithm 555: Chow–Yorke algorithm for fixed points or zeros of c^2 maps. *ACM Trans. on Math. Soft.* 6(2):252–259.

Waxman, A., and S. Ullman. 1983. Surface structure and 3–D motion from image flow: a kinematic analysis. Tech. Rept. CS-TR-1332, Univ. of Maryland, October.

Waxman, A., and K. Wohn. 1984. Contour evolution, neighborhood deformation and global image flow: Planar surfaces in motion. Tech. Rept. CS–TR–1394, Univ. of Maryland, April.

Webb, J., and J. Aggarwal. 1983. Shape and correspondence. *Comput. Vision Graph. Image Process.* 21:145–160.

Wohn, K., and A. Waxman. 1985. Contour evolution, neighborhood deformation and local image flow: Curved surfaces in motion. Tech. Rept. CS–TR–1531, Univ. of Maryland, July.

CS-TR Scanning Project
Document Control Form

Date : 6/29/95

Report # AI-TR-1162

Each of the following should be identified by a checkmark:
Originating Department:

☒ Artificial Intellegence Laboratory (AI)
☐ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 190 (197-IMAGES)
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☒ Laser Print
☐ InkJet Printer ☐ Unknown ☐ Other:_____

Check each if included with document:

☒ DOD Form (2) ☐ Funding Agent Form ☒ Cover Page
☐ Spine ☐ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages (by page number): 2,4,6,8,12

Photographs/Tonal Material (by page number): 94,95,97,98,113,121,123,130,135,137,138,143,146

Other (note description/page number):

Description : Page Number:

(A) IMAGE MAP: (1) UN#'ED TITLE PAGE
(2-190) PAGES #'ED 2-190
(191-194) SCANCONTROL, COVER, DOD(2)
(195-197) TRGT'S (3)

(B) PAGES WITH PHOTO'S ARE NOT ORG,

Scanning Agent Signoff:

Date Received: 6/29/95 Date Scanned: 7/12/95 Date Returned: 7/13/95

Scanning Agent Signature: Michael W. Cook

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AI-TR 1162	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Three-Dimensional Motion Estimation Using Shading Information in Multiple Frames		5. TYPE OF REPORT & PERIOD COVERED technical report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jean-Pierre Schott		8. CONTRACT OR GRANT NUMBER(s) N00014-85-K-0124
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, MA 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, VA 22209		12. REPORT DATE August 1989 13. NUMBER OF PAGES 190
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Office of Naval Research Information Systems Arlington, VA 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

motion recovery	motion vision
3-D structure	3-D vision
shape from shading	multiple frames

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Reverse

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 cont'd

Traditionally motion and shading have been treated as two disjoint problems. On the one hand, researchers studying motion or structure from motion often assume uniform lighting conditions over the whole surface and good contrast at high spatial frequencies to minimize the effects of variations of the image irradiance of the patch as the surface moves. On the other hand, researchers primarily concerned with the shape from shading problem only consider static brightness data in order to recover the shape without considering the change of brightness induced by motion.

A new formulation for recovering the structure and motion parameters of a moving patch is presented. It is based on using the spatiotemporal derivatives of irradiance that are computed from a time–varying irradiance sequence and combined into a differential constraint equation. The new approach determines the rigid body motion and the structure of the patch directly from the irradiance sequence using *both* motion and shading information.

A new constraint equation, the full irradiance constraint equation (FICE), is derived. It links the spatiotemporal gradients of irradiance to the motion and structure parameters *and* the temporal variations of the surface shading. This equation separates the contribution to the irradiance spatiotemporal gradients of the gradients due to texture from those due to shading and allows the FICE to be used for textured and textureless surface. The new approach combining motion and shading information, leads directly to two different contributions: it can compensate for the effects of shading variations in recovering the shape and motion; and it can exploit the shading/illumination effects to recover motion and shape when they cannot be recovered without it. The FICE formulation is extended to multiple frames, and several methods are presented for efficiently computing the structure and motion parameters directly from a sequence of data.

Overall, the examples demonstrate the superiority of the FICE algorithms to the classical CE algorithms in two distinct areas: the accuracy of the results is higher for textured surfaces and a solution can be determined in the case of textureless surfaces.

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**