Technical Report 1296

Temporal Surface Reconstruction

Joachim Heel

MIT Artificial Intelligence Laboratory

This blank page was inserted to preserve pagination.

Temporal Surface Reconstruction

by

Joachim Heel

Dipl. Ing. Universität Karslruhe, Germany
(1987)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

at the
Massachusetts Institute Of Technology
May 3, 1991

Signature of Author_____

Joachim Heel
Department of Electrical Engineering and Computer Science
May 3, 1991

Certified by_____

Professor Berthold K. P. Horn
Thesis Supervisor

Accepted by_____

Professor Arthur C. Smith
Chairman, Departmental Graduate Committee

Temporal Surface Reconstruction

by

Joachim Heel

Submitted to the Department of Electrical Engineering and Computer Science on May 3 1991, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

This thesis investigates the problem of estimating the three-dimensional structure of a scene from a sequence of images. Structure information can be recovered from images through a number of *visual mechanisms* such as shading, motion and stereo. Image information is commonly available in a time-continuous fashion and this work proposes a method for estimating structure information in a *temporally continuous* manner for a variety of visual mechanisms.

Structural information about a scene is represented in a dense *depth map* in which the distance to the scene is stored for each pixel location in the image. In addition, uncertainty about the structure values is represented explicitly by the estimate covariance. This representation is maintained over time by a stochastic recursive estimator, the *Kalman filter*. The estimator consists of two stages which are repeated for each new arriving image. The update stage improves the current depth estimate by incorporating the latest image measurement. It depends on the particular visual mechanism being employed and amounts to an iterative relaxation algorithm similar to conventional single-frame algorithms. The prediction stage transforms the current depth estimate into the next time-step to account for changes in the depth values that may occur if the camera moves relative to the (rigid) scene during the acquisition of the sequence. This step requires a three-dimensional transformation (translation and rotation) of each depth map entry followed by a resampling operation to maintain the regular map representation.

The temporal reconstruction algorithm is described in detail for the recovery of structure from motion with and without optical flow and for structure from shading. Extensive experimental evaluation shows that the temporal algorithm not only improves the quality of estimates significantly over time but also requires orders of magnitude less time per image than previous "instantaneous" techniques.

Thesis Committee: [redacted], Chairman

Acknowledgments

I spent four years on the research described in this thesis and it would not be here in front of you if it hadn't been for the help and support of so many wonderful people. Mentioning them here is the least I can do to thank them.

My parents were not only the first to ever teach me, they also believed that I could grow to learn beyond their teachings and unconditionally supported me in doing so. They kindled my desire for knowledge and taught me to cherish what I had learned. I was fortunate to have exceptional teachers to satisfy my hunger for knowledge in school and in college and I am deeply indebted to them for sharing their wisdom and guiding me along the way: Gottfried Stumpf, Alwin Sennefelder, Edgar Nikolaus, Otto Föllinger and Peter Deussen.

Coming to study at the AI Lab was certainly the most outstanding educational experience in my life. The informal atmosphere of constant communication with disciplines from biology to philosophy and nationalities from Australia to Brazil is truly exceptional. Where else can you spend hours discussing proofs of convex hull algorithms on a Sunday at midnight, race robot boats around a bath tub and write programs to squirt your professor with a water pistol? All the people that make up this amazing place contributed to this thesis.

Among my friends at the lab, many have directly influenced the contents of this thesis through their comments, ideas and numerous conversations. Dave Michael, Davi Geiger, Ron Chaney, Satyajit Rao, Ed Gamble, David Beymer, Barbara Moore and Jean-Pierre Schott are just a few of those that made their mark on my work. Ellen Hildreth, Hans-Hellmut Nagel, John Aloimonos, Shahriar Negahdaripour, Richard Szeliski, Larry Matthies, Tomaso Poggio and the INRIA vision group guided me along the way to this thesis and shared their experience with me. I'd like to especially thank my friends Brian Subirana and Sundar Narasimhan for supporting me through all the ups and downs of these four years.

I was fortunate enough to have Berthold Horn as my thesis advisor. He allowed me to benefit from his knowledge and experience while not constraining me in my own creative efforts. Learning from him was truly an exceptional experience. Olivier Faugeras and Alan Willsky not only read this thesis but shaped it into what is here before you. I thank all of them for devoting their time to me.

Financial support for this work was provided through the AI Laboratory in part by the Advanced Research Projects Agency under contract number DACA 76-85-K-0685 and under Office of Naval Research contract N00014-85-K-0124, by the Massachusetts Institute of Technology, the Studienstiftung des Deutschen Volkes, the Deutscher Akademischer Austauschdienst and Gertrud and Helmut Heel.

To Johanna

Ich saz uf eime steine,
und dahte bein mit beine:
dar uf satzt ich den ellenbogen:
und hete in mine hant gesmogen
daz kinne und ein min wange.
do dahte ich mir vil ange,
wie man zer welte solte leben:
deheinen rat kond ich gegeben,
wie man driu dinc erwurbe,
der keines niht verdurbe.
diu zwei sint ere und varnde guot,
daz dicke ein ander schaden tuot:
daz dritte ist gotes hulde,
der zweier ubergulde.
die wolte ich gerne in einen schrin.
ja leider desn mac niht gesin,
daz guot und weltlich ere
und gotes hulde mere
zesamene in ein herze komen.
...

Walther von der Vogelweide

Contents

List of Figures

Introduction

Chapter 1

1.1 Problem Statement

How can we perceive three-dimensional structure? Brightness images of three-dimensional scenes contain a wealth of information which humans can exploit through a variety of mechanisms to extract information about the structure of objects. Moreover, this cognitive process has a temporal dimension: humans can maintain and improve an "idea" of a three-dimensional structure as they acquire more images of a scene from varying viewpoints.

The objective of this thesis is to formalize the problem of temporal surface reconstruction outlined above and to investigate computational visual algorithms for its solution. Let us begin by stating the problem more precisely:

> *Temporal Surface Reconstruction:*
>
> We are given a sequence of intensity images of a three-dimensional scene. The objective is to estimate the three-dimensional structure of the observed scene.

The solution to the above problem will consist of answers to the following questions:

- *Representation:* What are the representations of three-dimensional structure suitable for surface reconstruction, considering in particular the ability to maintain the representation over time?

- *Visual Mechanisms:* What are the visual mechanisms that can be exploited to recover information about three-dimensional surfaces from brightness images?

- *Algorithms:* What are the computational algorithms which are best suited to exploit the above visual mechanisms and the above structure representations to obtain estimates of the three-dimensional surfaces which are closest to the true surfaces at the lowest computational cost?

This thesis will investigate one solution to the above problems and contrast it with other possible alternatives.

1.2 An Illustrative Example

To gain some insight into the difficulties involved, let us consider a straightforward solution to the above problem. It is well-known that stereoscopic vision is a primary source of three-dimensional perceptive capability in humans. Marr and Poggio [61] argue that humans match brightness "edges" in the left and right images. The disparity between matching edge locations is inversely proportional to the distance of the corresponding point in the world. In this case, the *representation* of the three-dimensional structure would be the distance of points on the surface that project to edge locations in the image. The *visual mechanism* is the inverse relationship between the distance of surface points and the disparity of matching edge locations in the image. The *algorithm* (such as the one by Grimson [31]) consists of extracting edge locations, matching them in the left and right images and calculating the depth from the resulting disparity.

Neither this description nor the original papers cited above address the temporal aspect of the problem i.e. how structure information can be recovered using stereo if an entire sequence of stereo pairs is available. Of course, it would be straightforward to repeat the instantaneous algorithm for every pair of images in the sequence as it becomes available. This appears counterintuitive since the calculation for a given pair of frames completely disregards estimates from previous frames and can therefore not hope to provide the continuous estimation and estimate improvement which humans exhibit. We can formulate the difficulties and disadvantages affiliated with such an "instantaneous" surface reconstruction procedure more precisely:

1. Instantaneous structure estimates are sensitive to measurement errors and noise. Combining estimates from a number frames introduces redundancies that can be exploited to reduce the effect of errors. However, the instantaneous approach cannot combine measurements and therefore has no temporal error-reduction effect.

2. In order to combine estimates from different frames, the estimates must be compatible. However, the relative position of camera and scene may change during the acquisition of frames and thereby cause instantaneous structure estimates taken at different positions to be incompatible. Transformations of structure estimates to account for camera displacement are necessary to overcome isolated processing of images.

3. Once temporal structure estimates are compatible, we need a procedure to "combine" them. Measurements taken at different times may vary in terms of error and noise. In particular in the case where camera and scene are in relative motion, the uncertainty in the structure estimate will vary spatially and temporally. The instantaneous approach has no way of representing the uncertainty and no way of using such a representation to improve estimates of high uncertainty by combining them with others of lower uncertainty.

4. Typically, instantaneous surface reconstruction procedures such as the Marr-Poggio/Grimson stereo algorithm mentioned above are computationally quite expensive. In the instantaneous scheme these expensive procedures must be repeated for each frame and since processing is done in isolation no computational benefit can be drawn from previous estimates.

1.3 The Temporal Dimension

As we will see in more detail in chapter 2, most work in visual surface reconstruction is of the instantaneous nature described above. The emphasis of this thesis is on the temporal aspect of surface reconstruction and the discussion above illustrates some of the specific issues that must be considered. At the same time, they can serve as the basis for the set of criteria which we may use to judge the effectivity of a temporal surface reconstruction scheme. Based on the observations made above, the following are minimal requirements for any procedure that we may consider:

1. *Quality improvement:*
   The quality of estimates should improve by combining estimates over time.

2. *Motion transformations:*
   Estimates should be maintained in such a way that a relative motion of camera and scene is accounted for.

3. *Uncertainty representation:*
   Estimates should be maintained along with their uncertainty and the combination of estimates should take the uncertainty into account.

4. *Computational simplicity:*
   Results obtained in previous time steps can be used to reduce the amount of computation necessary by providing initial values for the next step that are close to the solution.

In search of a solution to the temporal reconstruction problem that satisfies the above criteria a look at related problems in other disciplines is enlightening. *Estimation theory* addresses the problem of analyzing a set of measurements to estimate the value of a quantity which is related to the measurements in a defined way. This pertains to the problem at hand, since the image measurements available for surface reconstruction are related to the quantities which describe the surface structure in a known and predetermined way. We have seen this in the case of stereo surface reconstruction above and have referred to this relationship as the visual mechanism.

A brief look at the properties of recursive estimation methods provides insights on a number of the issues mentioned previously.

- Measurements and estimation quantities can be modeled stochastically by probability distributions to describe the effect of errors or measurement noise. Uncertainty can be represented explicitly as the covariance matrix of these probability distributions and can be used to weight measurements of differing quality appropriately.

- Optimal solutions (in terms of the difference between estimate and true value) have been proven and are readily available.

- Recursive estimation theory, in particular, addresses the problem of estimating the internal state of a dynamical system from external measurements. It provides a solution to this problem which incrementally improves an estimate of the system's state with every new measurement that becomes available.

These characteristics make it particularly interesting to investigate techniques from recursive estimation theory for the solution of the temporal surface reconstruction problem. As we will see, casting surface reconstruction in the framework of estimation theory does not force us to abandon the results of instantaneous methods but rather provides a natural way of embedding instantaneous techniques in a temporal estimation scheme and explicitly modeling uncertainty.

1.4 Contributions

The contributions of this research work are as follows:

- The problem of temporal surface reconstruction is formulated and formalized in the framework of recursive estimation theory. This formulation serves as

a unifying theory for previous approaches to temporal estimation and naturally subsumes existing instantaneous procedures by embedding them into a stochastic framework and explicitly representing uncertainty.

- A novel algorithm for the estimation of depth from motion image sequences using optical flow is derived from the temporal surface reconstruction theory, has been implemented and evaluated experimentally. Although this specific problem has been addressed previously, the solution presented here is not restricted to particular types of motion and incorporates prior models of surface structure (smoothness) in a new way based on the stochastic modeling from estimation theory.
- A novel algorithm for the estimation of depth from shading information is derived from the temporal surface reconstruction theory, has been implemented and evaluated experimentally.
- A novel algorithm for the estimation of depth from motion image sequences using the "direct" approach (without optical flow) is derived from the temporal surface reconstruction theory, has been implemented and evaluated experimentally.
- This thesis provides extensive experimental evaluation of the temporal surface reconstruction theory on a variety of real and synthetic images.
- A novel algorithm for the prediction/motion warping of three-dimensional surfaces represented as depth maps to account for the effect of rotational and translational motion on the relative position of observer and surface.
- A detailed computational evaluation of temporal surface reconstruction in terms of complexity, run-time and implementation on parallel processors.

1.5 Structure of the Thesis

Chapter 2: The presentation in this thesis builds on the work in instantaneous surface reconstruction and uses techniques from estimation theory. We begin by introducing some fundamental concepts and notation for imaging. A survey of previous works in instantaneous recovery of structure information from a variety of visual mechanisms is provided. Finally a set of detailed examples of instantaneous surface reconstruction procedures which will later be embedded into the temporal framework are studied here.

Chapter 3: This chapter summarizes previous work on the temporal aspect of visual surface reconstruction. It provides a categorization of research according to the representation and the type of algorithm used.

Chapter 4: Here we recapitulate the essential results from recursive estimation theory that will be used in this thesis. Some interesting and relevant properties of the Kalman filter are presented here.

Chapter 5: This chapter qualitatively outlines the primary contribution of this work: it describes how recursive estimation theory can be applied to the problem of temporal surface reconstruction such that the solution addresses the criteria set forth previously. The resulting solution is an iterative algorithm which consists of two stages for each new image that becomes available: an "update" stage which incorporates the new measurement into the current structure estimate and a "prediction" stage which accounts for relative motion between camera and scene during image acquisition. Only the update stage depends on the visual mechanism which is used to recover structure from image data.

Chapter 6: Here we discuss the initialization of the temporally recursive estimation algorithm and show how a proper choice of initial values permits prior models of surface structure such as smoothness to be imposed on the result of the estimation procedure.

Chapter 7: The prediction stage of the temporal reconstruction algorithm is independent of the particular visual mechanism and is described in chapter 7. The procedure is equivalent to the motion warping of a surface in three dimensions and the subsequent resampling of the warped surface on a regular grid.

Chapter 8: This chapter describes the update stage of the temporal estimator for the case of depth from motion using optical flow. It describes how to choose the matrices and vectors that determine the filter and how the optical flow which is used as a measurement in the formulation can be obtained from images.

Chapter 9: Here the update stage of the temporal estimator for the case of depth from shading is described.

Chapter 10: This chapter describes the update stage of the temporal estimator for the case of depth from motion without the use of optical flow.

Each one of the chapters 8, 9 and 10 shows the result of extensive experimentation with the temporal surface reconstruction algorithms on real images for the particular visual mechanism .

Chapter 11: Here we discuss and evaluate the temporal reconstruction algorithm from various points of view. A detailed analysis of complexity and run-time is provided. We discuss possible implementations on parallel processors and special-purpose hardware and assess possible performance improvements. Finally, a complete list of assumptions and approximations made throughout the thesis is provided to help identify weak points and to serve as the basis for comparisons.

Chapter 12: This chapter contains summarizing and concluding remarks as well as perspectives on future research.

Three appendices at the end of this thesis provide details which are not likely to be of interest to a casual reader but are useful for the purpose of implementation for

example. Appendix A describes the theoretical derivation for the implicit Kalman filter introduced in chapter 4. Appendix B provides the detailed equations for the prediction of depth and covariance that are introduced in chapter 7. Appendix C serves as an implementor's handbook. In it, the the Kalman filter equations for the filter update corresponding to the structure from optical flow case of chapter 8 are worked out in such a detail that they may be immediately reimplemented by the reader. It also provides a set of practical ideas which can support the implementation and make it more flexible.

Instantaneous Surface Reconstruction

Chapter 2

This chapter begins by introducing some basic notation used in the recovery of structure information from images. It then provides a survey of previous work on the instantaneous reconstruction of surfaces from images. Finally, it gives three detailed examples of visual surface reconstruction procedures which operate in an instantaneous fashion in the sense described in the introduction. The goal is to later (in chapters 8, 9, 10) embed these procedures in the Kalman filter framework so that they may operate in a time-continuous fashion.

2.1 Basic Definitions and Notation

We begin with some basic definitions and notation which is used in all of the surface reconstruction procedures. Figure 2.1 shows the imaging situation that we will consider. A viewer-centered *coordinate system* is introduced such that the origin lies at the focal point, the $x-y$ plane is parallel to the image plane and the z-axis points outward along the optical axis. A point $P = [X, Y, Z]$ on the surface of an object in the scene is projected to a point $P' = [x, y, f]$ in the image plane (on the surface of the CCD image sensor), where the coordinates are related by the equations of perspective projection[1]

$$x = f\frac{X}{Z} \qquad \text{and} \qquad y = f\frac{Y}{Z} \tag{2.1}$$

and f is the *focal length* of the camera.

The camera sensor provides measurements of *brightness* values on a rectangular grid which we represent in an array (E_{ij}) where i and j index the rows and columns of the image array as shown in figure 2.2. Note that the coordinate system is centered on the sensor grid while the indexing of the image array begins at (0,0) in the upper

[1]Shading methods use orthographic instead of perspective projection.

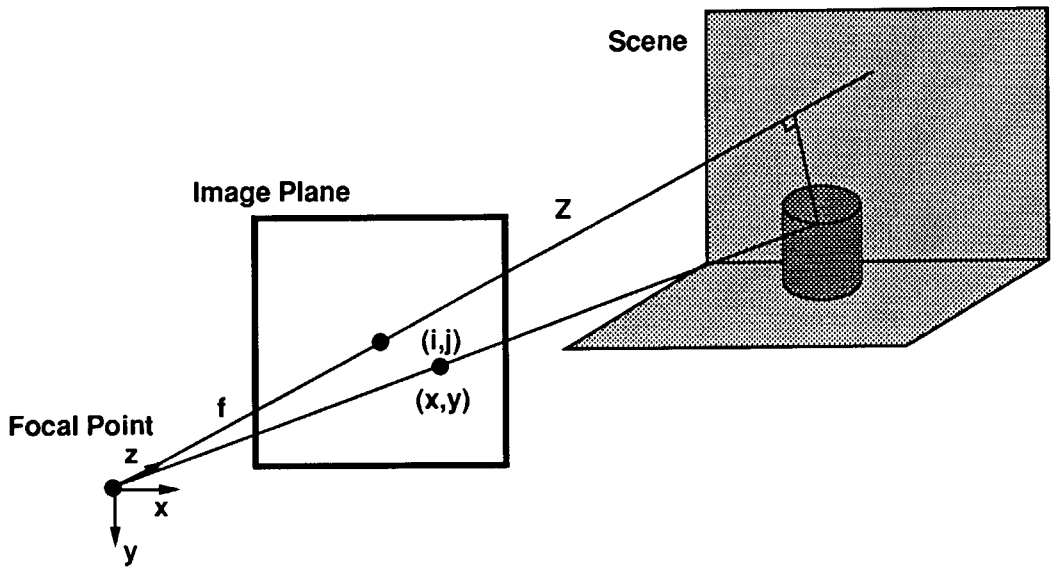

Figure 2.1: The imaging situation.

left corner in row major order. The brightness array has n rows and m columns. If the physical size of the image sensor is $w \times h$, the spacing between sensor elements is

$$\Delta x = \frac{w}{m} \quad \text{and} \quad \Delta y = \frac{h}{n} \tag{2.2}$$

The relationship between physical coordinates and indices in the image array is given by

$$x = (j - \frac{w-1}{2})\Delta x \quad \text{and} \quad y = (i - \frac{h-1}{2})\Delta y \tag{2.3}$$

$$j = \frac{x}{\Delta x} + \frac{w-1}{2} \quad \text{and} \quad i = \frac{y}{\Delta y} + \frac{h-1}{2} \tag{2.4}$$

The Z-coordinate of a point P is the distance from the origin of the coordinate system to the perpendicular projection of P onto the optical axis and is referred to as the *depth* of that point. The array (Z_{ij}) consisting of the depth values corresponding to each of the locations in the image sensor grid is referred to as a *depth map.*

For the description of relative motion between camera and scene we restrict ourselves to rigid body motions. Such motions can be described by a translation vector $\mathbf{t}$ and a rotation matrix $\boldsymbol{\Omega}$ both of which will be given relative to the coordinate system of the camera before the motion. Points P_k and P_{k+1} before and after the motion transformation are related by

$$P_{k+1} = -\mathbf{t} - \boldsymbol{\Omega} P_k \tag{2.5}$$

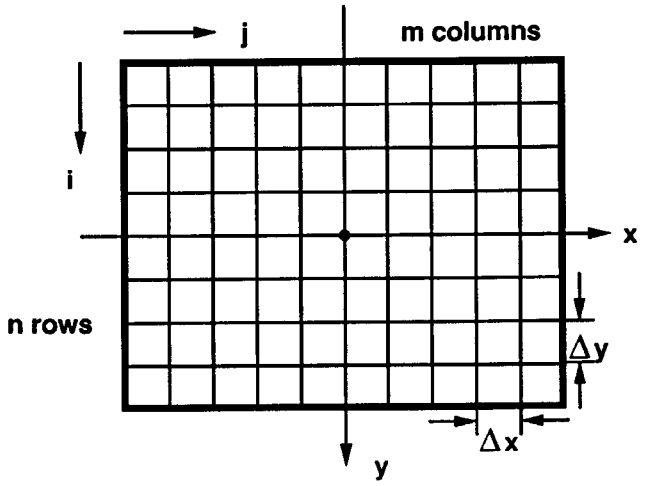

Figure 2.2: The image sensor array and indexing.

In the case of small motions between frames, the rotation can be described by a vector ω:

$$P_{k+1} = -\mathbf{t} - \omega \times P_k \tag{2.6}$$

2.2 Instantaneous Surface Reconstruction

Research in instantaneous reconstruction of surface structure from images can be categorized by the visual mechanism which is used. A brief overview of past work using this categorization in roughly chronological order follows.

2.2.1 Structure from Shading

When the image brightness can be modeled as a known function of the surface structure, this knowledge can be used to infer the surface structure from an single image. Horn [43] first formulated and proposed a solution to this problem in his thesis. A variational approach was presented in [50]. Pentland [77] introduced a greatly simplifed solution approach by assuming that surfaces are locally spherical. Photometric Stereo [49] is an interesting variant in which several images under different illumination conditions are used to eliminate some of the ambiguity inherent in a single image. For a collection of essential papers and complete a bibliography, the reader is referred to Horn and Brooks [46].

2.2.2 Structure from Texture

If the surface of an object is covered by a texture pattern in which texture elements have a constant or known size, the relative size of these patterns in the image can be used to infer the objects shape. This problem was already addressed in Horn's thesis [43] and received detailed attention later by Bajcsy and Lieberman [5] and in the theses of Stevens [89] and Kender [56]. Obviously, this visual cue is restricted to only a specific class of scenes or parts of scenes.

2.2.3 Structure from Stereo

The strongest visual mechanism used by humans to discern the three-dimensional structure is stereoscopic vision. Since we have two eyes, the difference in position at which a point in the scene appears in the two images can be used to determine its depth via a simple triangulation. Marr and Poggio [60], [61] first studied the human stereoscopic vision system and proposed a model for an underlying computational mechanism. Grimson [31], [32] combined this theory with the Marr-Hildreth approach to edge detection to formulate an algorithm that recovered structure information at the location of edges in the image. Using the variational approach later referred to as "regularization", this algorithm recovered dense structure information from two images.

2.2.4 Structure from Motion

When a camera moves relative to a surface, two or more images can be used much in the same manner as in stereoscopic vision: the difference in projected location of a scene point contains information about the three-dimensional location of the point. While the use of small camera displacements can greatly reduce the matching problem which plagues stereo algorithms, the fact that the relative camera motion is either unknown or uncertain introduces other complications. A first class of algorithms assumes that the optical flow, an approximation of the projection of the three-dimensional velocity field has been computed and can be used to recover structure. Examples are Tsai and Huang [102], Bruss and Horn [16], Longuet-Higgins and Prazdny [59], Waxman and Ullman [105], Mitiche [70], Spetsakis and Aloimonos [88] and Heeger and Jepson[34]. Since the computation of the optical flow is expensive, recent proposals have sought to avoid this step and directly extract structure information from image brightness: Kanatani [55] Negahdaripour, Weldon and Horn [75], [48] and Aloimonos and Herve [1] are examples of this approach

2.2.5 Structure from Focus

For a given focal length of the imaging device, only certain scene points at a specific distance (given by the Gaussian lens law) from the camera will appear in focus. By looking at the frequency content of the image regions, the focussed regions can be identified and their depth computed using the lens law. Krotkov [58] analyzed the performance of several criteria to identify image regions in focus and showed how it could be used to recover depth. Pentland [78] addresses the same problem as do Nayar and Nakagawa [73] for the case of rough surfaces. In any case, the use of this method requires the ability to change the focal length of the camera in a very precise and known manner.

2.2.6 Other Related Work

Ikeuchi, Horn and Schunck [50], [47] as well as Grimson [31] had realized the need to impose restrictions on the structure of the surfaces that they reconstructed from various visual mechanisms. This was done both to reduce the effect of measurement noise and to obtain dense information because reconstruction was only done at sparse locations. Terzopoulos [96], [97], contributed much to formalizing this process by introducing the membrane and thin plate surface models and analyzing them theoretically and experimentally. Poggio et al. [80] cast this as the regularization approach to both instantaneous surface reconstruction and the solution of other ill-posed problems in early vision. The deterministic regularization approach which dominates most instantaneous procedures as we will see next was later enhanced by probabilisitic modeling (Geman and Geman [29], Marroquin [63], Blake and Zisserman [10] Poggio, Gamble and Little [79] and Geiger and Girosi [26]) which are reviewed in some more detail in section 3.7.

2.2.7 Depth from Motion using Optical Flow

When a scene moves relative to an observer each scene point can be assigned an instantaneous velocity in space. The projection of this velocity field into the image plane of the observer is called the *motion field* and can be represented by a vector (u_{ij}, v_{ij}) at every pixel location (i, j). A variety of methods exists that estimate an approximation to the motion field which is typically referred to as the *optical flow* from a pair of images: Horn and Schunck [47], Hildreth [42], Nagel and Enkelmann [71], Heeger [33], Anandan [3]. Two images from a motion sequence along with the computed optical flow is shown in figure 2.3. Notice that the optical flow estimates contain errors due to a variety of reasons.

If an algorithm for the computation of the optical flow is available (see section 8.2 for an example), we can assume that the optical flow (u, v) has been computed for

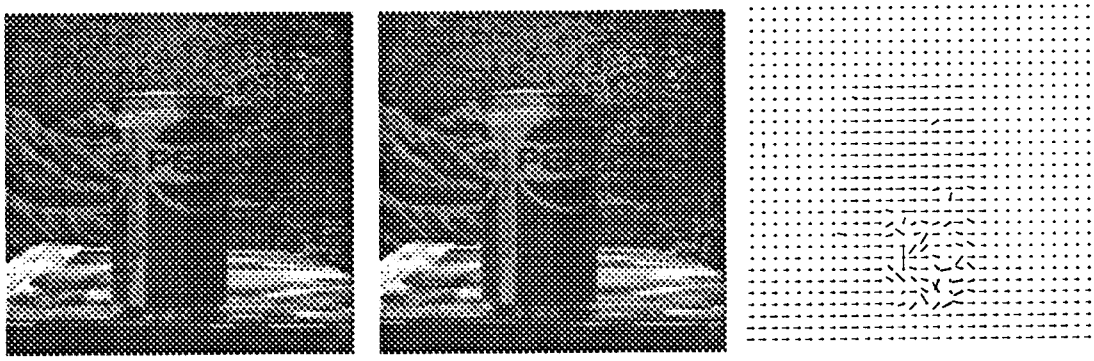

Figure 2.3: The first frame and an optical flow field from the pepsi sequence.

every pixel (i, j). The instantaneous depth from motion algorithm uses the relationship between the optical flow vector of a point and the corresponding depth value Z as the visual mechanism. This relationship is given by the Longuet-Higgins/Prazdny formulas [59]

$$u_{ij} = \frac{-U + x_j W}{Z_{ij}} + A x_j y_i - B(x_j^2 + 1) + C y_i \tag{2.7}$$

$$v_{ij} = \frac{-V + y_j W}{Z_{ij}} + A(y_i^2 + 1) - B x_j y_i - C x_j \tag{2.8}$$

where $\mathbf{t} = [U, V, W]^T$ is the relative translation between camera and scene, $\boldsymbol{\omega} = [A, B, C]^T$ is the relative rotation and (x_j, y_i) is the physical image plane location of pixel (i, j).

We simplify these equations by introducing the inverse depth or *disparity* $d = 1/Z$ (thereby making the problem linear) and abbreviating the known coefficients $\alpha_{ij} = -U + x_j W$ and $\beta_{ij} = -V + y_j W$ as well as the known rotational components (which are independent of depth) $u_{ij}^r = A x_j y_i - B(x_j^2 + 1) + C y_i$ and $v_{ij}^r = A(y_i^2 + 1) - B x_j y_i - C x_j$. This results in

$$u_{ij} = \alpha_{ij} d_{ij} + u_{ij}^r \tag{2.9}$$

$$v_{ij} = \beta_{ij} d_{ij} + v_{ij}^r. \tag{2.10}$$

The surface reconstruction objective is to determine the disparity map (d_{ij}) which satisfies (2.9) and (2.10) when given the optical flow field (u_{ij}, v_{ij}). Mathematically, the problem is overdetermined, as we have $2nm$ constraints for nm unknowns so it may not be possible to satisfy all of the constraints. In addition, the optical flow field is known to contain errors and to reduce their effect on the reconstructed surface, we impose a *smoothness constraint* on the disparity field (d_{ij}). This is achieved by

constructing an energy function the minimum of which is the disparity field which best reconciles the conflicting objectives

$$J(d) = \sum_i \sum_j (u_{ij} - \alpha_{ij} d_{ij} - u^r_{ij})^2 + (v_{ij} - \beta_{ij} d_{ij} - v^r_{ij})^2 + \lambda[(d_{i,j+1} - d_{i,j-1})^2 + (d_{i+1,j} - d_{i-1,j})^2] \tag{2.11}$$

The first two terms enforce the compatibility of d with the measured optical flow values according to (2.9) and (2.10); the last two terms penalize for large differences in neighboring values of d and thereby enforce surface smoothness.

The "optimal" value of d_{ij} is obtained by minimizing the functional J which can be done by iterative relaxation methods (see Golub and Van Loan [30]). The Gauss-Seidel iteration equations in the above case would be

$$d^{n+1}_{ij} = \frac{\alpha_{ij}(u_{ij} - u^r_{ij}) + \beta_{ij}(v_{ij} - v^r_{ij}) + \lambda \bar{d}^n_{i,j}}{\alpha^2_{ij} + \beta^2_{ij} + 4\lambda} \tag{2.12}$$

where $\bar{d}_{ij} = d_{i,j+1} + d_{i,j-1} + d_{i+1,j} + d_{i-1,j}$ and n is the iteration index. The above energy functional method is used widely in surface reconstruction and is also referred to as "regularization" (Terzopoulos [97], Poggio et al. [80]). The surface reconstruction method described here is similar to the one suggested by Bruss and Horn [16] and Barron [7].

2.2.8 Depth from Motion without Optical Flow

The instantaneous procedure for estimating depth from motion described in the previous section requires the computation of optical flow. Since this is a computationally expensive procedure, Horn, Negahdaripour and Weldon [75], [48], [74], [76] developed a *direct* method for the computation of structure from image sequences which does not use the optical flow as an intermediate representation. It is briefly described below.

The *brightness constancy assumption*

$$\frac{dE}{dt} = 0 \tag{2.13}$$

states that the brightness of a fixed point in the scene is unchanged between temporally subsequent image frames. This assumption is not true for most real scenes, motions and lighting conditions but it is a popular approximation which is also the basis for the estimation of optical flow fields (Horn and Schunck [47]). By expanding the absolute derivative in (2.13) we obtain

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = E_x u + E_y v + E_t = 0 \tag{2.14}$$

where E_x, E_y, E_t are the brightness derivatives in spatial and temporal directions and (u, v) is the optical flow introduced previously. Note that the brightness derivatives can be approximated by taking finite differences of image brightness values and are therefore known quantities.

Conceptually, the idea is as follows: The brightness change constraint equation (2.14) links brightness values to optical flow. The motion field equations (2.7), (2.8) link optical flow to rigid body motion and structure. By plugging (2.7), (2.8) into (2.14) we obtain one equation that links image brightness values directly to the desired depth values.

$$\frac{\mathbf{s} \cdot \mathbf{t}}{Z} + \mathbf{v} \cdot \boldsymbol{\omega} + E_t = 0 \tag{2.15}$$

where $\mathbf{t}$ and $\boldsymbol{\omega}$ are the rigid body translation and rotation between camera and scene object and

$$\mathbf{s} = \begin{bmatrix} -fE_x \\ -fE_y \\ xE_x + yE_y \end{bmatrix} \quad \text{and} \quad \mathbf{v} = \begin{bmatrix} E_y f + y(xE_x + yE_y)/f \\ -E_x f - x(xE_x + yE_y)/f \\ yE_x - xE_y \end{bmatrix}. \tag{2.16}$$

Note that $\mathbf{s}$ and $\mathbf{v}$ can be computed completely from image brightness values.

For the purpose of surface reconstruction, we transform the problem into a linear one by using the disparity $d = 1/Z$ instead of the depth Z. Our objective is then to find a disparity surface which satisfies (2.15). Since (2.15) is expected to be only approximately true and since the image brightness measurements are noisy, we impose the additional constraint that (d_{ij}) be smooth so as to reduce the effect of these errors. An energy functional is constructed which formalizes these objectives

$$\begin{aligned} J(d) = \sum_i \sum_j \; & ((\mathbf{s}_{ij} \cdot \mathbf{t}) d_{ij} + \mathbf{v}_{ij} \cdot \boldsymbol{\omega} + E_{tij})^2 \\ & \lambda[(d_{i,j+1} - d_{i,j-1})^2 + (d_{i+1,j} - d_{i-1,j})^2] \end{aligned} \tag{2.17}$$

Again, the first term enforces the compatibility of d with the direct motion constraint (2.15) and the second term penalizes for large differences in neighboring values of d and thereby enforces surface smoothness.

The "optimal" value of d_{ij} is obtained by minimizing the functional J which can be done by iterative relaxation methods. The Gauss-Seidel iteration equations are

$$d_{ij}^{(n+1)} = \frac{\lambda \bar{d}_{ij}^{(n)} - (\mathbf{s}_{ij} \cdot \mathbf{t})(\mathbf{v}_{ij} \cdot \boldsymbol{\omega} + E_{t,ij})}{4\lambda + (\mathbf{s}_{ij} \cdot \mathbf{t})^2} \tag{2.18}$$

in which $\bar{d}_{ij}$ is a local sum of neighbors of d_{ij} and n is the iteration index.

2.2.9 Depth from Shading

In shape from shading (see Horn and Brooks [46]) the visual mechanism is a known functional relationship between the brightness E observed at a location on the surface and the surface normal $\mathbf{n} = [-p, -q, 1]$ there

$$E = R(p, q), \tag{2.19}$$

where R is called the *reflectance function* and $p = Z_x$ and $q = Z_y$ are the partial derivatives of depth. Reflectance functions have been determined for a number of surfaces such as the lunar surface and Lambertian surfaces.

The objective is to recover the surface structure Z from the image brightness E using the known reflectance properties (2.19). As before we introduce an energy function on the depth Z

$$\begin{aligned} J(Z) = \sum_i \sum_j \; & (E_{ij} - R(Z_{i,j+1} - Z_{i,j-1}, Z_{i+1,j} - Z_{i-1,j}))^2 \\ & + \lambda[(Z_{i,j+1} - Z_{i,j-1})^2 + (Z_{i+1,j} - Z_{i-1,j})^2] \end{aligned} \tag{2.20}$$

which in addition to enforcing compatibility of Z with the reflectance function also enforces smoothness to reduce the effect of errors in the reflectance model and the measured brightness values. Note that the partial derivatives have been approximated by finite differences for the purpose of computation.

The iterative Gauss-Seidel solution to the above minimization problem is given by

$$\begin{aligned} Z_{ij}^{n+1} = \; & \frac{1}{4\lambda}[(E_{i,j-1} - R_{i,j-1})R_{p,i,j-1} - (E_{i,j+1} - R_{i,j+1})R_{p,i,j+1} + \\ & (E_{i+1,j} - R_{i+1,j})R_{q,i+1,j} - (E_{i-1,j} - R_{i-1,j})R_{q,i-1,j}] + \bar{Z}_{ij}^n \end{aligned} \tag{2.21}$$

where

$$R_{ij} = R(Z_{i,j+1} - Z_{i,j-1}, Z_{i+1,j} - Z_{i-1,j}) \tag{2.22}$$

$$R_{p,ij} = R_p(Z_{i,j+1} - Z_{i,j-1}, Z_{i+1,j} - Z_{i-1,j}) \tag{2.23}$$

$$R_{q,ij} = R_q(Z_{i,j+1} - Z_{i,j-1}, Z_{i+1,j} - Z_{i-1,j}) \tag{2.24}$$

$$\bar{Z}_{ij} = (Z_{i,j+1} + Z_{i,j-1} + Z_{i+1,j} + Z_{i-1,j})/4 \tag{2.25}$$

As the reflectance function R is usually highly nonlinear, the energy function (2.20) can have many local minima and the convergence of the iterative scheme is a concern; it has been addressed by alternative formulations of the energy function (see Horn and Brooks [46]).

2.2.10 Summary

In this chapter we have seen several examples of how visual mechanisms such as motion and shading (an example for stereo was given in the introduction chapter 1) can be used to reconstruct the three-dimensional structure of a scene represented in terms of a depth or disparity map. We have seen that energy functionals not only provide a simple way to formalize the visual constraint on the surface but also allow to incorporate additional prior information such as surface smoothness. Although the dense representation of a surface in depth/disparity maps is not the only one possible, it elegantly complements the energy functional method and contains the maximum amount of information which we can hope to recover from the images.

Related Work

Chapter 3

This section provides an overview over previous work that addresses the problem of reconstructing three-dimensional scenes from images in a temporal framework. As we have pointed out in the introductory chapter 1, three attributes are useful in characterizing the variety of approaches that have been proposed:

- Representations
- Visual Mechanisms
- Algorithms

We have already explored different visual mechanisms that can be used for the reconstruction of three-dimensional structure in chapter 2. Most previous work in 3D reconstruction is linked to a particular visual mechanism although it has rarely ever been investigated, whether a particular representation or algorithm is better suited for reconstruction using a given visual mechanism. We will attempt to gain some insight into the different alternatives that have been proposed while realizing that it may be impossible to obtain an absolute and objective answer to questions of best representation and best algorithm.

3.1 Representations

Representations for the three-dimensional structure of reconstructed scenes fall into two categories

1. Dense representations. They usually involve some tesselation of the scene or surface area and structural information for each tesselation unit.
2. Sparse representations. Structural information is only provided at selected locations.

For a more complete discussion of representation alternatives for three-dimensional shapes see Faugeras [22].

Dense representations such as depth maps, voxels, surface triangulations etc. have the advantage of representing details of the surface structure and being well-suited to representing the dense information that is available from images. On the other hand, the computational burden is higher for dense representations. But most importantly, while most visual mechanisms such as stereo and motion can provide dense information about surface structure, they provide *useful* information only at a subset of all locations in the image plane (for example near edges or in areas of significant texture).

Sparse representations include feature points, line/edge segments, polyhedral approximations and super-quadrics. In all cases, the representation is restricted to certain selected locations of the three-dimensional scene. This has the advantage of considerably reducing the computational complexity. On the other hand it requires the locations of the sparse structure representations to be identified by some other computational mechanism (which may be rather complex itself). In addition, the sparse representation may not contain sufficient information to perform any useful tasks based on it (such as navigation or recognition).

In comparing these representation alternatives, we can think of a sparse representation as a subset of a dense representation. For example a feature-point based depth estimator will retrieve structure information at selected image plane coordinates while a depth map based estimator will obtain information at every image plane location. Ideally, the dense representation would contain the same information as the sparse representation at the feature point locations.[1] This thought can be carried further: if we restrict ourselves to certain image locations because we believe that useful information can be recovered only there, we should have some notion of what "useful" means. We are implying, that structure information recovered at any image location has a certain "usefulness" and in selecting a sparse representation we restrict ourselves to the locations where this utility index exceeds a certain threshold. As a consequence, we can reconcile sparse and dense representations by maintaining a measure of uncertainty along with each estimate in a dense representation. Then the sparse representation can be extracted at any time by suitably thresholding the uncertainty index.

When the temporal aspect of surface reconstruction is considered, an additional issue gains relevance in comparing representations. In order to combine structure information across image measurements from several frames, it must be possible to determine the correspondence of structure information between frames. For sparse representations, this correspondence must be established explicitly and can involve a rather complex matching procedure. Dense representations have an advantage here,

[1] This is usually not the case, since dense algorithms often involve some influence between spatially "neighboring" pieces of structure information.

since structure information is available everywhere and a corresponding surface point in another frame can be identified by a simple geometric calculation.

3.2 Algorithms

The computational procedures that are used to perform the three-dimensional reconstruction task in a temporal fashion fall into one of the following two categories:

1. Incremental Algorithms: With each new frame of image information that becomes available a current representation of the three-dimensional structure is updated.

2. Batch Algorithms: A given number of frames of image information are accumulated before the processing begins.

Incremental algorithms have a practical advantage in that they require only the latest frame of image information to be present in memory at any given time. On the other hand, the incrementally improved estimate after n frames may not necessarily be the best one that could be obtained, if all n frames were available.

Batch algorithms attempt to achieve the maximal estimate quality by processing all available frames together and determining the values for the structure parameter that best fit the entire sequence. To do this, however, storage for all of the frames of image information is required.

As a consequence, a decision for a particular type of algorithm will depend on two factors: the type of representation of three-dimensional structure we are using and the application in which the structure information is to be employed. A dense structural representation will favor an incremental algorithm, as it is impractical to store dense image information for any sequence of significant length. For a sparse set of feature points, on the other hand, storage may not be a consideration at all. Real-time applications such as navigation require structure information to be available and updated at any time which points towards incremental procedures. On the other hand, a recognition task that operates on three-dimensional information may allow an entire sequence of frames to be accumulated before processing begins.

From a practical point of view, two additional considerations deserve attention: First, video image acquisition devices provide a continuous stream of images that is not limited in length at the outset. This fact does not rule out the use of batch methods, since we could partition the stream into groups of frames that a batch method could process. It does raise the question, however, whether the results obtained from one group of frames can be carried over to the processing of the next group: the very same problem we faced in extending two-frame algorithms to image sequences. Second, batch procedures typically not only have high storage demands

but also involve rather complex numerical optimization procedures, in particular for dense representations.

Ideally, we would like to find an incremental method that yields the same estimate of three-dimensional structure parameters after n frames as a batch procedure that is run on the same n frames. One incremental procedure, the Kalman filter, which is also used in this thesis, is known to have this property under certain circumstances which will be explored in more detail in chapter 4. While these preconditions are rarely met in practice, the recursive estimator may constitute a compromise that enjoys the computational benefits of an incremental algorithm while providing estimates that are comparable in quality to batch procedures.

3.3 Incremental Algorithms for Sparse Representations

The first incremental algorithm that used a sparse represenstion was Ullman's "incremental rigidity scheme" [103], [104]. This method estimates the three-dimensional coordinates of a set of points that are identified as features in a sequence of monocular frames. The matching of points across frames is achieved by determining the transformation and the correspondences that minimize the amount of distortion (maximize the rigidity) of the rigid body on which the feature points lie. The algorithm uses orthographic projection and will therefore fail for any motion with components along the optical axis. In addition, it can be shown that minimizing model distortion may not lead to convergence to the true three-dimensional shape. Nevertheless, this work first identified the importance of recovering structure information in a temporal framework, it first proposed the use of an iterative method and it established important guidelines for later work in this domain.

Broida and Chellappa [13], [14] first suggested the use of Kalman filtering, an incremental stochastic estimation procedure, to obtain feature point location estimates and improve them over time. A set of point features are matched over a sequence of frames and estimates of their corresponding three-dimensional locations are maintained with an extended Kalman filter. The filter formulation also includes the camera motion parameters in the filter state. The important relationship between the incremental and the batch solution to the estimation problem are addressed in this work for the first time. While the early work applied the recursive estimator to the feature point estimation problem in a straightforward way, which led to a highly nonlinear filter, the later work in conjunction with Chandrashekhar [12], [18] seeks to simplify both the dynamical model and the measurement equations. The conclusions are very similar to the ones in the work of Faugeras et al. and to the ones in this thesis: a suitable choice of the state and measurement quantities can greatly reduce the complexity of the estimation procedure and improve the quality of the estimates.

The INRIA vision group has pioneered the use of the Kalman filter for the incremental estimation of parameters for three-dimensional geometric features such as lines and planes. Faugeras et al. [23] used an extended Kalman filter to obtain an estimate of the three-dimensional location of line segments observed in pairs of stereo images. Ayache and Faugeras [4] performed the same task using a trinocular stereo system. Faugeras and Lustman [24] showed that finding corresponding features between frames which is an essential part of any feature-based temporal algorithm is considerably simplified if the scene can be assumed to be piecewise planar. More recently, Jezouin and Ayache [52] investigated the tradeoffs of tracking and estimating point and line segment features in the image plane versus in three dimensions using an Extended Kalman filter. It is possible to trade off complexity of the measurement model for complexity of the matching procedure. Navab et al. [72] explore how stereo measurements of line segments can be combined in a Kalman filter framework with information obtained by tracking points on the segments over time to produce estimates of three-dimensional position. In applying a Kalman filter to the estimation of line segment features, two difficulties arise: First, the line segments must be matched from one frame to the next, in order to combine the information that is contained in these measurements. This is not a simple task, as the above work has revealed. Second, a number of the prerequisites for the application of the Kalman filter such as a linear measurement model, non-correlation between measurement and state and non-correlation between measurements are often not guaranteed for the proposed models. Extensive experimentation, however, seems to show that these concerns can be overcome in practice.

Bharwani, Riseman and Hanson [8] use correspondence at the brightness level to identify matching feature points in monocular sequences. The interesting feature of this approach is that the depth values estimated from feature matches in all frames up to a given time are used to predict the location of the feature in the next frame. This greatly reduces the search effort for the new correspondence, but it does not incorporate the past estimates into the new and most current one; it only uses them as a starting point.

Dickmans [20], [21] and Zapp [111] proposed a Kalman filtering based algorithm for the analysis of monocular image sequences which he termed "4D dynamic scene interpretation". The outstanding feature of this approach is that it was implemented on specially designed hardware and successfully used for the autonomous guidance of vehicles. The algorithm extracts the location of the road boundary within several small rectangular windows as features and maintains an estimate of the vehicle position relative to the edge of the road with the help of a recursive estimator. This estimator feeds directly into a control system for the navigation of the vehicle. Since the algorithm is implemented and integrated into a functioning system, many of the philosophical issues concerning representations and algorithms are resolved against the ultimate criterion: it works.

Wu et al. [108] use an extended Kalman filter to estimate the three-dimensional location of a set of feature points as well as the camera displacement between frames. The particular regularity of the object used in the experiments (a calibration grid) and the restriction to small motions over a short sequence greatly simplifies the necessary matching step which is the key difficulty for sparse representations as we have seen.

Korsten [57] investigates the estimation of rigid body parameters such as plane normals from image sequences using a deterministic least-squares estimation procedure. This can be considered as a deterministic version of the Kalman filtering algorithm used in this thesis. Since the measurement model relating image brightness to the desired parameters is nonlinear, a linearization about the current estimate is performed for each new frame similar to the Extended Kalman Filter.

Sobh and Wohn [86] presented an incremental algorithm that estimates the parameters of a planar surface from optical flow fields. The temporal integration was achieved by computing a weighted average between the parameters estimated from the current flow field and the average of all previous fields.

Källdahl [53] uses a set of features that have been identified and matched over a sequence of monocular frames. He then uses two interleaved recursive estimators to update an estimate for both the three-dimensional locations of the features and the relative camera motion parameters. The interleaved approach is similar to the one outlined previously in [36]. Due to the interleaved scheme, all theoretical results concerning recursive estimators are not applicable and the convergence cannot be guaranteed.

Maybank [68] investigated alternative incremental schemes for the estimation of feature locations from a sequence of monocular images. He compared the extended Kalman filter (see the work of Faugeras et al. and Broida and Chellappa above) which uses a first order Taylor series approximation to the nonlinear measurement equations to a scheme in which second order derivatives of the Taylor series are included. For a simple one-dimensional example it was demonstrated that the second-order approximative filter produces superior estimates.

3.4 Batch Algorithms for Sparse Representations

Iu and Wohn [51] use the tracked locations of a feature point in the image throughout a sequence of frames. The location of the corresponding three-dimensional surface point is modeled with a truncated Taylor series, the coefficients of which are estimated from the feature locations in a least-squares fashion.

Spetsakis and Aloimonos [87] propose an interesting approach based on physical intuition. If observations of a set of feature points are available throughout a sequence of frames, the rays through these feature points will usually not all go through the

same point in three dimensions due to noise. We can hypothesize a set of matches and assign an error to each match which is proportional to the distance between the rays that should actually coincide in one point (as if they had springs attached between them). An optimal set of matches and consequently the three-dimensional feature point locations can be found by minimizing the overall spring energy. This minimization involves the solution of an eigenvalue problem for a matrix with $O(n^2)$ elements where n is the number of frames. Consequently, this can be rather complex in practice and experimental evaluation with real images is necessary.

Shahriat and Price [85] use a single point feature that has been identified and matched across a sequence of monocular frames. They show that the translational component of interframe motion can be eliminated if the five frames are used. The resulting equations can be solved for the rotational motion component using a non-linear least squares procedure. Then structure information can be obtained for other features on the same rigid surface. The key question of how the feature matching is achieved across frames is not addressed in this work. It is also left open, how more than one feature or the use of more than five frames could help in reducing the effect of errors in the matching and thereby improve the quality of the estimates.

Sawhney and Oliensis [82] provide an interesting solution to the case in which a set of feature points is obtained from a monocular sequence of a rotating rigid object. Under these circumstances, the trajectory of the feature points in the image plane is shown to be a conic section. A conic section is fit to the observed features in a least-squares sense and the corresponding three-dimensional conic-section is computed. This involves the solution to an eigenvalue problem for a three-dimensional matrix. Although the central problem of feature-point correspondence is not addressed in this work, results from real images and the computational simplicity indicate that this algorithm can be very useful in practice.

Recently, Tomasi and Kanade [99], [100] described a mathematically elegant and computationally simple method for the estimation of structure from a set of matched feature points in a monocular sequence. They showed that the observed feature point locations are the result of multiplying two matrices, one describing the scene structure and the other describing camera motion between frames. Both can be obtained by a singular value decomposition. This work is ongoing and current limitations such as the orthographic projection model and the batch nature of the processing are being addressed.

3.5 Incremental Algorithms for Dense Representations

Matthies, Szeliski and Kanade [66], [67] independently developed a Kalman filter based algorithm for the dense estimation of depth from a monocular motion sequence

that is very similar to work presented in this thesis (originally [35]). Using the optical flow as a measurement and the inverse depth as the state, structure estimates of unprecedented quality were recovered from real images for pure translation of the camera perpendicular to the optical axis. The theoretical foundation was expanded to allow for more general motions. This work was instrumental in introducing recursive estimation for dense structure representations to the vision community. Matthies [64] later applied the Bayesian estimation framework to the dense estimation of depth from sequences of stereo images. His thesis thoroughly evaluates the method from both a theoretical and an experimental point of view which prompted me to exclude stereo as a visual mechanism from my own investigations. One shortcoming of this work, however, is the prediction stage of the filter, which achieves temporal consistency of estimates. The filter design requires the motion of the stereo cameras to be restricted to a translation along a line perpendicular to the optical axis.

Szeliski [95], [94] explored a volumetric dense representation for three-dimensional structure. Octrees represent tesselations of space in which a unit can either be occupied or empty. Szeliski showed how such a representation can be used in a the recursive, Bayesian estimation of structure from optical flow fields and from the silhouette of a rotating object.

3.6 Batch Algorithms for Dense Representations

Tsai [101] positioned a camera at eight fixed locations in a plane and used the resulting images to calculate a dense depth map by extending the stereo principle to multiple images. He shows both theoretically and through simulation that the quality of the resulting surface estimate is improved considerably as compared to the case in which just two frames are used.

Bolles and Baker [11] and later Yamamoto [109] introduced the concept of the epipolar image. In this approach, the monocular images acquired by a camera translating perpendicular to its optical axis were collected into a "volume" of images. Slices through this volume in the temporal direction can be analyzed without establishing correspondences and provide information not only about rigid body motion and structure but also about occlusion and segmentation. More recently [6] Baker and Bolles have generalized this method to handle more complex motions and to work incrementally as new images become available. In this case, the method becomes similar to previously mentioned Kalman filtering methods.

Subbarao [92], [91] estimated surface structure from measurements of optical flow over time by interpreting both the depth and the optical flow values as functions of time and locally approximating these functions by first and second order Taylor series. Under these assumptions, the structure parameters can be determined in closed form. The practicality of this approach is in question, however, since it involves spatial and

temporal derivatives of the optical flow and experiments with real images were not presented.

Schott [83] investigated how shading and motion could be used for the dense recovery of structure information in a monocular sequence. The central idea was the extension of the brightness constancy assumption of Horn and Schunk [47] by a term that modeled the brightness change due to shading. A second major contribution was the formulation of a least-squares problem that used the enhanced motion/shading constraint equation to recover structure information. This involved the warping of image information across the sequence so as to compensate for interframe motion, a procedure related to the prediction stage of Kalman filter based algorithms. To solve the nonlinear least-squares problem complex numerical optimization procedures were employed. The restriction to simple shading models reduced the practical applicability of the algorithm.

3.7 Other Related Work

This section summarizes previous work that does not deal with the estimation of structure information in a temporal manner but contains representations, algorithms or experiments that put temporal surface reconstruction into perspective.

Stuller and Krishnamurthy [90] were among the first to use Kalman filtering for the estimation of visual information from image sequences. In their case, the optical flow was the state of a dynamical system, the measurements were the image brightness values themselves. Although the assumptions used in this work were fairly restrictive such as a locally planar approximation of the image brightness function, it helped to introduce recursive estimation to the vision community. Rougee et al. [81] designed a Kalman filter that measured the normal component of the optical flow using the brightness constancy constraint and estimated the full flow vector. This approach was shown to significantly improve the results obtained by Hildreth [42] on only two frames. More recently, Black and Anandan [9] proposed a new energy function based approach to optical flow estimation that achieved temporal coherence by computing a weighted average between estimates in sequential frames.

Sethi and Jain [84] addressed the problem that most of the feature-based approaches described above seek to avoid: how to establish correspondence between features over a sequence of frames. The algorithm is based on the assumption that trajectories are smooth and a utility function which measures trajectory coherence is established. A greedy algorithm is then used to assign feature points to trajectories. This algorithm requires the feature points to be available for all frames before processing begins and can therefore be classified as a batch approach.

Franzen [25] introduced the concept of chronogeneous coordinates in which a homogenous coordinate representation of a three-dimensional point is extended to

contain time as an additional component. Although the advantage of this representation in accomplishing the goal of scene reconstruction was not demonstrated in the paper, it is an interesting framework that may help put temporal estimation schemes in perspective.

Thompson and Kearney [98] suggested that geometric representations of three-dimensional structure may be both difficult to compute and unnecessary for most tasks that require vision. Their proposal for "inexact vision" called for research on qualitative representations of structure, specifically structure boundaries, time to collision and direction of translation. Weinshall [106] recently presented results of such research in which it was demonstrated, that information about the Gaussian surface curvature can be extracted in a simple manner from stereo disparities or optical flow vectors and can be used to classify surfaces.

The probabilistic modeling of surfaces for the purpose of visual reconstruction has an extensive history. Geman and Geman [29] first showed how a constraint on a surface modeled by an energy function (such as surface smoothness) can be understood as a probabilistic Markov random field model with the help of Gibb's distribution. In the MRF model, probabilities are assigned to configurations of values of the field representing the depth map in a limited neighborhood. Marroquin [62] explored how this probabilistic model could be used for the reconstruction of surfaces from visual information while imposing a smoothness constraint on the surface but also allowing for discontinuities. Poggio et al. [79] showed how this framework could be used to achieve the integration of information from different visual mechanisms. The major disadvantage of the MRF approach was the enormous computational demand of the simulated annealing procedure required for the computation of the MAP surface estimate. The graduated non-convexity scheme of Blake and Zisserman [10] and the mean-field approximation of Geiger and Girosi [27] were efforts directed at reducing this computational burden.

Bülthoff and Fahle [17] provide evidence, that a Bayesian framework in which measurements are used to obtain an estimate according to the measurement uncertainty is compatible with observations made about the binocular perception of depth in humans.

Probabilistic models such as the Markov random fields and the Kalman filter used in this thesis are based on the premise that stochastic modeling is possible and that the estimate that maximizes the likelihood of coinciding with the true value should be chosen. But as we have seen, most stochastic models are difficult to obtain and approximate at best. Dengler [19] therefore proposes a new objective called the minimum description length criterion. The basic idea is that the best representation is the one which is most concise. Promising experimental results are presented for the application of this idea to the estimation of optical flow.

3.8 Situation of this Thesis

This thesis presents an incremental algorithm for a dense structural representation. More precisely, a Kalman filter based algorithm is used to improve estimates of a depth map over a sequence of frames of arbitrary length. The motives for these choices of representation and algorithm resulted from the observations made in sections 3.1 and 3.2 and the survey of previous work:

- The dense representation was perceived as a superior to a sparse representation since the temporal correspondence problem is reduced to a geometric computation. In addition, when uncertainty is explicitly modeled, a sparse representation can be regarded as a subset of a dense model.

- Only an incremental algorithm is practical for the depth map representation both under storage and computational considerations. In addition, the recursive estimator can be expected to perform very close to any batch procedure.

The methodology and results presented in this thesis are the result of a series of research efforts by the author. I have developed algorithms that use a variety of visual mechanisms for the dense estimation of structure represented as a depth map. They reflect the stages in which the results presented in this thesis evolved. In [35], [37], [36] a Kalman filter based algorithm for the dense estimation of structure from monocular optical flow was described. The algorithm would also provide an estimate of camera motion by alternately estimating structure with the recursive estimator and motion with a least-squares method. The next step [39], [38] was to apply the recursive estimation framework to the reconstruction of surfaces from monocular sequences without the use of optical flow. This work built on the "direct" method of Negahdaripour, Weldon and Horn [75], [48] and produced the first structure estimates from real images for this method. Up to this point, all of the algorithms were designed to process depth estimates at each pixel separately and neglect correlations for reasons of computational simplicity. Influenced by the work of Szeliski [93], the approach was reformulated in [41] and [40] to incorporate prior models of surface structure directly into the filtering algorithm and to use visual cues other than motion for the reconstruction process.

The temporal surface reconstruction framework most notably distinguishes itself from other work in the incremental reconstruction of dense representations in the following points

- It has been formulated and implemented for several visual mechanisms, not only motion. This work shows, that any instantaneous energy-function based reconstruction process can be embedded into the temporal framework presented here.

- Prior models of surface structure are directly incorporated into the filtering process. Previous work had included a "smoothing" step inbetween the update and prediction parts of the filter with both theoretical and practical disadvantages.
- The prediction stage of the filter is entirely new and requires no more restrictions on the motion the camera may undergo between frames.

Recursive Estimation Theory

Chapter 4

In this chapter I will briefly summarize the relevant details of recursive estimation theory and how it applies to problems in visual surface estimation. For details the reader is referred to Gelb [28], Brown [15], Willsky [107] and the dynamical systems literature.

4.1 Linear measurement filter

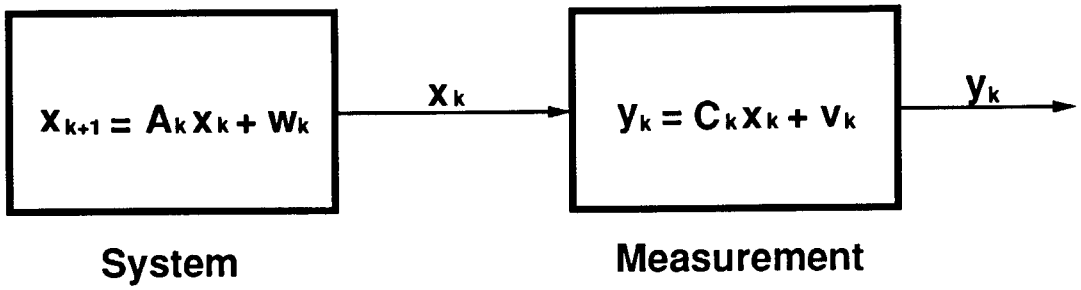

Figure 4.1: A block diagram of a linear dynamical system.

In 1960, Kalman [54] formulated a solution to the following estimation problem: We are given a dynamical system

$$\mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{w}_k \tag{4.1}$$

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{v}_k \tag{4.2}$$

where $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$ and $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$. $\mathbf{x}$ is referred to as the *state* of the dynamical system, $\mathbf{y}$ is called the *measurement*. State and measurement noise must be uncorrelated, i.e. $E[\mathbf{v}_k \mathbf{w}_k^T] = \mathbf{0}$. The block diagram in figure 4.1 depicts the functionality of the dynamical system model.

Informally stated, we have a system that is determined by a state vector $\mathbf{x}$ that changes over time as described by (4.1). While $\mathbf{x}$ determines the behavior of the system, only the measurement vector $\mathbf{y}$ is measurable and its relationship to the state is given by (4.2).

The objective is to estimate $\mathbf{x}_k$ given the measurements $\mathbf{y}_k$ and the above dynamics of the system. In the *Kalman filter* the estimate $\hat{\mathbf{x}}$ is obtained by repeating the following two-step process at each time k:

Update:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}_k^T [\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k]^{-1} \tag{4.3}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \tag{4.4}$$

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\mathbf{P}_k^- \tag{4.5}$$

Prediction:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k^+ \tag{4.6}$$

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k^+ \mathbf{A}_k^T + \mathbf{Q}_k \tag{4.7}$$

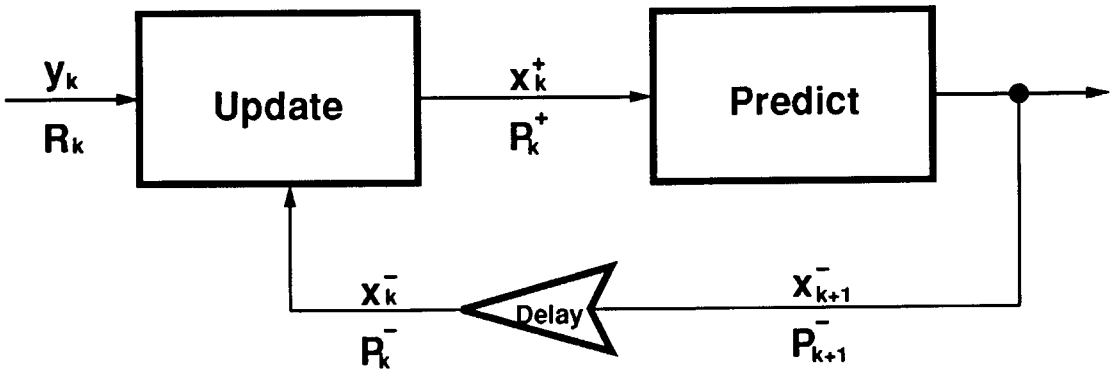

Figure 4.2: A block diagram of the linear Kalman filter.

Figure 4.2 depicts the operation of the Kalman filter in a block diagram. Conceptually, the update stage incorporates a new measurement $\mathbf{y}_k$ into the current estimate of the state $\hat{\mathbf{x}}_k$ by correcting for the difference between actual measurement $\mathbf{y}_k$ and expected measurement $\mathbf{C}_k \hat{\mathbf{x}}_k^-$. The gain $\mathbf{K}_k$ is chosen so that the variance of the new estimate (trace of the covariance matrix $\mathbf{P}_k^+$) is minimal. The prediction stage transforms state and covariance estimate using the known system dynamics. The filter is an optimal estimator in the sense that it minimizes the length of the error vector $\mathbf{x}_k - \hat{\mathbf{x}}_k^+$ which is guaranteed to always decrease.

4.2 Nonlinear measurement filter

Many real-world systems cannot be modeled by a linear measurement equation (4.2). A nonlinear measurement model is

$$\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k) + \mathbf{v}_k. \tag{4.8}$$

In this case the update equations can be reformulated:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{C}_k^T [\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{R}_k]^{-1} && (4.9) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{f}(\hat{\mathbf{x}}_k^-)) && (4.10) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\mathbf{P}_k^- && (4.11) \end{aligned}$$

where $\mathbf{C}_k = d\mathbf{f}/d\mathbf{x}_k$. This result is obtained (Gelb [28]) after Taylor series expansion of $\mathbf{f}$ so that $\mathbf{K}_k$ is only an approximation to the truly optimal gain. It is often referred to as the *extended Kalman filter.*

4.3 Implicit measurement filter

As we will see, some estimation problems in vision cannot even be formulated in the nonlinear form (4.8) since it requires the measurement to be an explicit function of the state vector. In these cases the relationship between state and measurement is implicit

$$\mathbf{g}(\mathbf{y}_k - \mathbf{v}_k, \mathbf{x}_k) = \mathbf{0}. \tag{4.12}$$

This case is not considered in the Kalman filter literature but in appendix A we show that using a Taylor series expansion as for the nonlinear filter above similar update equations can be derived:

$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{C}_k^T [\mathbf{C}_k \mathbf{P}_k^- \mathbf{C}_k^T + \mathbf{D}_k \mathbf{R}_k \mathbf{D}_k^T]^{-1} && (4.13) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}_k) && (4.14) \\ \mathbf{P}_k^+ &= (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k)\mathbf{P}_k^- && (4.15) \end{aligned}$$

where $\mathbf{C}_k = \partial \mathbf{g}/\partial \mathbf{x}_k$ and $\mathbf{D}_k = \partial \mathbf{g}/\partial \mathbf{y}_k$. This result also follows from the implicit function theorem. Just as in the previous case the gain is not truly optimal due to the linearized approximation.

4.4 Alternative formulation of the filter

The form of the filter equations (4.3) - (4.7) is the most common one as it immediately reflects the operation of the estimation procedure. From a computational

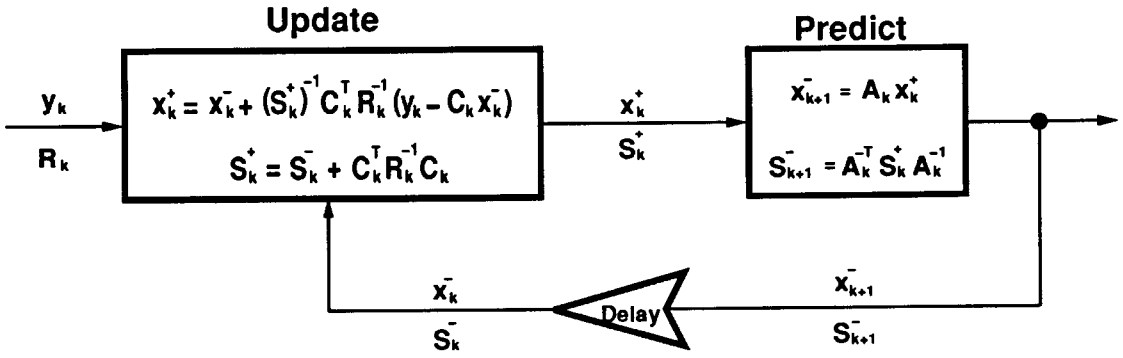

Figure 4.3: A block diagram of the simplified linear Kalman filter.

point of view, however, the number of matrix multiplications and inversions is very high and can be reduced considerably (see [15]). Instead of using the covariance matrix $\mathbf{P}_k$ we introduce its inverse $\mathbf{S}_k = \mathbf{P}_k^{-1}$ which could be termed the "certainty matrix". The filter equations now become
Update:

$$\mathbf{S}_k^+ = \mathbf{S}_k^- + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \tag{4.16}$$

$$\mathbf{K}_k = (\mathbf{S}_k^+)^{-1} \mathbf{C}_k^T \mathbf{R}_k^{-1} \tag{4.17}$$

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{C}_k \hat{\mathbf{x}}_k^-) \tag{4.18}$$

Prediction:

$$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k^+ \tag{4.19}$$

$$\mathbf{S}_{k+1}^- = \mathbf{A}_k^{-T} \mathbf{S}_k^+ \mathbf{A}_k^{-1} \tag{4.20}$$

The operation of the simplified linear filter is depicted in the block diagram of figure 4.3. The same simplifications apply to the nonlinear filter (4.9) - (4.11) without modification and for the implicit filter (4.13) - (4.15) by replacing $\mathbf{R}_k$ with $\mathbf{D}_k \mathbf{R}_k \mathbf{D}_k^T$. In each case the number of matrix operations is reduced considerably.

4.5 Filter update and energy functionals

I will show that the update stage of the filter minimizes a simple energy function on the state vector. The energy function

$$E(\mathbf{x}^+) = \frac{1}{2}(\mathbf{x}^+ - \mathbf{x}^-)^T \mathbf{S}^- (\mathbf{x}^+ - \mathbf{x}^-) + \frac{1}{2}(\mathbf{y} - \mathbf{C}\mathbf{x}^+)^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{C}\mathbf{x}^+) \tag{4.21}$$

has a minimum state $\mathbf{x}^+$ that is "closest" to both the current state $\mathbf{x}^-$ and the current measurement $\mathbf{y}$ each weighted by its covariance matrix (It can be derived by marginalizing the posterior Gaussian distribution of $\mathbf{x}^+$ given the measurement $\mathbf{y}$). In addition, if $\mathbf{S}^-$ is non-diagonal the energy function will enforce a correlation of the elements of the vector $\mathbf{x}^+$ among each other (such as a smoothness constraint).

Differentiation with respect to $\mathbf{x}^+$ yields

$$\frac{\partial E}{\partial \mathbf{x}^+} = (\mathbf{S}^- + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C})\mathbf{x}^+ - \mathbf{S}^-\mathbf{x}^- - \mathbf{C}^T\mathbf{R}^{-1}\mathbf{y} = 0 \tag{4.22}$$

which simplifies to

$$\mathbf{x}^+ = \mathbf{x}^- + (\mathbf{S}^+)^{-1}\mathbf{C}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{C}\mathbf{x}^-) \tag{4.23}$$

if we introduce $\mathbf{S}^+ = \mathbf{S}^- + \mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$. This is just the modified update equation for the Kalman filter (4.17) and $\mathbf{S}^+$ is the updated covariance matrix (4.16).

As we have seen in chapter 2, energy functionals are used in instantaneous surface reconstruction to formulate constraints on the desired surface. The fact that the update stage of the Kalman filter minimizes such an energy functional, will help us to establish the essential connection between the Kalman filter update and conventional surface reconstruction techniques.

4.6 Properties of the Kalman filter

A number of properties of the Kalman filter are noteworthy, beyond the fact that it minimizes an energy function.

- The estimates of the linear Kalman filter can be shown to be *optimal* in the sense that the expected deviation between the estimate and the true value is minimal among all possible estimators (linear or nonlinear). For this reason, the linear Kalman filter is frequently referred to as the *minimum variance* estimator.

- By considering the probability density functions of the estimation and measurement processes, we can show that the estimate of the Kalman filter is the one of *maximum likelihood* in the sense that the conditional probability density of the current estimate given all previous measurement values has a maximum at the value that the Kalman filter computes.

- It can be shown that the linear Kalman filter estimate *converges* to the true value.

In plain terms: the linear Kalman filter will determine the correct value and do so faster than any other estimator.

For our purposes, it is useful to see if these properties of the Kalman filter persist if some of the rather stringent preconditions are dropped. We will consider three cases that are relevant to the work in this thesis:

- If the noise processes on state or measurement are not Gaussian, the Kalman filter is no longer an optimal estimator, however, it is still the optimal linear estimator. In other words, there may be nonlinear estimators that converge faster than the Kalman filter, but among all linear estimators, it is the best.

- If the noise processes of state and measurement vector are not uncorrelated, an alternative formulation of the Kalman filter (see [28]) exists which has all the properties of the original version. Using the original version of the filter on a problem with correlated noise processes (effectively ignoring the correlation) results in a loss of the optimality property of the filter.

- The nonlinear and implicit versions of the Kalman filter use Taylor series approximations to the nonlinear functions. As a consequence, they need not be optimal or even converge. Gelb [28] writes about the extended Kalman filter on page 189: "There is no guarantee that the actual estimate obtained will be close to the truly optimal estimate. Fortunately, the extended Kalman filter has been found to yield accurate estimates in a number of important practical applications."

Recursive Estimation and Temporal Surface Reconstruction

Chapter 5

In this chapter we will investigate how temporal surface reconstruction can be formulated as a recursive estimation problem. At the heart of this formulation is the task of determining the quantities that constitute the state and measurement variables of the dynamical system as well as its state and measurement equations. Other sections in this chapter are devoted to discussions of the update and prediction stages of the Kalman filter for temporal surface reconstruction and the initialization of the filter.

5.1 Intuitive concepts

Intuitively, the temporal surface reconstruction problem maps nicely into a dynamical system so that the desired surface structure can be estimated with a recursive estimation procedure. In our case the surface or structure is unknown and we would like to estimate it. In addition, the surface may change over time due to the possible relative motion between observer and scene. Likewise, a dynamical system has an internal state that changes dynamically over time and is the subject of the recursive estimation. Hence, we can think of the three-dimensional surface structure as the state of a dynamical system.

For the purpose of estimation, we have available to us the sequences of images or derived quantities such as the optical flow. They are related to the unknown surface structure via the "visual mechanism" such as stereo, motion or shading. Similarly, the measurement part of a dynamical system relates the externally available measurement vector to the unknown internal state vector. This suggests that we could interpret the images/optical flow as measurement values and the visual mechanism as the measurement equations of a dynamical system.

These choices for the state and measurement vectors determine a dynamical system that models the imaging process for a temporally dynamic surface. Figure 5.1

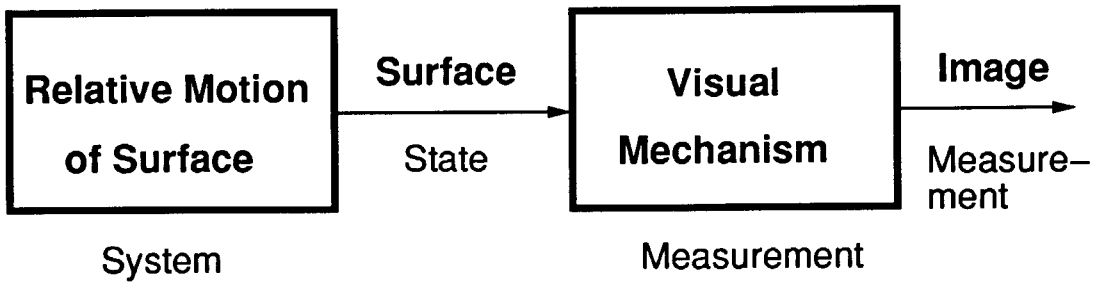

Figure 5.1: The qualitative dynamical system that describes the imaging process for temporally dynamic surfaces.

qualitatively depicts this dynamical system.

The system model (the left block) describes the dynamic change in the state vector which is the depth map in our case. Therefore the system dynamics corresponding to (4.1) must describe the change in depth values from one time instant k to the next. As we are assuming rigidity of scene objects, a temporal change in depth values can only be due to a relative motion between the camera and the surface. Hence, the system dynamics are simply the kinematic transformation (translation and rotation) of points in space.

The measurement model (the right block) describes the relationship between the state (the depth in our case) and the measurable values (the image values). In accordance with (4.2) the measurement model will therefore encapsulate the visual mechanism which does precisely this: it relates structure and image values. As a consequence, the measurement model depends on the specific visual mechanism that is being used for the surface reconstruction. In the case of stereo for example, the measurement equation describes the relationship between stereo matches and the distance to scene points. For shape from shading, the measurement encapsulates the reflectance function that relates brightness and depth gradients. The construction of a recursive estimator will have to be different for each case.

5.2 A Recursive Estimator for the Temporal Surface Model

Now that we have gained some insight on how a temporal surface can be modeled as a dynamical system, we will construct a recursive estimator, that will recover the surface when given the measured image values. This section gives a general description of the filter construction and operation. A detailed discussion of each aspect of the recursive estimator follows in the subsequent chapters.

As we recall from chapter 4 and figure 4.2, the Kalman filter maintains an estimate $\hat{\mathbf{x}}_k$ of the state of a dynamical system along with its covariance $\mathbf{P}_k$ by processing the sequence of measurements $\mathbf{y}_k$ and corresponding covariances $\mathbf{R}_k$. The processing proceeds in two stages for each iteration k: the update and the prediction stage.

5.2.1 The Filter State and Measurement

Our first task is to determine the state and measurement vectors along with their associated covariance matrices such that the resulting recursive estimator will provide a solution to the temporal surface reconstruction problem. The intuitive model of the dynamical system outlined above provides the basis for our choices.

Following the intuitive concepts introduced in section 5.1, the state vector represents the surface structure that we intend to recover. A structural description of a visual scene is given by a depth map (Z_{ij}) (see section 2.1) in which the corresponding depth value is stored for each pixel (i, j) in the image plane. To construct the state vector $\mathbf{x}_k$ of the dynamical system we collect all depth values of frame k into a column vector which is accomplished by adjoining the rows of the depth map

$$(Z_{ij}) = \begin{bmatrix} Z_{0,0} & \cdots & Z_{0,m-1} \\ \vdots & & \vdots \\ Z_{n-1,0} & \cdots & Z_{n-1,m-1} \end{bmatrix} \longrightarrow \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} Z_{0,0} \\ Z_{0,1} \\ \vdots \\ Z_{0,m-1} \\ Z_{1,0} \\ \vdots \\ Z_{n-1,m-1} \end{bmatrix} \tag{5.1}$$

The construction of the state vector from the depth map can be formalized as

$$x_{im+j} = Z_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1. \tag{5.2}$$

Consequently, the covariance matrix $\mathbf{P}$ of the state vector is an $N \times N$ matrix that contains the variances of the depth estimates in the diagonal and the covariances between depth values in the remaining entries. Note the extraordinary size of this matrix, as N is the number of pixels in the image value array! A matrix of this size is beyond computational manageability. However, by using the inverse $\mathbf{S} = \mathbf{P}^{-1}$ in the simplified version of the Kalman filter (see section 4.4) and by taking advantage of the local nature of surface correlation, the filter operates with a sparse, banded representation of the inverse covariance matrix that requires significantly less storage. Details are described in section 6.3.

The measurement vector $\mathbf{y}$ represents the image values that are available to us

through the visual mechanism.[1] The image brightness values are given in an array (E_{ij}) and can therefore be mapped into the measurement vector $\mathbf{y}$ in exactly the same way as the depth map was mapped into the state vector above. We collect all the elements of the image frame (E_{ij}) into a column vector $\mathbf{y}_k$ so that

$$(E_{ij}) = \begin{bmatrix} E_{0,0} & \cdots & E_{0,m-1} \\ \vdots & & \vdots \\ E_{n-1,0} & \cdots & E_{n-1,m-1} \end{bmatrix} \longrightarrow \mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix} = \begin{bmatrix} E_{0,0} \\ E_{0,1} \\ \vdots \\ E_{0,m-1} \\ E_{1,0} \\ \vdots \\ E_{n-1,m-1} \end{bmatrix} \tag{5.3}$$

which can be written explicitly as

$$y_{im+j} = E_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1. \tag{5.4}$$

The covariance matrix $\mathbf{R}$ of the measurement vector represents the uncertainty in the image values. In the case of brightness measurements E_{ij} for example, this uncertainty is due to measurement noise the distribution of which can be assumed to be Gaussian $N(0, \sigma)$ as required by the Kalman filter. If pixel noise is uncorrelated and identically distributed the measurement covariance is $\mathbf{R} = \sigma^2\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. Note that $\mathbf{R}$ is once again a matrix of very large dimensions (here $N \times N$) but remains computationally manageable due to its special structure.

5.2.2 The Filter Update Stage

The next task is to establish the operation of the update stage of the recursive estimator. As shown in figure 5.2, this part of the filter combines the latest measurement $\mathbf{y}_k$ with the current estimate $\hat{\mathbf{x}}_k^-$ to compute the updated estimate $\hat{\mathbf{x}}_k^+$. In terms of the surface reconstruction problem: the update stage will combine the newly obtained image values with the current estimate of surface structure to produce an improved structure estimate.

As a consequence, this update process depends on the specific image values that are used as measurements and the specific visual mechanism that is used to link them to the surface structure values. For each visual mechanism such as stereo, shading or motion, the measurement equation

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad \text{or} \quad \mathbf{y} = \mathbf{f}(\mathbf{x}) \quad \text{or} \quad \mathbf{g}(\mathbf{x}, \mathbf{y}) \tag{5.5}$$

[1] As we have seen in chapter 2 these values need not be exclusively image brightness but could also be optical flow or stereo matches. For the purpose of introducing the concepts, we will work with image brightness in this section and describe alternatives in the following chapters.

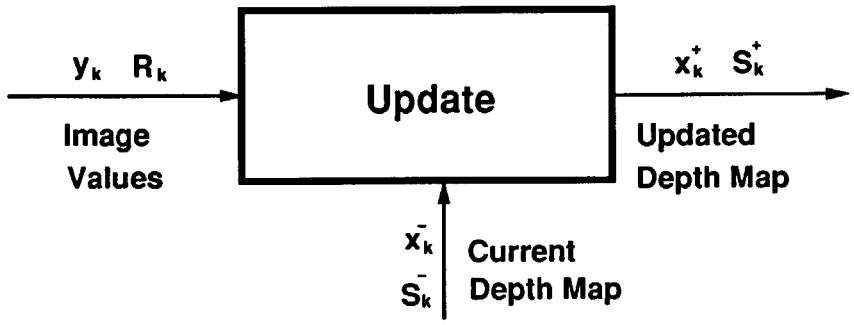

Figure 5.2: The update stage of the Kalman filter for temporal surface reconstruction.

will be different and so will the corresponding update stage. Chapter 8 describes the update stage for temporal surface reconstruction from optical flow measurements in detail. Chapter 9 does the same for the shading mechanism and chapter 10 focuses on the direct motion mechanism.

There are, however, a number of general statements that can be made about the update stage without restriction to a particular visual mechanism that highlight the similarity between the update procedure and the instantaneous surface reconstruction techniques from chapter 2.

The update stage of the simplified recursive estimator (4.16), (4.17), (4.18) involves the matrix inversion of the certainty matrix $\mathbf{S}_k^+$:

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + (\mathbf{S}_k^+)^{-1}\mathbf{C}_k^T\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-) \tag{5.6}$$

We abbreviate the residual $\mathbf{p} = \hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^-$ and $\mathbf{q} = \mathbf{C}_k^T\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)$ and the problem simplifies to

$$\mathbf{p} = (\mathbf{S}_k^+)^{-1}\mathbf{q} \tag{5.7}$$

where $\mathbf{S}_k^+$ is a sparse, banded matrix. [2] Due to this sparse nature of the matrix $\mathbf{S}_k^+$ we can solve (5.7) by an iterative relaxation method such as Jacobi or Gauss-Seidel (see Golub and Van Loan [30]). Gauss-Seidel is the preferred method for serial implementations and is described by the following iteration equation

$$p_i^{(n+1)} = \frac{1}{S_{ii}}(q_i - \sum_{j \epsilon L(i)} S_{ij} p_j^{(n)}) \tag{5.8}$$

where $L(i)$ is the set of indices j with non-zero entries in the ith row of $\mathbf{S}$. The iterative process is initialized with $\mathbf{p} = \mathbf{0}$ (this requires only a small number of iterations in the steady state of the filter estimation) and the state vector can be

[2] $\mathbf{S}_k^+$ is only sparse and banded in general if we make the assumption of a viewpoint independent surface model as explained in section 6.3.

obtained easily from $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{p}$. Algorithmically, the update stage (5.8) of the Kalman filter is therefore exactly the same iterative Gauss-Seidel procedure used for instantaneous surface reconstruction such as (2.12), (2.18) and (2.21).

From section 4.5 we recall that the update stage of a Kalman filter determines the new estimate $\hat{\mathbf{x}}_k^+$ such that it minimizes an energy function

$$E(\mathbf{x}^+) = \frac{1}{2}(\mathbf{x}^+ - \mathbf{x}^-)^T\mathbf{S}^-(\mathbf{x}^+ - \mathbf{x}^-) + \frac{1}{2}(\mathbf{y} - \mathbf{C}\mathbf{x}^+)^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{C}\mathbf{x}^+) \qquad (5.9)$$

which enforces

- Compatibility of the new estimate $\hat{\mathbf{x}}_k^+$ with the current measurement $\mathbf{y}_k$.
- Closeness of the new estimate $\hat{\mathbf{x}}_k^+$ to the previous estimate $\hat{\mathbf{x}}_k^-$.
- "Smoothness" or similar correlation of the elements of $\hat{\mathbf{x}}_k^+$ for appropriate values of $\mathbf{S}^-$.

If we now remember that each one of the instantaneous surface reconstruction procedures in chapter 2 employed an energy function that enforced compatibility of the reconstructed surface with the measured image values as well as surface smoothness we come to the following realization: The update stage of the recursive estimator for a given visual mechanism is identical to the corresponding instantaneous surface reconstruction procedure with two important additional features:

- The recursive estimator additionally enforces "closeness" to the estimate from the last iteration and thereby carries over the information from previous estimates.
- The energy terms are weighted with the covariance matrices so as to explicitly take the uncertainty into account.

In other words: the Kalman filter performs a (stochastic weighted) surface reconstruction at each time k and thereby accomplishes the primary goal of extending surface reconstruction into the temporal domain.

5.2.3 The Filter Prediction Stage

Here we discuss the basic operation of the prediction stage of the recursive estimator. As shown in figure 5.3, this part of the filter takes the updated estimate $\hat{\mathbf{x}}_k^+$ and its covariance/certainty $\mathbf{S}_k^+$ at time k and predicts what these values will be at time $k+1$. For the purpose of temporal surface reconstruction, this means that the prediction stage must transform the estimate of surface structure (depth map) corresponding to the image values in frame k to the depth map corresponding to frame $k+1$.

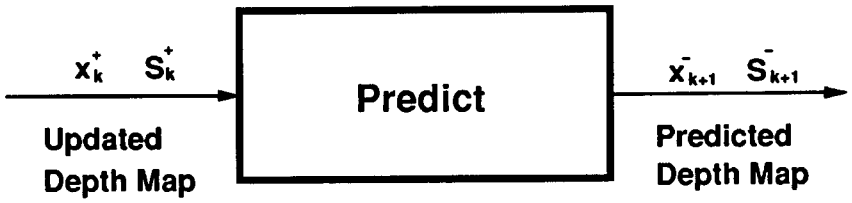

Figure 5.3: The prediction stage of the Kalman filter for temporal surface reconstruction.

Under the rigid body assumption, the only way in which a change in surface structure can occur between frames is by a relative motion of the surface with respect to the camera. At first glance, this appears to be a very simple task. Let the motion between time k and $k+1$ be represented by a translation $\mathbf{t}$ and a rotation $\boldsymbol{\Omega}$. The depth value Z_{ij} at location (x_j, y_i) in the depth map corresponds to a point $P = [xZ/f, yZ/f, Z]$ on the surface. This point will move according to the kinematic equation

$$P_{k+1} = -\mathbf{t} - \boldsymbol{\Omega} P_k. \tag{5.10}$$

The predicted surface can therefore be obtained by applying this transformation to all the points P_k corresponding to entries (i, j) in the current depth map/state vector and filling the predicted depth map with the Z-coordinates of the transformed points P_{k+1}.

This approach is missing one important point: The depth map entry Z_{ij} must be the distance along the optical axis at which a ray through pixel (x_j, y_i) strikes the surface. When a depth map entry Z_{ij} is converted to a point P and transformed according to (5.10) not only does the distance Z change but also the X and Y coordinates of P may change and hence the image plane location (x_j, y_i). The transformed Z may have to be assigned to a new depth map location (i', j'). In essence, it is necessary to resample the warped surface at the depth map grid point locations.

In addition the prediction stage must accomplish the transformation of the covariance/certainty matrix $\mathbf{S}$ in such a way that the resampling of the warped surface is taken into account. Fortunately, the prediction of the depth map and its covariance are independent of the particular visual mechanism that is used to recover structure from image values. Chapter 7 describes the algorithm for prediction of the depth map state vector and the associated covariance in detail.

5.2.4 Filter initialization

Since the filter process is recursive in nature it must be initialized at some point. We must provide initial values for the state $\mathbf{x}_0$ and the associated covariance matrix $\mathbf{P}_0$. The standard procedure in recursive estimation is to initialize the entries of $\mathbf{x}_0$

to zero and the entries of $\mathbf{P}_0$ to ∞. This reflects the fact that the uncertainty in the initial value is very high.

In our particular case, however, some additional information is available and can be used to initialize the state estimate. As the detailed derivation in chapter 6 will show, a proper initialization of the inverse covariance matrix $\mathbf{S}$ can be used to incorporate prior models of the surface structure such as the smoothness constraints that we are already familiar with.

5.3 To Do

This section enumerates the components that constitute a temporal surface reconstruction algorithm. The purpose is two-fold: First, it will give the reader an overview of what to expect in the following detailed description of the parts of the temporal method. Second, since visual mechanisms other than the ones discussed in this thesis can be embedded into the temporal reconstruction framework, this section provides a recipe for the embedding.

1. Pick a state vector $\mathbf{x}$ and an meausurement vector $\mathbf{y}$. For the purpose of surface reconstruction the state vector will be the concatenation of all depth values Z_{ij} or related values such as the disparity as shown in (5.1). The measurement vector must contain values that are directly obtainable from the imaging process.

2. Determine the relationship between the state $\mathbf{x}$ and measurement $\mathbf{y}$ from the particular visual mechanism. Find the matrix $\mathbf{C}$ from either

$$\begin{array}{lll} \mathbf{y} = \mathbf{C}\mathbf{x} & & \text{if the relationship is linear} \\ \mathbf{y} = \mathbf{f}(\mathbf{x}) & \mathbf{C} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} & \text{if the relationship is non-linear} \\ \mathbf{g}(\mathbf{x},\mathbf{y}) = \mathbf{0} & \mathbf{C} = \frac{\partial \mathbf{g}}{\partial \mathbf{x}} & \text{if the relationship is implicit non-linear} \end{array}$$

   In the implicit non-linear case, the matrix $\mathbf{D} = \frac{\partial \mathbf{g}}{\partial \mathbf{y}}$ must also be determined. The linear case is the preferred one and can often be attained through suitable choice of the measurement vector.

3. Determine the covariance $\mathbf{R}$ of the measurement vector. If the measurement vector consists of the brightness values, they can be modeled as independent and identically distributed, and $\mathbf{R} = \sigma_E^2\mathbf{I}$. If the measurement vector is derived from the brightness values, the variances must be propagated through the relationship used in the derivation to obtain $\mathbf{R}$.

This three-step process determines all of the quantities necessary for the update stage of the filtering algorithm (4.16) - (4.18). Examples of this process for three different visual mechanisms are given in chapters 8, 9 and 10.

Two other steps are necessary before a temporal surface reconstruction algorithm can be implemented: the prediction and initialization. Both, however, are independent of the visual mechanism. They are described in chapters 7 and 6 and can be implemented directly regardless of the particular application.

Filter Initialization and Prior Surface Models

Chapter 6

This chapter describes how initial values for the state $\mathbf{x}_0$ and the certainty matrix $\mathbf{S}_0$ can be chosen. Moreover, we show that a proper choice of $\mathbf{S}_0$ will enforce prior models that we may have of the surface structure such as smoothness.

6.1 Surface Models

Surface models represent information about the structure of a surface that we may have even before we begin any surface reconstruction from images. Smoothness is the most common example of a prior model used in computational vision. Surface models have evolved from simple deterministic models to complex stochastic models. The work of Szeliski [93] provides an excellent overview and serves as the basis for the presentation in this chapter.

A common representation of prior models is in terms of energy functions on the depth map entries. They impose a certain constraint on the surface when it is computed as the minimum of the energy function. An example that we have already encountered in chapter 2 on instantaneous surface reconstruction is the *membrane* model

$$E(\mathbf{Z}) = \frac{1}{2}\sum_{i,j}(Z_{i+1,j} - Z_{i,j})^2 + (Z_{i,j+1} - Z_{i,j})^2. \tag{6.1}$$

Introduced by Terzopoulos [96] and Grimson [31], this model is used as a "stabilizer" in the regularized solution of ill-posed early vision problems (Poggio et al. [80]). The effect of the energy function is easily identified: it penalizes for differences between horizontally and vertically neighboring depth map entries and thereby forces the surface to be smooth. If used alone, any constant surface (Z_{ij}) will minimize the energy function (6.1). Alternative surface models such as the *thin plate* model have also been studied extensively in the work cited above and can be used in the same way as the membrane model in the following derivation.

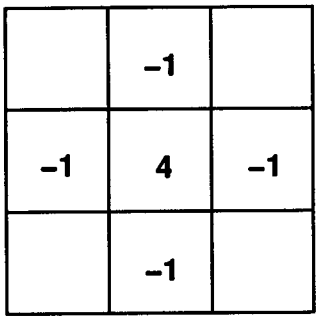

Figure 6.1: The interaction between depth map entries in the membrane surface model.

If we collect all the depth map entries into a one-dimensional vector

$$x_{im+j} = Z_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \tag{6.2}$$

the membrane energy model can be written as a quadratic form

$$E(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{S}\mathbf{x} \tag{6.3}$$

where $\mathbf{S}$ is a sparse banded matrix with 5 entries per row:

$$S_{k,l} = \begin{cases} -1 & l = k-1, k+1, k-m, k+m \\ 4 & l = k \\ 0 & \text{otherwise} \end{cases} \tag{6.4}$$

The matrix $\mathbf{S}$ establishes an interaction between a depth map entry Z_{ij} and its four-connected neighbors as shown in figure 6.1. Note that the construction of the vector $\mathbf{x}$ is identical to the one used for the state vector of the recursive estimator (5.1).

6.2 Probabilistic Surface Models

Surface models represented by energy functions can also be modeled stochastically. This is a prerequisite for the application of probabilistic estimation techniques such as Bayesian estimators that represent uncertainty. The idea is to model the surface by giving a probability distribution for the depth field. Geman and Geman [29] showed how to arrive at a probability distribution for the field values given the energy function constraining its shape. It is the *Gibb's distribution*

$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x})/T)}{\sum_{\mathbf{x}} \exp(-E(\mathbf{x})/T)} \tag{6.5}$$

where T is the "temperature" of the field. If $E(\mathbf{x})$ involves only interactions among neighboring field values as for the membrane energy above the probability distribution describes a *Markov Random Field*, a widely studied stochastic model for surface structure (Marroquin [63], Poggio et al. [79]).

We observe two key features of the stochastic model for membrane-type surfaces: The distribution (6.5) is a multivariate Gaussian distribution and the covariance of the distribution is $\mathbf{P} = \mathbf{S}^{-1}$ (for suitable choice of T). The certainty matrix $\mathbf{S}$ plays the central role in this formulation as it encapsulates the particular prior model that is being imposed on the surface.

6.3 Filter Initialization

The Kalman filter estimation process (4.16) - (4.20) must be initialized by choosing initial values $\mathbf{x}_0$, $\mathbf{S}_0$ for the state vector and the associated certainty matrix. These choices must be made such that

$$\mathbf{x} \sim N(\mathbf{x}_0, \mathbf{S}_0^{-1}) \tag{6.6}$$

at the outset, i.e. such that they determine the prior stochastic model of surface structure. As we have seen above, the choice of the certainty (inverse covariance) $\mathbf{S}$ determines the surface model. If, for example, we set $\mathbf{S}_0$ as in (6.4), we will impose membrane smoothness as the prior model on the surface. This is done throughout the experiments described in this thesis. Other prior models such as the thin plate can be imposed in the same way described above.

As mentioned before, any surface of constant depth

$$x_{im+j} = Z_{ij} = Z_0 \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \tag{6.7}$$

minimizes the constraint energy for the membrane model (6.1). In the experiments in this thesis, we chose Z_0 to be an average value of depth in the scene as it may be obtained from a crude depth sensor such as an ultrasonic device.

The sparse and banded structure of the matrix $\mathbf{S}$ is of particular importance for the calculation of the Kalman filter update (4.16) - (4.18) which requires the inversion of $\mathbf{S}$. In general, the inversion is a formidable task considering that $\mathbf{S}$ is of size $nm \times nm$. If $\mathbf{S}$ is sparse and banded, the inversion can be accomplished through an iterative process such as (5.8) that costs only $O(nm)$.

Although the sparse banded nature of $\mathbf{S}_k$ is not preserved by the filter prediction equations, we can make the simplifying assumption that prior models such as surface smoothness are independent of the camera viewpoint and should therefore be unaffected by the Kalman filter equations. This allows us to separate the inverse covariance matrix $\mathbf{S}_k$ at time k into a diagonal matrix $\tilde{\mathbf{S}}_k$ and the prior matrix $\mathbf{S}$:

$$\mathbf{S}_k = \tilde{\mathbf{S}}_k + \mathbf{S}. \tag{6.8}$$

Note that we can safely replace S_k with $\tilde{S}_k$ in the filter update equation (4.16) by subtracting $\tilde{S}$ on both sides while we ignore the transformation of off-diagonal elements in the (4.20). This is an important contribution to the computational feasibility of the temporal surface reconstruction scheme.

Filter Prediction

Chapter 7

This chapter describes the prediction stage of the recursive estimator for temporal surface reconstruction. The algorithm accomplishes the warping of a depth map corresponding to a rigid body motion. In addition, the prediction of the covariance matrix corresponding to the depth map state vector is described. As an alternative, a computationally less expensive approximative prediction algorithm concludes the chapter.

7.1 Prediction of the Depth Map

The prediction stage (4.6), (4.7) of the filter process must account for changes in the state vector that occur between sequential measurements. For the depth estimation process this is equivalent to a change in depth from one frame to the next which can only be due to a relative motion between scene and imaging system. The prediction process must therefore transform all depth map entries according to this relative motion. This is a purely geometric transformation.

We are given a depth map (Z_{ij}) at time k, the relative motion between camera and scene from k to $k+1$ as a translation vector $\mathbf{t}$ and a rotation matrix $\boldsymbol{\Omega}$ and the projection geometry (focal length f, pixel spacing Δx, Δy and image size $w \times h$). This situation is shown in figure 7.1. The objective is to determine the depth map (Z'_{ij}) at time $k+1$.

The idea of the algorithm is as follows: each entry Z_{ij} in the depth map corresponds to a point P_{ij} in space via perspective projection. The transformation of P_{ij} from time k to $k+1$ can then be accomplished by applying the rotation matrix and the translation vector. Since the transformed points P'_{ij} will not necessarily project back onto depth map grid points at time $k+1$ it becomes necessary to interpolate and resample the surface.

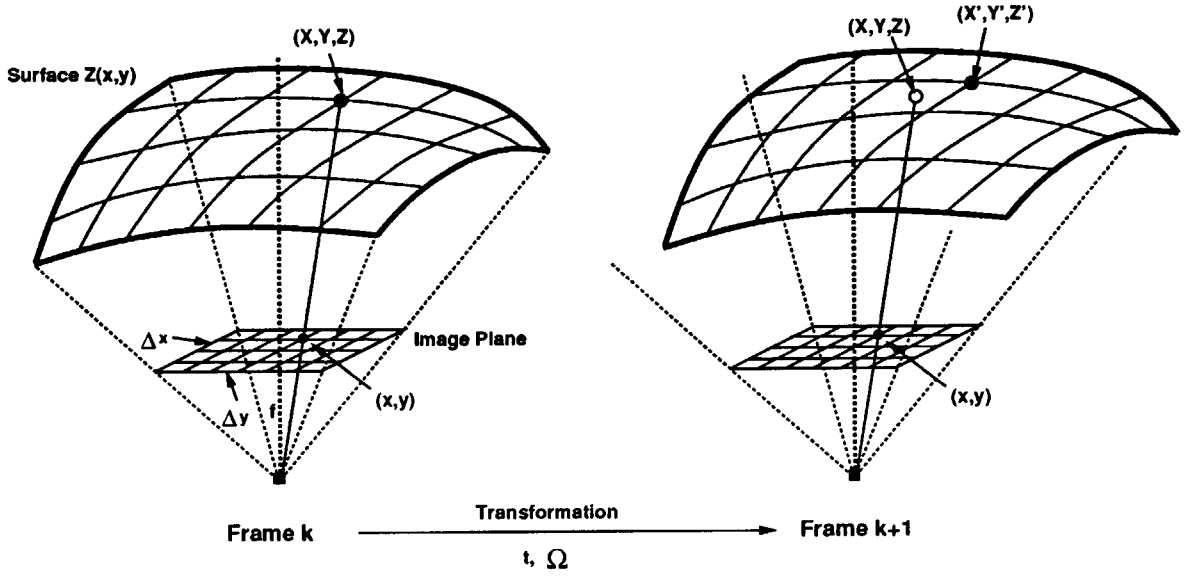

Figure 7.1: A surface corresponding to a depth map and the effect of a motion transformation.

I will now describe the steps of the algorithm in more detail:

1. *Inverse Projection:*
   Each depth map entry Z_{ij} corresponds to a depth value at physical location

$$x_j = (j - (w-1)/2)\Delta x \qquad \text{and} \qquad y_i = (i - (h-1)/2)\Delta y \tag{7.1}$$

   in the image plane. This represents a point $P_{ij} = [X, Y, Z]^T$ in space, the coordinates of which are given by inverting the perspective projection equations:

$$X_{ij} = \frac{x_j Z_{ij}}{f} \qquad \text{and} \qquad Y_{ij} = \frac{y_i Z_{ij}}{f} \tag{7.2}$$

2. *Warping:*
   Now we can account for the relative motion between frames by applying the rotation and translation transformations to each point P_{ij}:

$$P'_{ij} = -\mathbf{t} - \boldsymbol{\Omega} P_{ij} \tag{7.3}$$

3. *Resampling:*
   The straightforward approach would be to project the points P'_{ij} back into the

image plane to obtain the new depth map. However, this will not maintain the representation of depth on a grid corresponding to the image pixels since the warped points do not necessarily project onto grid point locations (shown in figure 7.1 on the right). It is necessary to resample the warped depth map at grid point locations.

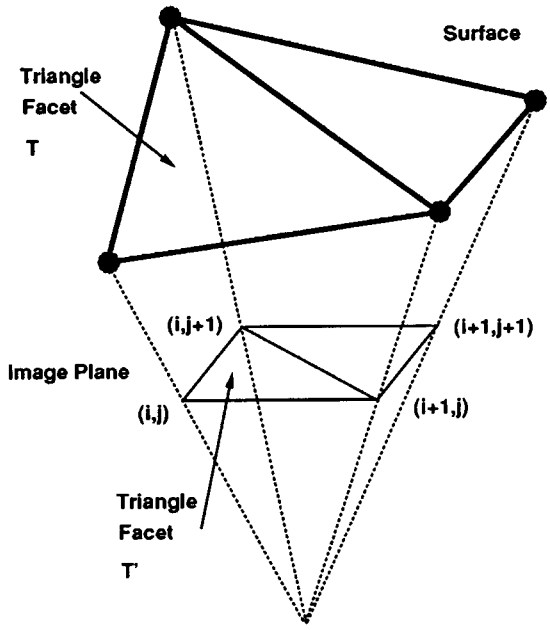

Figure 7.2: Triangular facet subdivision.

To accomplish this we group the pixel locations (i, j) into triplets by dividing each grid square into two triangles as shown in figure 7.2. We approximate the surface between the 3D points P_{ij} corresponding to a triplet by a plane. Although other surface approximations are possible (bilinear, bicubic etc.) they make the resampling more difficult. After warping (7.3) each triplet still defines a plane so that the warped surface can be approximated by its planar triangulation.

To resample the triangulated surface at grid point (x_j, y_i) we must determine the intersection of the ray through the grid point with the closest spatial triangle in the warped surface approximation. This is accomplished by initializing the new depth map with ∞ everywhere and repeating the following steps for each warped triplet of points P'_{ij}:

- Project all three corner points $P' = [X', Y', Z']^T$ back into the image plane

using

$$x' = f\frac{X'}{Z'} \quad \text{and} \quad y' = f\frac{Y'}{Z'} \tag{7.4}$$

- Determine all the grid locations (i,j) that lie within the backprojected triplet of points (x',y').
- For each grid location (i,j) identified in the previous step find the intersection of the ray through that grid point with the spatial triangle under consideration. If the Z value of the intersection point is less than the one stored in the depth map at (i,j) then replace it by Z.

We notice that the ray through a given grid point can have multiple intersections with the warped surface which means that part of the surface has been occluded by another region. In this case the algorithm chooses the physically correct value: the one closest to the camera.

4. *Unassigned grid points:*
The resampling procedure will not necessarily assign a depth value to each grid point that physically corresponds to new areas in the scene that became visible due to the camera motion. In general, we can say nothing about the correct depth value at these locations. If, however, we assume that the surface is somewhat smooth then we can extrapolate the depth values at these points from known values at neighboring locations.

So far I have described how the prediction of depth corresponding to (4.6) can be accomplished. Due to the discrete grid representation of the depth map the piecewise linear surface approximation was introduced and hence the prediction algorithm only approximates the Kalman filter state prediction (4.6).

7.2 Prediction of the Depth Covariance

The simplified Kalman filter represents uncertainty in the depth map state vector by the inverse $\mathbf{S} = \mathbf{P}^{-1}$ of the covariance matrix (certainty matrix). The prediction stage of the recursive estimator must therefore determine the certainty $\mathbf{S}_{k+1}$ after a possible motion transformation when given the certainty $\mathbf{S}_k$.

As we have seen in chapter 6, the matrix $\mathbf{S}_k$ can be decomposed

$$\mathbf{S}_k = \tilde{\mathbf{S}}_k + \mathbf{S}. \tag{7.5}$$

where $\tilde{\mathbf{S}}_k$ is a diagonal matrix containing the inverse variances of the depth values in the state vector and $\mathbf{S}$ is the sparse banded matrix that represents the prior model that we may have imposed on the surface. Our prediction algorithm will only transform the inverse depth variances while leaving the prior model unaffected.

It is possible to propagate the variance values through the prediction equations used for the depth values above. This requires a Taylor series approximation of the nonlinear projection equations and the assumption of stochastically independent depth map entries (which is not the case if a non-trivial prior model is imposed). The description of the algorithm is rather lengthy and has therefore been included as appendix B.

A more practical approximation was proposed by Szeliski [93] and seems to work just as well: Due to the piecewise planar surface approximation in the prediction of the depth values, a small error is introduced. The variance σ_Z^2 of the depth value Z reflects the uncertainty in the prediction and should therefore be inflated by a small multiplicative factor ϵ to reflect the higher uncertainty. For the update of the diagonal inverse covariance matrix $\tilde{\mathbf{S}}_k$ this means

$$\tilde{s}_{ii}^{(k+1)} = \frac{1}{1+\epsilon}\tilde{s}_{ii}^{(k)} \tag{7.6}$$

The depth map prediction algorithm described previously may leave some locations on the image grid "unassigned" corresponding to regions in the image that appear in frame $k+1$ and were not visible in frame k. Although the algorithm eventually fills in these locations with neighboring values, the filled-in depth information has very high uncertainty and the diagonal values in the predicted certainty matrix $\mathbf{S}_{k+1}$ must reflect this. To accomplish this, the diagonal entries in $\mathbf{S}_{k+1}$ that correspond to formerly "unassigned" grid locations in the depth prediction are set to the same value that $\mathbf{S}$ was initialized with (see chapter 6).

7.3 An Efficient Approximative Prediction Algorithm

The prediction algorithm for depth and certainty described above is the computationally most expensive part of the temporal surface reconstruction procedure (see section 11.1). This is mainly due to the resampling step. This section describes an approximative resampling algorithm that is considerably less complex and yields good results in practice.

As before, we are given a depth map (Z_{ij}) on a rectangular grid. Suppose that the depth map has been warped according to the interframe motion as described in steps 1 and 2 of the prediction algorithm in section 7.1. The new depth value Z_{ij} as well as its new location (x_j, y_i) have been calculated but x_j and y_i may not coincide with the coordinates of any grid point.

For the computation of the new depth map value $Z(x,y)$ at a grid location (x,y) we will consider all of the warped depth values Z that are closer to (x,y) than to

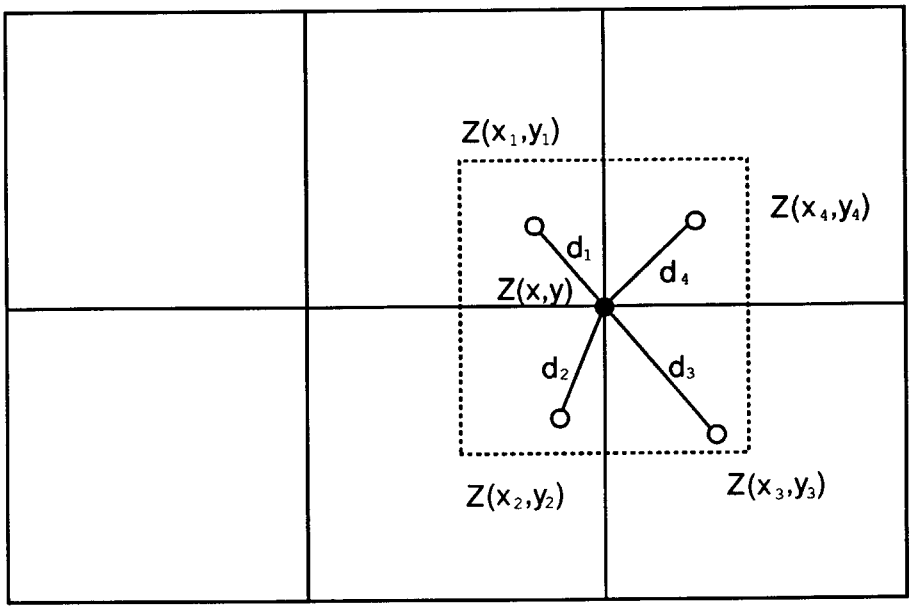

Figure 7.3: Distance weighted resampling of depth values.

any other grid location. Suppose that there are n such depth values and that they are subscripted $k = 1, \ldots, n$. This is shown in figure 7.3 for $n = 4$.

The new depth map value $Z(x, y)$ is computed as the weighted sum of the $Z(x_k, y_k)$

$$Z(x, y) = \sum_{k=1}^{n} w_k Z(x_k, y_k). \tag{7.7}$$

A good weighting function should fulfill the following requirements:

- $0 \leq w_k \leq 1$
- $\sum_{k=1}^{n} w_k = 1$
- The w_k should decrease as the distance $d_k^2 = (x_k - x)^2 + (y_k - y)^2$ decreases.

We therefore choose the weighting factors to be

$$w_k = \frac{\frac{1}{d_k^2}}{\sum_{k=1}^{n} \frac{1}{d_k^2}}. \tag{7.8}$$

This clearly fulfills all of the above requirements but some special cases must be considered. Suppose all n estimates involved in the interpolation have equal distance $d_k = d$ from the grid-point. In this case

$$w_k = \frac{\frac{1}{d^2}}{\sum_{k=1}^{n} \frac{1}{d^2}} = \frac{1}{n} \tag{7.9}$$

which means that all estimates are weighted equally as one would expect. A more tricky case occurs when some number l of the estimates are actually located at the grid-point, i.e. $d_k = 0$ for $k = 1, \ldots, l$. For estimates on the grid-point we can rearrange the expression for the weights (7.8) to obtain

$$w_k = \frac{1}{\frac{d_1^2}{d_k^2} + \cdots \frac{d_l^2}{d_k^2} + \sum_{k=l+1}^{n} \frac{d_l^2}{d_k^2}} \rightarrow \frac{1}{l} \tag{7.10}$$

as $d_k \rightarrow 0$ for $k = 1, \ldots, l$. Similarly we rearrange the expression for the weights in the case of an estimate that does not coincide with the grid-point

$$w_k = \frac{\frac{d_l^2}{d_k^2}}{\frac{d_1^2}{d_1^2} + \cdots \frac{d_l^2}{d_k^2} + \sum_{k=l+1}^{n} \frac{d_l^2}{d_k^2}} \rightarrow 0 \tag{7.11}$$

as $d_k \rightarrow 0$ for $k = 1, \ldots, l$. In other words we ignore all estimates that are not on the grid-point and obtain the interpolated value as the mean of the estimates located at the grid-point.

The idea of distance-weighted resampling can be implemented in an algorithm that runs in time proportional to the number of entries in the depth map (Z_{ij}) by maintaining a weight array (w_{ij}).

1. Initialize the predicted depth map (Z_{ij}^{k+1}) and the weight array (w_{ij}) to zero.
2. For each i and each j do the following:
   (a) Calculate the point $P = [X, Y, Z]$ corresponding to Z_{ij}^{k}. Apply the rigid body motion transform to P and project it back into the image plane (see section 7.1). The result is Z at location (x, y).
   (b) Calculate the (possibly non-integral) grid coordinates (i, j) corresponding to (x, y) using equation (2.4) and the closest integral grid point coordinates $(m, n) = (r(i), r(j))$ where $r()$ is the rounding operator. The physical grid point coordinates (p, q) closest to the warped depth map value are obtained from (m, n) via equation (2.3).

(c) Calculate the distance between depth map value and closest grid point $d = (x-p)^2 + (y-q)^2$.
If d is not zero then add $1/d$ to w_{mn} and add Z/d to Z_{mn}^{k+1}.
If d is zero and w_{mn} is less than zero, then decrement w_{mn} and subtract Z from Z_{mn}^{k+1}.
If d is zero and w_{mn} is greater or equal zero, then set $w_{mn} = -1$ and set $Z_{mn}^{k+1} = -Z$.

3. For each m and each n do the following:
If w_{mn} is positive, set $Z_{mn}^{k+1} = Z_{mn}^{k+1}/w_{mn}$.

Upon completion, the depth map (Z_{mn}^{k+1}) will contain the predicted depth values where w_{mn} is non-zero. If w_{mn} is zero, the location (m,n) is "unassigned" and a filling algorithm such as the one outlined in section 7.1 can be applied. If we apply the same calculations to the diagonal entries of the certainty matrix $\mathbf{S}^k$, the algorithm also accomplishes the prediction of the certainty matrix.

Filter Update: Depth from Motion Using Optical Flow

Chapter 8

In this chapter we will explore the application of the temporal surface reconstruction algorithm to the problem of estimating depth from motion using optical flow. A corresponding instantaneous algorithm that accomplishes this task when given a single measurement of the optical flow was described in section 2.2.7. Only the update stage of the recursive estimator is dependent on the visual mechanism being employed. Therefore, the focus in this chapter is exclusively on that part of the algorithm.

8.1 The Update Algorithm

When an observer moves relative to a scene, each scene point can be assigned an instantaneous velocity in space. The projection of this velocity field into the image plane of the observer is called the *optical flow* and can be represented by a vector (u_{ij}, v_{ij}) at every pixel location (i, j). The relationship between the optical flow, the motion $\mathbf{t} = [U, V, W]^T$, $\boldsymbol{\omega} = [A, B, C]^T$ and the inverse depth $d = 1/Z$ is given by the Longuet-Higgins/Prazdny formulas [59]

$$\begin{aligned} u_{ij} &= (-U + x_j W)d_{ij} + Ax_j y_i - B(x_j^2 + 1) + Cy_i \qquad (8.1) \\ &= (-U + x_j W)d + u_{ij}^r \\ v_{ij} &= (-V + y_j W)d_{ij} + A(y_i^2 + 1) - Bx_j y_i - Cx_j \qquad (8.2) \\ &= (-V + y_i W)d + v_{ij}^r \end{aligned}$$

where (x_j, y_i) is the physical image plane location of pixel (i, j). We will assume that the dimensions of the optical flow fields (u_{ij}), (v_{ij}) and the disparity field (d_{ij}) are $n \times m$.

The state vector $\mathbf{x}$ is constructed by concatenating the rows of the disparity field to an nm-dimensional vector:

$$x_{im+j} = d_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \qquad (8.3)$$

The measurement vector $\mathbf{y}$ is constructed by combining the adjoined rows of both the (u_{ij}) and (v_{ij}) fields minus the rotational components (which are independent of depth) into a $2nm$-dimensional vector:

$$y_{im+j} = u_{ij} - u^r_{ij} \qquad i = 0, \ldots, n-1 \tag{8.4}$$

$$y_{im+j+nm} = v_{ij} - v^r_{ij} \qquad j = 0, \ldots, m-1 \tag{8.5}$$

With these choices we can write the motion field equations (8.1), (8.2) for all pixels in the form of a linear Kalman filter measurement (4.2) $\mathbf{y} = \mathbf{C}\mathbf{x}$ where $\mathbf{C}$ is a $2nm \times nm$ matrix with

$$C_{k,im+j} = \begin{cases} -U + x_j W & k = im + j \\ -V + y_i W & k = im + j + nm \\ 0 & \text{otherwise} \end{cases} \tag{8.6}$$

Finally we must consider the covariance matrix $\mathbf{R}$ of a measurement $\mathbf{y}$. We assume that an optical flow measurement (u_{ij}, v_{ij}) has an error distribution that is Gaussian with covariance

$$\mathbf{R}_{ij} = \begin{bmatrix} p_{ij} & r_{ij} \\ r_{ij} & q_{ij} \end{bmatrix} \tag{8.7}$$

where p, q and r depend on the algorithm that is used to compute the optical flow measurements from image brightness values. An example of such an algorithm and the resulting covariance values is given in section 8.2.

The measurement covariance matrix $\mathbf{R}$ can now be constructed by collecting all the covariance matrices of individual optical flow measurements, under the assumption that optical flow measurements are uncorrelated:

$$R_{kl} = \begin{cases} p_{ij} & k = im + j, l = im + j \\ q_{ij} & k = im + j + nm, l = im + j + nm \\ r_{ij} & k = im + j, l = im + j + nm \text{ and } k = im + j + nm, l = im + j \\ 0 & \text{otherwise} \end{cases} \tag{8.8}$$

The above choices for the state $\mathbf{x}$, the measurement $\mathbf{y}$, its covariance $\mathbf{R}$ and the measurement matrix $\mathbf{C}$ completely determine the update stage of the Kalman filter. By plugging these values into the update equations (4.16), (4.17), (4.18), the filter update can be performed. The detailed equations which are obtained after these algebraic manipulations have been summarized in appendix C so that the reader can easily implement the temporal surface reconstruction from optical flow.

Note that the resulting Kalman filter is linear. However, the Kalman filter convergence and optimality properties of section 4.6 do not necessarily apply for the following reasons: The error in the optical flow estimates will generally not have a Gaussian distribution. Moreover, neighboring optical flow estimates usually exhibit correlation.

The formulation of the estimation of dense depth from optical flow in terms of a Kalman filter was first done by Matthies, Szeliski and Kanade [67] and by Heel [37] as a special case of the above presentation. Both assumed non-correlation as the prior model (i.e. all non-diagonal elements of **S** are zero). In this case each disparity map entry (i,j) can be updated independently from all others

$$S^+ = S^- + a^2p + b^2q + 2abr \tag{8.9}$$

$$x^+ = x^- + [a(pu + rv - ad) + b(ru + qv - bd)]/S^+ \tag{8.10}$$

where we have omitted the subscripts i,j and have abbreviated $a = U + x_jW$ and $b = V + y_iW$. Consequently, the iterative relaxation algorithm is avoided and only a small number of multiplies and adds are required at each pixel. Both formulations then tried to impose the surface smoothness constraint by explicitly smoothing the depth map after the update stage of the filter.

8.2 Computation of Optical Flow and its Covariance

The temporal surface reconstruction from optical flow requires that the optical flow field $(u_{ij}, v_{ij})^T$ has be calculated at every pixel (i,j). Algorithms that accomplish this task have been studied in detail: The differential method by Horn and Schunck [47], the edge-based approach of Hildreth [42], Nagel and Enkelmann's enhanced second-order differential approach [71], Heeger's spatio-temporal filters [33], and Anandan's correlation-based scheme [2] are prominent examples. The recursive estimator for temporal surface reconstruction does not require the use of any one particular algorithm. However, the optical flow algorithm employed will determine the uncertainty in the measurement provided to the Kalman filter and hence the covariance matrix **R**. For this reason and for the sake of providing an example, a matching optical flow algorithm similar to the one by Anandan is described here and the derivation of the measurement covariance **R** is given.

Let us begin by recalling that the optical flow is an estimate of the projection of the 3-D velocity field into the image plane. An optical flow field will therefore contain a vector (u_{ij}, v_{ij}) for every point P projected into the image describing which point P' it projects to in the next image.

The main idea that is exploited in the sum-of-squared differences (SSD) optical flow is depicted in figure 8.1. For a given point $P = (x,y)$ in image 1 we wish to determine where it moves to in image 2. We assume two things:

- The interframe displacements do not exceed a certain number of pixels in each dimension.

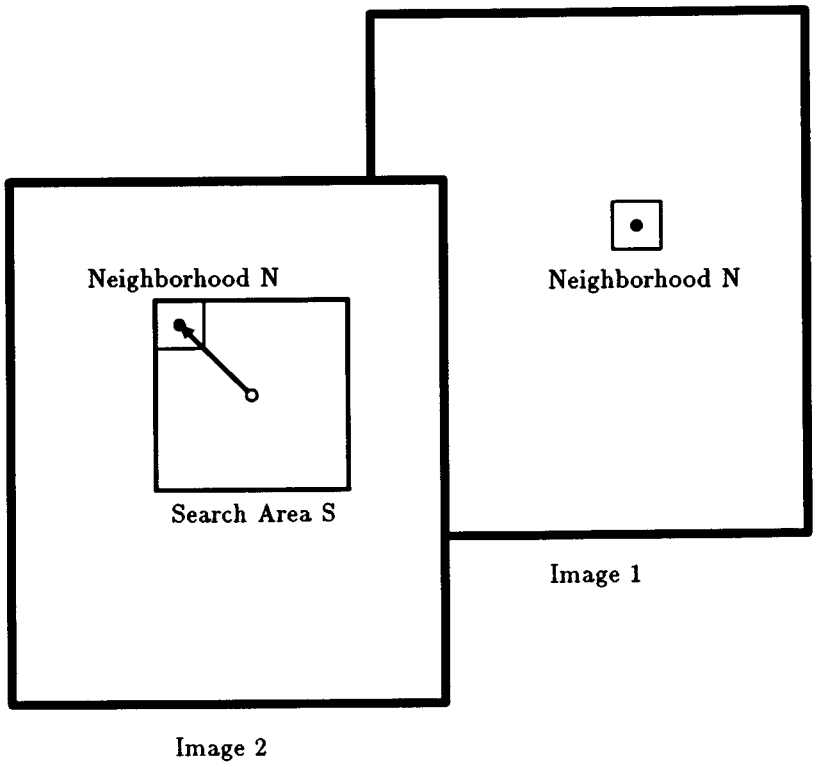

Figure 8.1: Computation of SSD optical flow

- The brightness in an area surrounding the point remains approximately unchanged by the motion.

The first assumption leads to the concept of a search area which is an area of pixels in image 2 that we will consider as possible correspondences for P. The second assumption introduces the sum of squared differences. As a measure of how well P corresponds to each candidate point P' in image 2 we will use the difference between a neighborhood surrounding P and the corresponding neighborhood around P'. The measure is computed as the sum of the squared differences of corresponding pixel values over the entire neighborhood. More formally, if we let $E_1(i,j)$ denote the brightness value at location $P = (i,j)$ in image 1 and $E_2(i,j)$ a brightness value in image 2 then for every $P = (i,j)$ in image 1 we seek a $P' = (i+v, j+u)$ in image 2 such that

$$\min_{u,v \epsilon S} SSD(u,v) = \min_{u,v \epsilon S} \sum_{k,l \epsilon N} [E_1(i+k, j+l) - E_2(i+v+k, j+u+l)]^2. \quad (8.11)$$

N is called the neighborhood, S is referred to as the search-area. The displacement

(u, v) is called the optical flow vector at point P. [1]

The displacement (u, v) that minimizes (8.11) is determined only to pixel accuracy. It can be improved to sub-pixel accuracy as follows: consider the surface of the sum-of-squared differences $SSD(u, v)$. In searching for the minimum of (8.11), we have calculated samples of this surface at the integral grid point locations within the search area S. Suppose that (u, v) was the integral displacement found to yield the minimal value of the SSD function. The true sub-pixel displacement is the minimum of the continuous SSD surface and is located between (u, v) and the neighboring integral displacements $u - 1, u + 1$ and $v - 1, v + 1$ respectively. We will determine the minimum of the continuous SSD-surface, by fitting a quadratic function to the minimum SSD and its neighbors and then analytically determining its minimum.

We fit a one-dimensional quadratic function to the samples in both the u and v dimensions respectively. [2] An example of an SSD-surface with the neighborhoods indicated as well as the corresponding interpolation in the u direction is shown in figure 8.2.

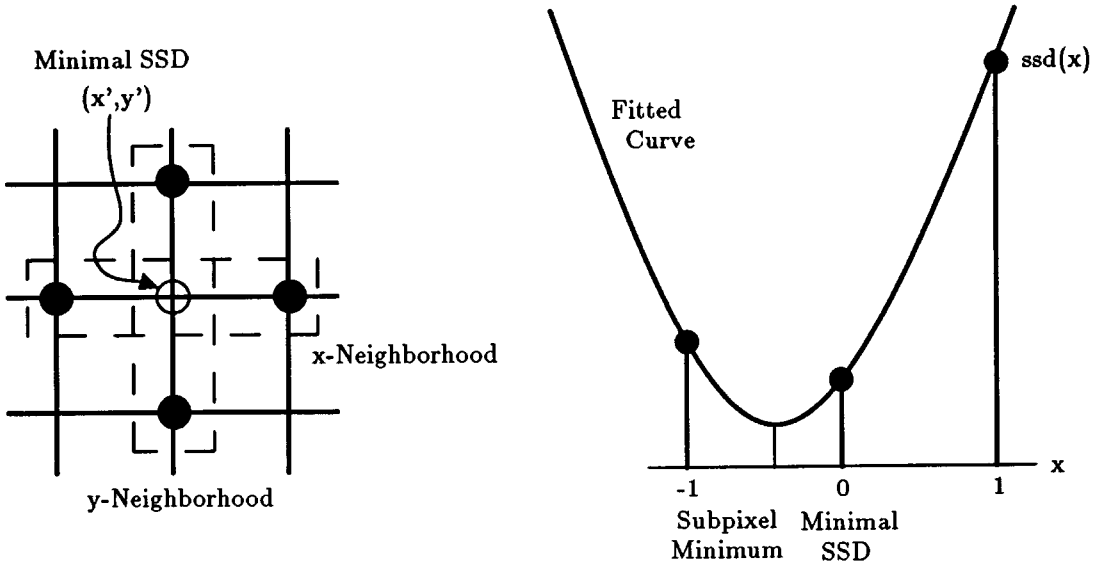

Figure 8.2: Subpixel interpolation of the optical flow

[1] The displacement (u, v) that minimizes (8.11) is in terms of indices in the image array. The optical flow in terms of physical distances is easily obtained by multiplication with the pixel distance $(\Delta x, \Delta y)$.

[2] We could fit a two-dimensional quadratic to the sample points in the immediate neighborhood, but this surface may have a minimum outside the neighborhood and may therefore not always be practical.

More formally we fit the quadratic

$$s(x) = ax^2 + bx + c \tag{8.12}$$

to the three samples s_{-1}, s_0, s_1 where the origin is located at the center of the neighborhood (x stands for either u or v dimension here). The coefficients of the best-fitting surface are

$$a = \frac{1}{2}(s_1 + s_{-1}) - s_0 \tag{8.13}$$

$$b = \frac{1}{2}(s_1 - s_{-1}) \tag{8.14}$$

$$c = s_0. \tag{8.15}$$

Once these coefficients have been determined we find that the subpixel minimum of the surface is located at

$$x' = -b/2a \tag{8.16}$$

and the value there is

$$s(x') = c - b^2/4a. \tag{8.17}$$

From our formulation, x' is a value between -1 and 1 and is simply added to the integral value of the optical flow determined from (8.11).

In their work, Matthies, Szeliski and Kanade [65] give an elegant derivation of the variance in an optical flow estimate obtained by the SSD method. Their method was restricted to translational camera motions perpendicular to the optical axis, so that optical flow would have only a u component. As a consequence, the derivation of the variance was also restricted to this case, but can be extended to the general case presented here as follows.

We model the images $E_1(i,j)$ and $E_2(i,j)$ as originating from a single noise-free image with additive, uncorrelated Gaussian noise of variance σ_E^2, where E_2 is displaced by (u,v) at location (i,j) with respect to E_1

$$E_1(i,j) = E(i,j) + n_1(i,j) \quad \text{and} \quad E_2(i+v, j+u) = E(i,j) + n_2(i,j) \tag{8.18}$$

Now the expression for the SSD error (8.11) becomes [3]

$$\begin{aligned} SSD(\tilde{u},\tilde{v}) = \sum_{k,l \in N} & [E(i+k, j+l) - E(i+\tilde{v}-v+k, j+\tilde{u}-u+l) + \\ & n_1(i+k, j+l) - n_2(i+k, j+l)]^2. \end{aligned} \tag{8.19}$$

By Taylor series expansion of E around (u,v) and retaining the linear terms, we have

$$SSD(\tilde{u},\tilde{v}) = a(\tilde{u}-u)^2 + b(\tilde{u}-u)(\tilde{v}-v) + c(\tilde{v}-v)^2 + d(\tilde{u}-u) + e(\tilde{v}-v) + f \tag{8.20}$$

[3] As is pointed out in [65], the last term is actually $n_2(i+\tilde{v}-v+k, j+\tilde{u}-u+l)$ but can safely be approximated as is done here.

where

$$a = \sum_{k,l \in N} E_x^2(i+k, j+l) \tag{8.21}$$

$$b = 2 \sum_{k,l \in N} E_x(i+k, j+l) E_y(i+k, j+l) \tag{8.22}$$

$$c = \sum_{k,l \in N} E_y^2(i+k, j+l) \tag{8.23}$$

$$d = 2 \sum_{k,l \in N} E_x(i+k, j+l)(n_1(i+k, j+l) - n_2(i+k, j+l)) \tag{8.24}$$

$$e = 2 \sum_{k,l \in N} E_y(i+k, j+l)(n_1(i+k, j+l) - n_2(i+k, j+l)) \tag{8.25}$$

$$f = \sum_{k,l \in N} (n_1(i+k, j+l) - n_2(i+k, j+l))^2. \tag{8.26}$$

E_x and E_y denote a suitable discrete approximation to the partial derivatives of E in the x and y directions.

The minimum of the quadratic error function (8.20) is located at

$$\begin{bmatrix} \tilde{u} - u \\ \tilde{v} - v \end{bmatrix} = \frac{1}{4ac - b^2} \begin{bmatrix} be - 2cd \\ bd - 2ae \end{bmatrix} \tag{8.27}$$

For the computation of the covariance matrix of this estimate we note that only the quantities d and e contain the noise processes and are stochastic while a, b and c are deterministic. From (8.24), (8.25) we compute the covariances

$$\sigma_d^2 = 8\sigma_E^2 a \tag{8.28}$$

$$\sigma_e^2 = 8\sigma_E^2 c \tag{8.29}$$

$$cov(d, e) = 4\sigma_E^2 b \tag{8.30}$$

and from there the covariance matrix of an optical flow vector

$$\mathbf{R} = \begin{bmatrix} p & r \\ r & q \end{bmatrix} = \frac{4\sigma_E^2}{4ac - b^2)^2} \begin{bmatrix} 2c(4ac - b^2) & b^3 \\ b^3 & 2a(4ac - b^2) \end{bmatrix} \tag{8.31}$$

is readily obtained. This covariance matrix is used in the construction (8.8) of the update stage for the temporal surface reconstruction from optical flow.

Note that the derivation of the measurement covariance described here is specific to the SSD optical flow estimation algorithm. However, the procedure of propagating the noise in the brightness measurements through the equations of the optical flow algorithm can be applied to other methods as well.

8.3 Experimental Evaluation

In this section I will present the results of experiments with the temporal surface reconstruction algorithm using optical flow. In both cases, a CCD camera with a focal length of 10 mm translated relative to the scene.

8.3.1 Bottle Experiment

In the first experiment (first presented in [37]) a the camera translates vertically over a scene consisting of a small spray bottle on a table before a flat background. The bottle was 730 mm away from the camera, the background was 1000 mm away. The camera translated 5 mm between each of the 7 frames in the sequence.

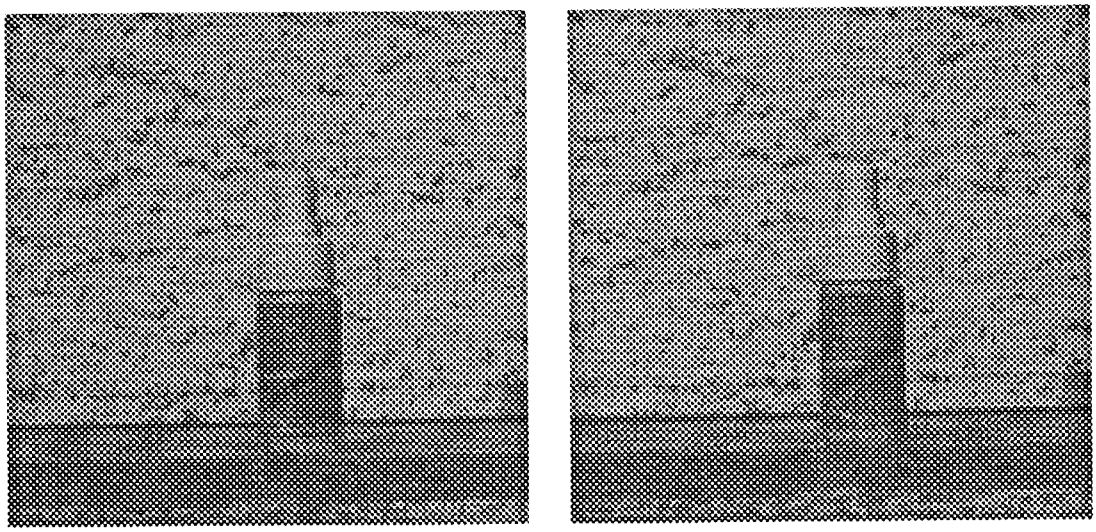

Figure 8.3: The first two images the bottle sequence.

Figure 8.3 shows the first two images from the bottle sequence. Figure 8.4 shows the optical flow fields computed from the first two image pairs of the bottle sequence using the matching optical flow algorithm described above in section 8.2. Variance data was also computed as described in that section. From the optical flow, we can see the translational motion of the camera in a vertical direction.

This data is used in the simplified reconstruction scheme (8.9) in which non-correlation of pixels is assumed as a prior model and a separate binomial filtering step is performed between update and prediction stage of the filter to enforce smoothness. The initial depth map is flat with a value of 900 mm. The reconstructed structure of the scene after every iteration of the algorithm is shown as a wire frame in figure 8.5 from left to right and top to bottom.

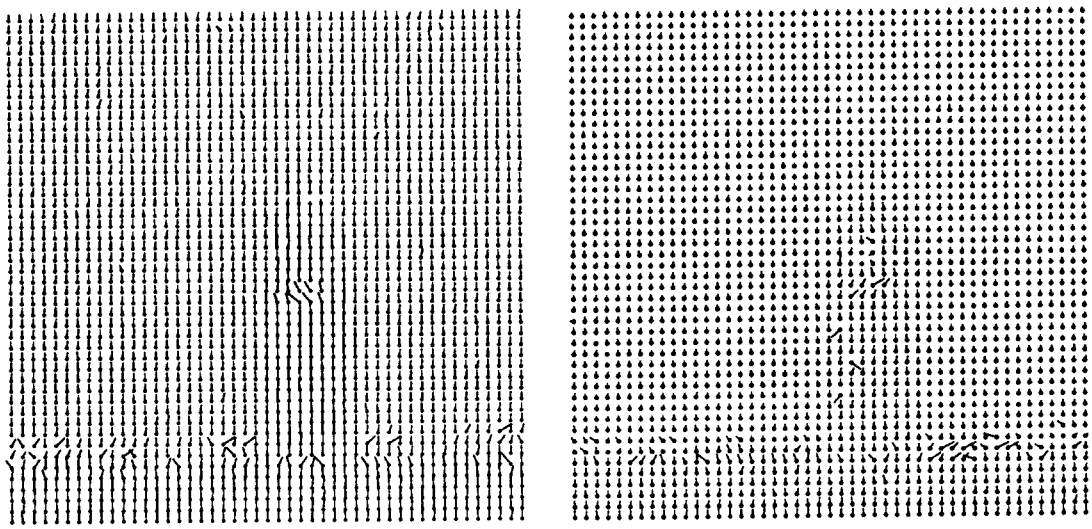

Figure 8.4: The first two optical flow fields from the bottle sequence.

The run-time of the filtering algorithm was approximately 1 minute per frame for the 300 by 300 image on a Sun 3/60 if the optical flow has been precomputed.

8.3.2 Pepsi Experiment

In a second experiment, a soda can was placed on a table at 570 mm, before a background parallel to the image plane at 1240 mm. The sequence of frames taken by the camera is shown in figure 8.6. The camera translated $\mathbf{t} = [1.5, 0, 0]$ mm between frames.

From this image sequence a sequence of optical flow fields was precomputed using the matching algorithm described previously in section 8.2. The first three optical flow fields are shown in figure 8.7 and the translational motion is evident to the human observer. Note that the optical flow fields are rather noisy, in particular in the image regions of fairly uniform brightness, as no smoothness constraint has been imposed.

The disparity map state vector was initialized to a value of $1/\hat{Z}_0 = 1/1000$ everywhere. The Kalman filter temporal surface reconstruction scheme was applied to the optical flow data using the thin plate prior model of surface smoothness. Fourty Gauss-Seidel iterations were used at each time step. With these parameters, each temporal iteration takes about 30 seconds on a Sun Sparcstation I if optical flow has been precomputed for the 200 by 200 images. Figure 8.8 shows a perspective wire-frame rendering of the three-dimensional structure recovered using this algorithm after each iteration. A closeup of the final result is shown in figure 8.9. It is note-

worthy, that considerable error remains even after 10 iterations and it is particularly prominent where the input optical flow contained errors. This is simply a reflection of the fact that image areas of nearly uniform brightness allow little information about structure to be estimated from motion.

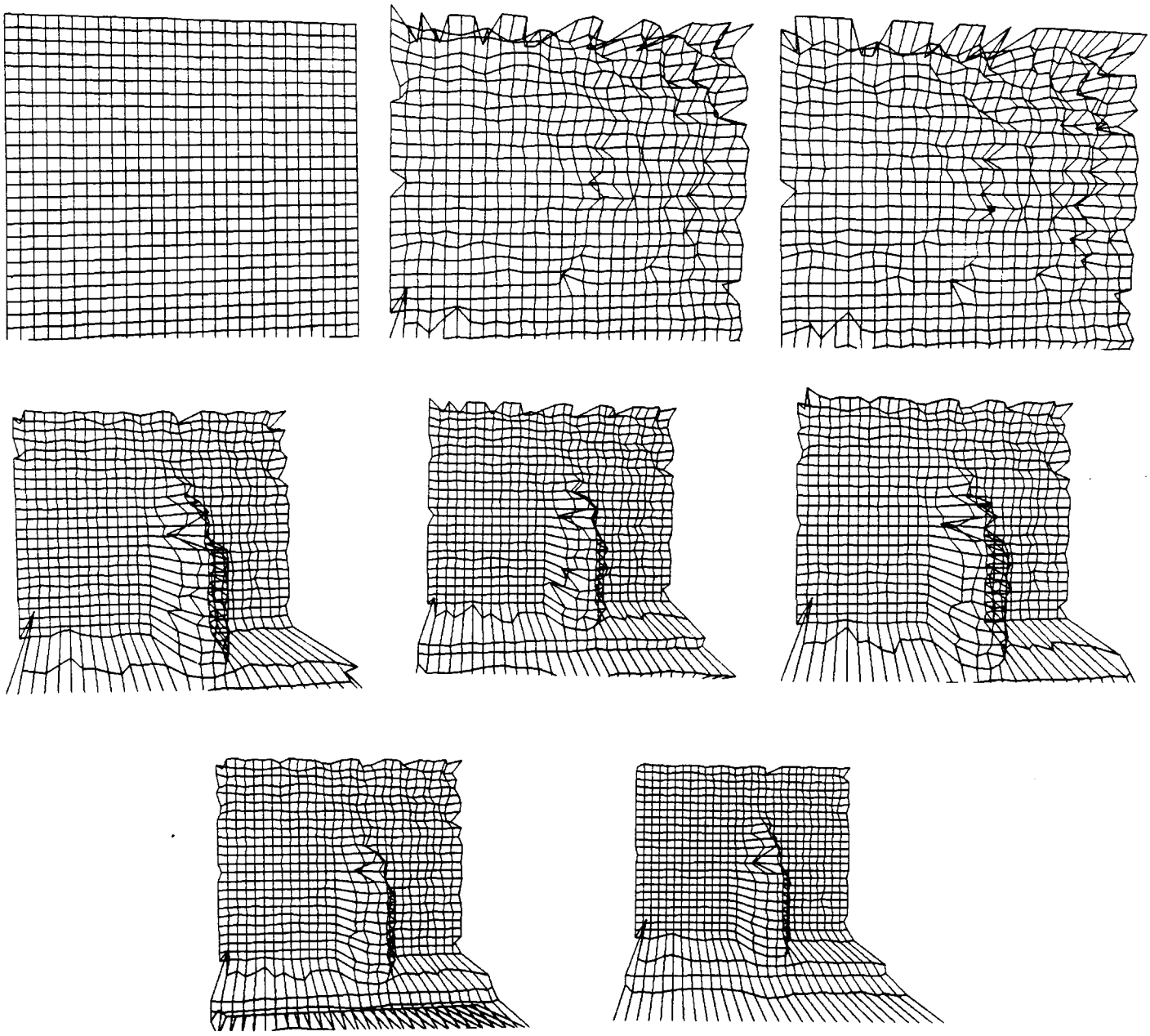

Figure 8.5: The structure recovered from the bottle sequence after each iteration of the Kalman filter depth from motion algorithm from left to right and top to bottom.

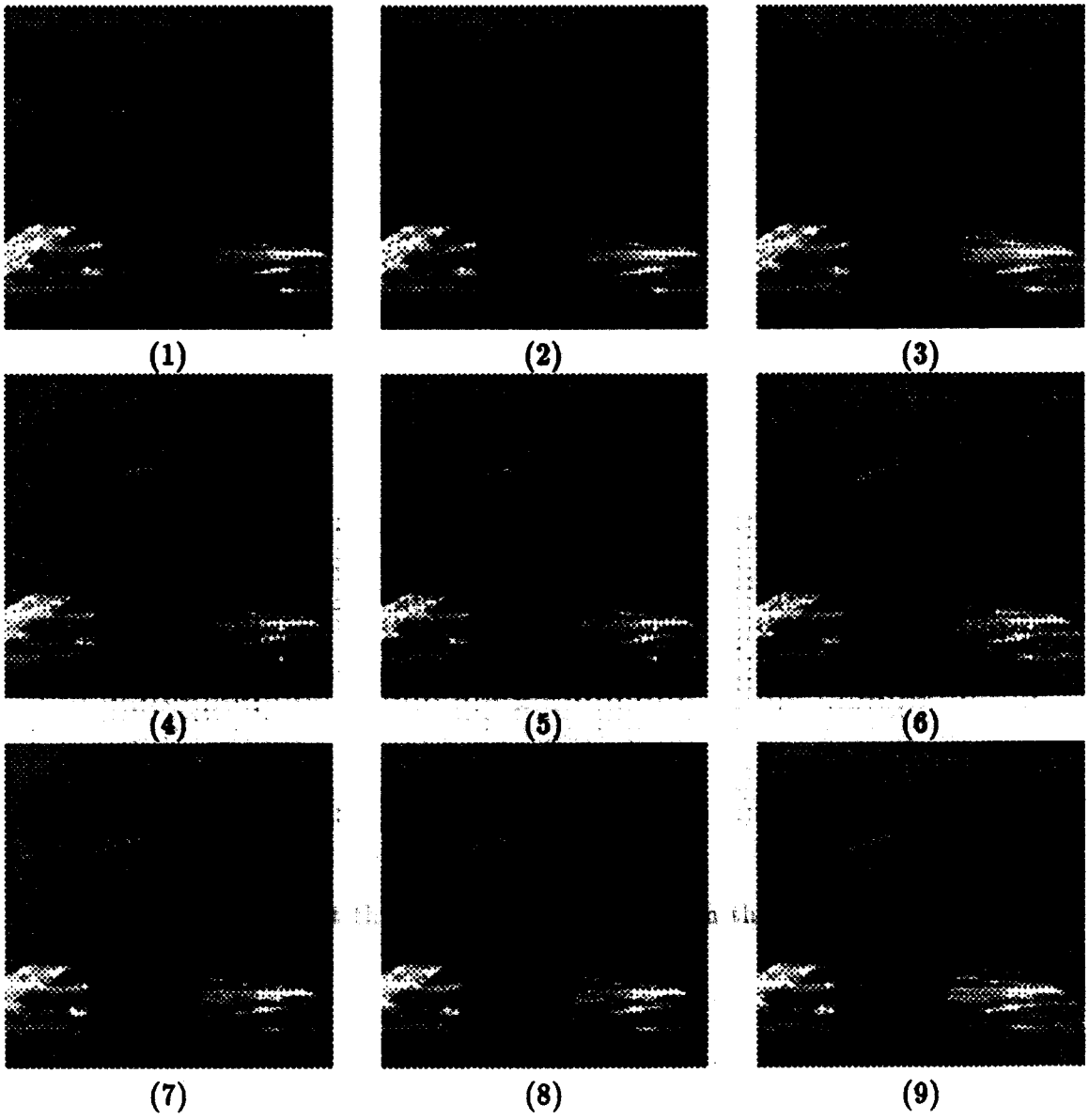

Figure 8.6: The first 9 images from the pepsi sequence from left to right and top to bottom.

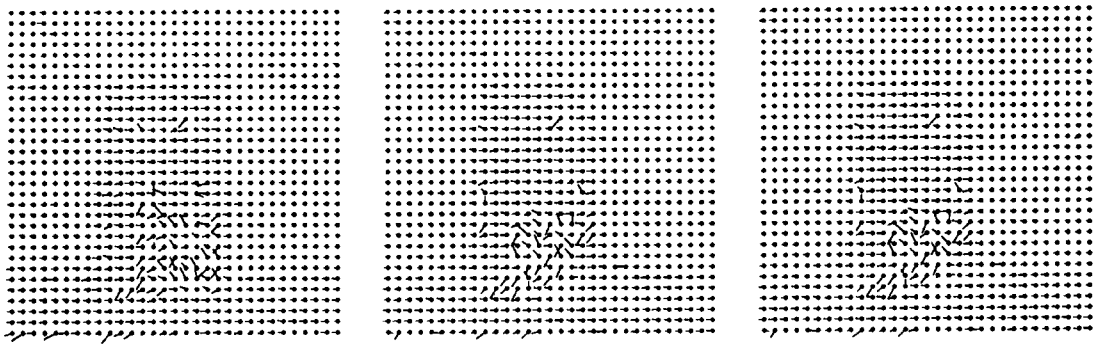

Figure 8.7: The first three optical flow fields from the pepsi experiment.

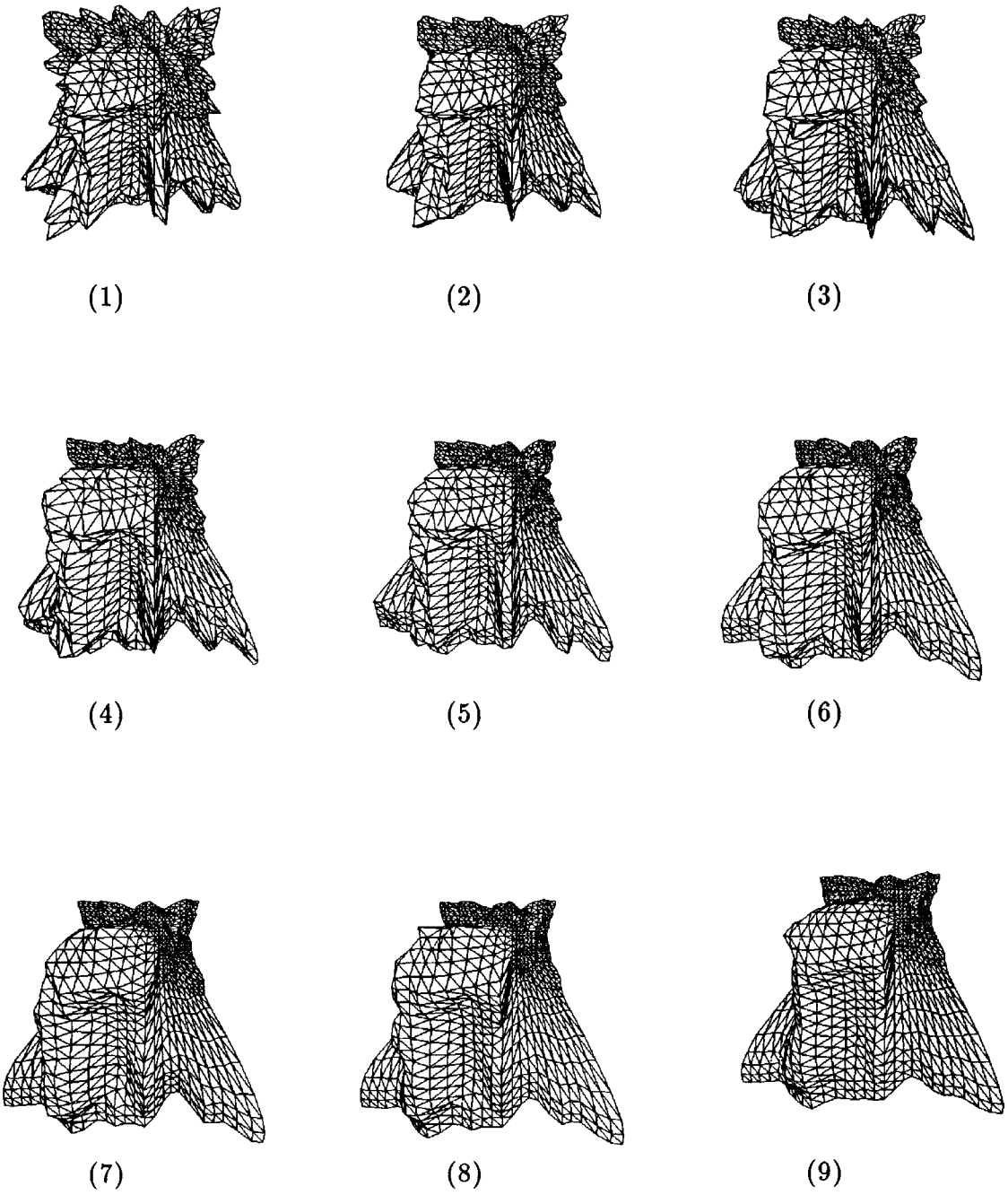

Figure 8.8: Wire frame renderings of the structure recovered after each of the first 9 iterations of the temporal surface reconstruction algorithm from the optical flow of the pepsi sequence.

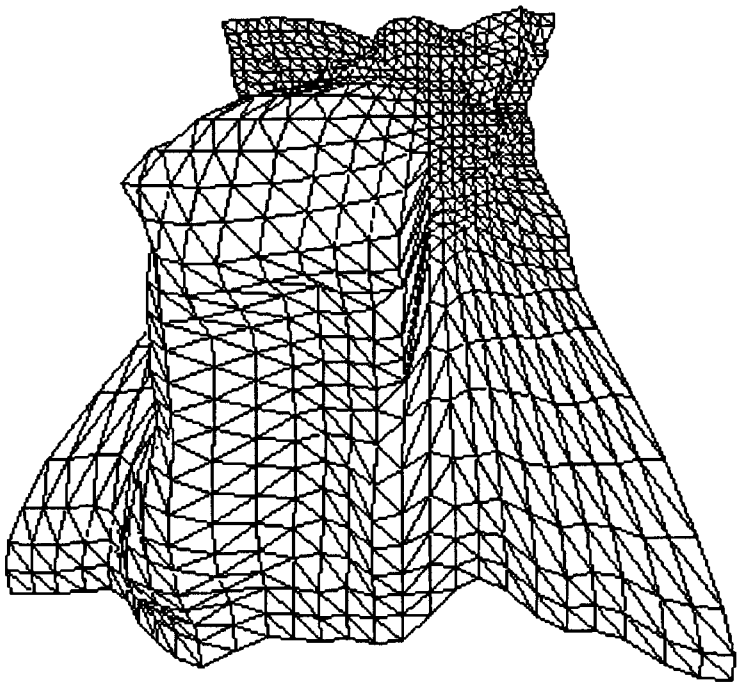

Figure 8.9: A closer look at the structure recovered after the 9th iteration of the temporal structure estimator using optical flow on the pepsi sequence

Filter Update: Depth from Shading

Chapter 9

In this chapter we will explore the application of the temporal surface reconstruction algorithm to the problem of estimating depth from shading. A corresponding instantaneous algorithm that accomplishes this task when given a single image was described in section 2.2.9. Since only the update stage of the recursive estimator is dependent on the visual mechanism, the focus in this chapter is exclusively on that part of the algorithm.

9.1 The Update Algorithm

In shape from shading [46] it is assumed that the image brightness E observed at a location on the surface is a known function of the surface normal $[-p, -q, 1]$

$$E = R(p, q). \tag{9.1}$$

where R is called the reflectance function. Reflectance functions have been determined for a number of surfaces such as the lunar surface and Lambertian surfaces. By noting that $p = Z_x$ and $q = Z_y$ (the partial derivatives of depth) and choosing a discrete approximation we obtain

$$E_{ij} = R(\frac{Z_{i,j+1} - Z_{i,j-1}}{2\Delta x}, \frac{Z_{i+1,j} - Z_{i-1,j}}{2\Delta y}) \tag{9.2}$$

This formula can be used as the nonlinear measurement equation $\mathbf{y} = \mathbf{f}(\mathbf{x})$ (4.8) if we construct the state $\mathbf{x}$ by concatenating the rows of the depth map

$$x_{im+j} = Z_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \tag{9.3}$$

and the measurement vector $\mathbf{y}$ by concatenating the rows of the image brightness field

$$y_{im+j} = E_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1. \tag{9.4}$$

The matrix $\mathbf{C} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ is found to be

$$C_{im+j,k} = \begin{cases} R_p/2\Delta x & k = im + j + 1 \\ -R_p/2\Delta x & k = im + j - 1 \\ R_q/2\Delta y & k = (i+1)m + j \\ -R_q/2\Delta y & k = (i-1)m + j \\ 0 & \text{otherwise} \end{cases} \tag{9.5}$$

where R_p and R_q denote the partials of R with respect to p and q.

Finally, the covariance of the measurement is

$$\mathbf{R} = \sigma_E^2 \mathbf{I} \tag{9.6}$$

if we assume brightness measurements to be uncorrelated at different pixels and identically distributed with variance σ_E^2.

Special thought must be given to the filter initialization, as the measurement 9.2 and therefore the filter update are a function of derivatives of the depth Z. Initializing the state vector to a constant value as suggested in chapter 6 is useless in this case, as it constitutes a local minimum in the solution space in which the iterative update scheme (5.8) becomes stuck. The solution is to initialize the depth to random numbers which will guarantee that the derivatives p, q will be non-zero.

As with all shape from shading algorithms, convergence and local minima are a particular difficulty. The implementation must take into account integrability of the derivatives and boundary conditions among others. In dealing with these problems, I have followed the suggestions of Horn [45]; the paper contains a careful discussion of implementation details for shape from shading algorithms.

A noteworthy point is the fact that conventional shape from shading is formulated for orthographic projections while the prediction stage of the temporal reconstruction scheme assumes perspective projection. It is possible to reformulate the shape from shading algorithms for perspective projection at the cost of increased mathematical complexity. In my implementation, I instead reformulated the prediction stage of the reconstruction scheme to work with orthographic projection.

9.2 Experimental Evaluation

The implementation of this Kalman filter for structure from shading (first presented in Heel [41]) uses the thin plate (6.1) as the prior model.

9.2.1 Sphere Experiment

For this experiment a synthetic image of a semi-sphere on a planar background was created using the Lambertian shading model. Brightness values are in the range $[0, 1]$

and Gaussian noise of variance 0.05 was added. The camera translated uniformly in both in both x and y directions.

Figure 9.1 shows the first 8 frames from the sequence of sphere images. Note that these images are only 50 by 50 pixels in size and have been enlarged for better inspection. Note also the visible effect of noise in the images.

The temporal surface reconstruction scheme was applied to the sphere sequence with natural boundary conditions imposed. Fifty Gauss-Seidel iterations were used per frame. Figure 9.2 shows wire frame renderings of the structure obtained after each iteration of the temporal reconstruction scheme. Figure 9.3 compares the structure obtained after the eighth iteration of the algorithm with the ground truth.

Since a synthetic sequence was used, the ground truth structure was known and can be compared quantitatively to the estimate. Figure 9.4 shows the development of the root mean squared error of the estimate with respect to the ground truth as a function of the frame number.

The convergence behavior of the temporal depth from shading algorithm is of particular interest. I compared it with Horn's [45] Height and Gradient from Shading on a single frame. Horn's algorithm will typically require about 1000 iterations and it may diverge for some illumination directions of the sphere. The temporal algorithm will diverge in the same cases, however when it converges, it required only about 50 iterations per frame. This is of course mainly due to the fact, that initialization of the iterative update procedure for each frame uses the predicted data from the previous temporal iteration while the single-frame algorithm "starts from scratch".

When noise is added, as in the above experiment, the temporal algorithm converges even when the single-frame algorithm does not and the result was in all cases closer to the ground truth than the single-frame algorithm. This is to be expected, as the temporal algorithm can use redundant measurements to reduce the effect of noisy measurements while the single-frame algorithm can only impose a surface smoothness constraint to eliminate noise.

9.2.2 Crater Experiment

In this experiment, images of Mars taken by the Viking orbiter were analyzed. Figure 9.5 shows a sequence of images of a crater on the surface that were taken from a larger image to simulate a translatory motion of the orbiter (recall that we are using orthographic projection). A vertical translation corresponding to about one pixel per frame was used.

The temporal surface reconstruction algorithm was applied to this sequence using fixed boundary constraints (i.e. the boundary was constrained to a constant) and 40 Gauss-Seidel iterations per frame. The light source direction was estimated at $(p_s, q_s) = (0, 10)$ and this estimate was verified by evaluating the resulting shading. The wire frame rendering of the recovered structure after each time step is shown in

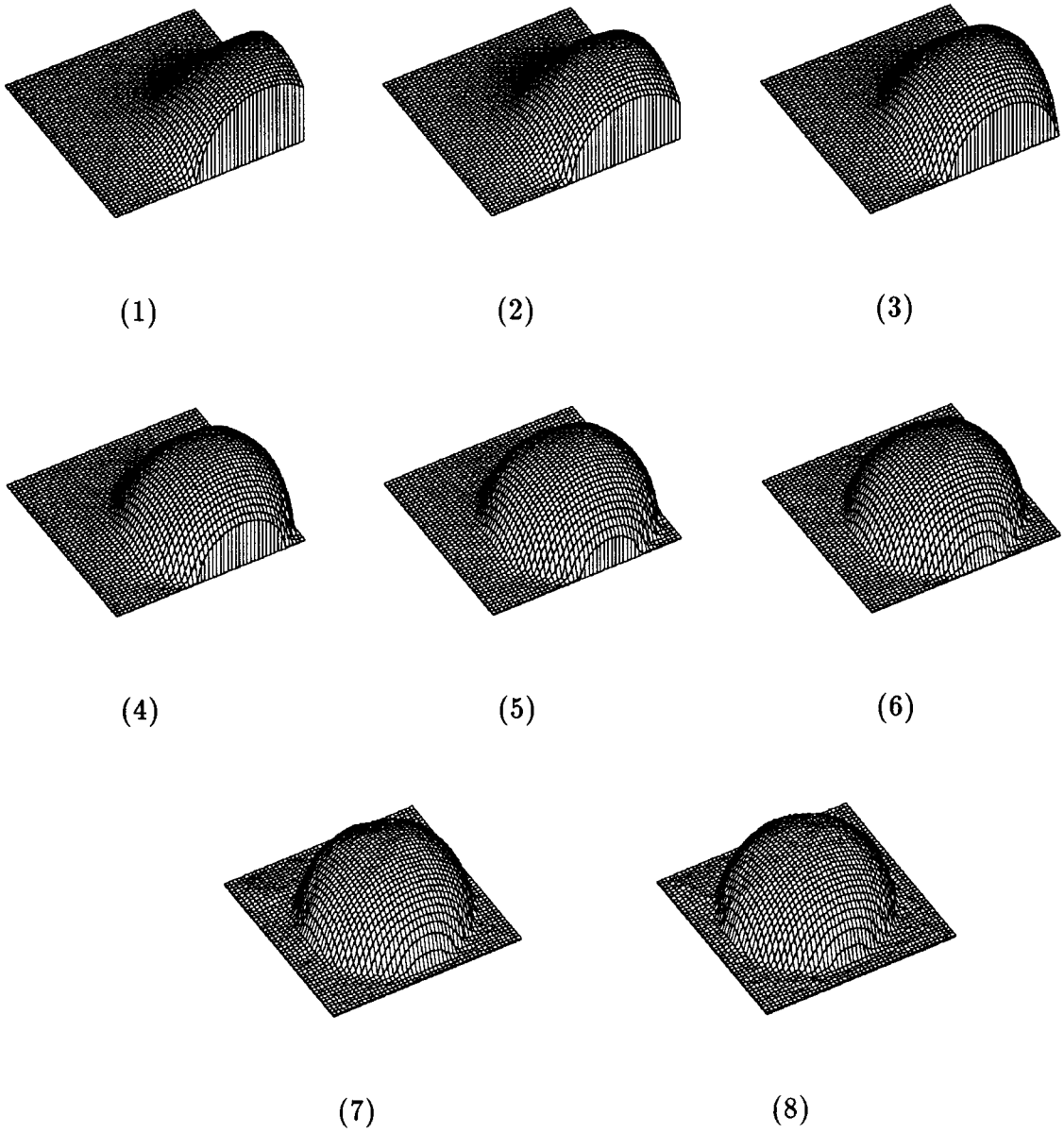

Figure 9.2: Wireframe renderings of the first 8 structure estimates for the sphere sequence.

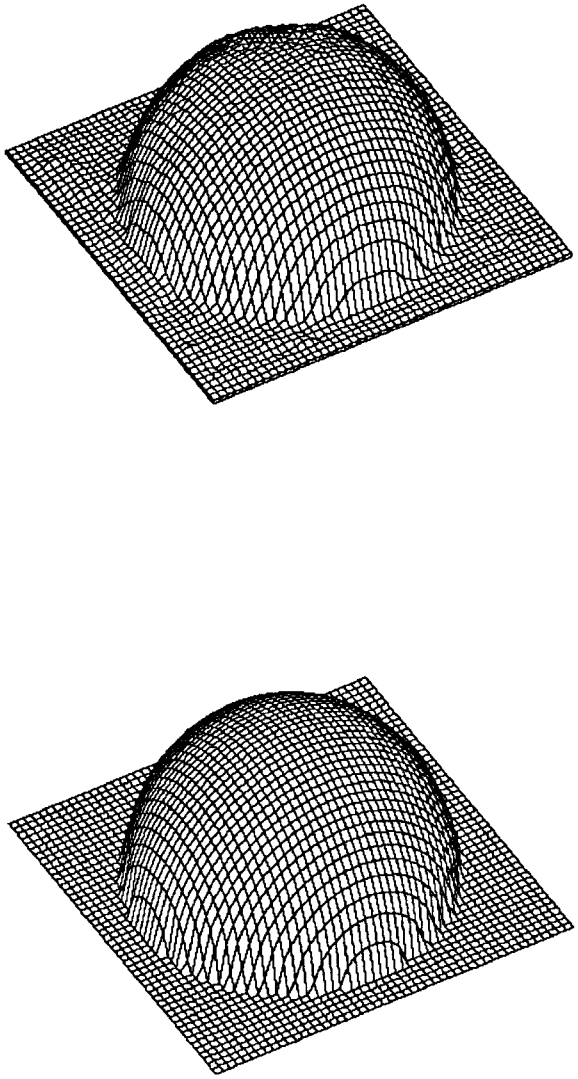

Figure 9.3: Wireframe renderings of the final structure estimate from the sphere sequence and the ground truth structure used to generate the corresponding synthetic image.

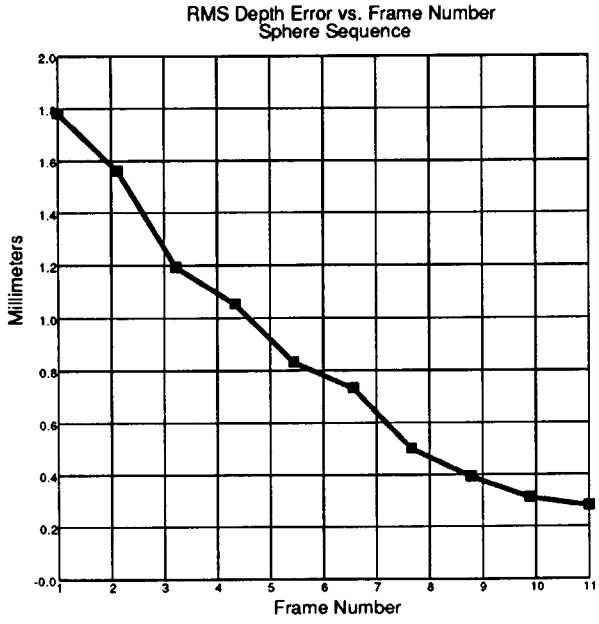

Figure 9.4: Root mean squared error of depth over frame number.

figure 9.6 from a viewpoint located just above the surface. A closeup of the result after nine iterations is shown in figure 9.7.

Since the ground truth structure is not known in this case, a quantitative evaluation of the result of the estimation is difficult. A method commonly used in shape from shading in this case is to shade the reconstructed surface using the same light source direction that was used in the reconstruction and to compare the resulting image with the input image. As Horn points out in [45], it is also imperative to additionally shade the surface using other light source directions. Figure 9.8 shows the result of shading the structure estimate from the ninth iteration using both the light source direction $(0, 10)$ used for reconstruction and the light source direction $(0, -10)$ that illuminates the surface from "above". These images can be compared with the last one in the sequence of figure 9.1

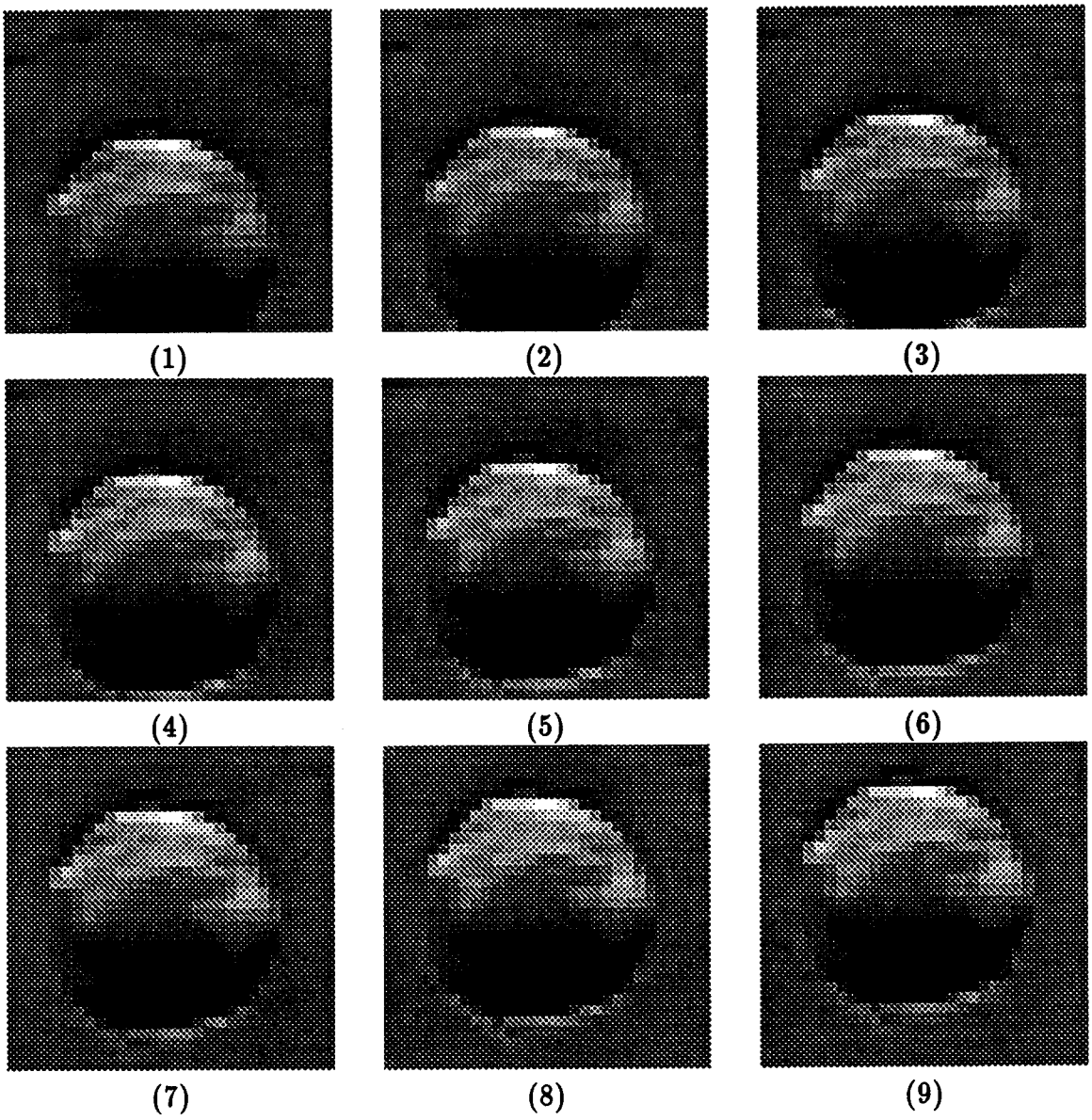

Figure 9.5: The first 9 images from the Mars crater sequence.

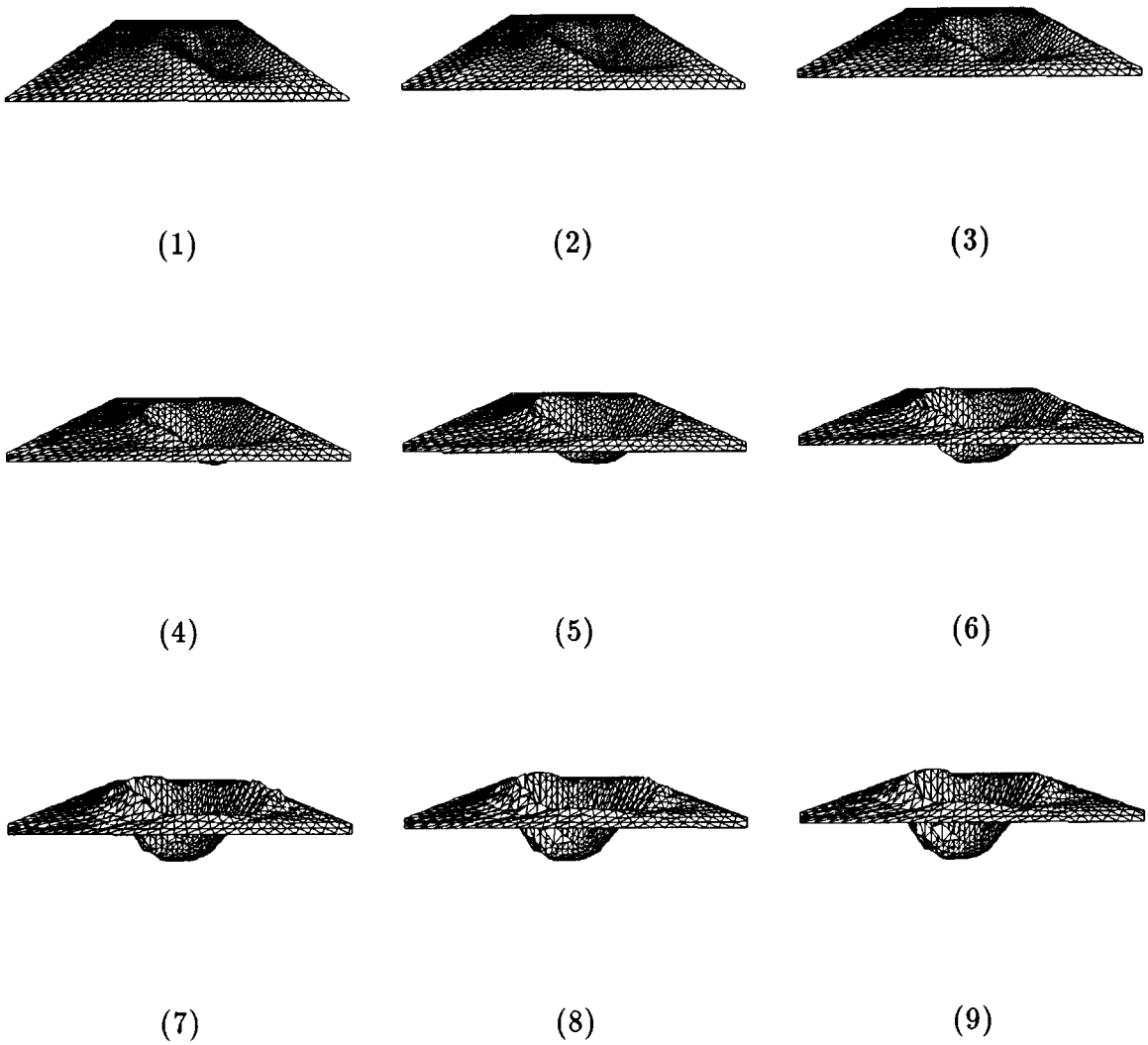

Figure 9.6: Wireframe rendering of the structure estimates from the Mars crater sequence after each iteration.

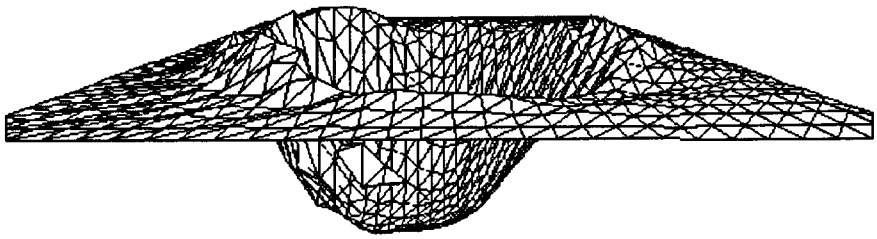

Figure 9.7: A closer look at the structure recovered after the 9th iteration of the temporal structure estimator using shading on the the crater sequence

Filter Update: Direct Depth from Motion

Chapter 10

In this chapter we will explore the application of the temporal surface reconstruction algorithm to the problem of estimating depth from motion without the use of optical flow. A corresponding instantaneous algorithm that accomplishes this task when given a single measurement of the optical flow was described in section 2.2.8. The focus in this chapter is exclusively on the update stage of the recursive estimator, as it is the only part that is dependent on the visual mechanism.

10.1 The Update Algorithm

Based on the brightness constancy assumption

$$\frac{dE}{dt} = 0 \tag{10.1}$$

Horn, Negahdaripour and Weldon [75], [48] derived a relationship that links image brightness E directly to motion $\mathbf{t}$, $\boldsymbol{\omega}$ and disparity d and obviates the need for expensive computation of the optical flow:

$$(\mathbf{s} \cdot \mathbf{t})d + \mathbf{v} \cdot \boldsymbol{\omega} + E_t = 0 \tag{10.2}$$

where $\mathbf{s} = [-E_x, -E_y, xE_x + yE_y]^T$ and $\mathbf{v} = [(1+y^2)E_y + xyE_x, -(1+x^2)E_x - xyE_y, yE_x - xE_y]^T$. Given at least two frames E and the motion $\mathbf{t}$, $\boldsymbol{\omega}$ we can compute the partial dertivatives E_x, E_y, E_t and hence the vectors $\mathbf{s}$, $\mathbf{v}$. Then d is easily computed.

We can construct a state vector $\mathbf{x}$ by concatenating all the rows of the disparity map

$$x_{im+j} = d_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \tag{10.3}$$

The measurement vector $\mathbf{y}$ is comprised of the brightness values E_{ijk} from two sequential images $k = 0, 1$, so that spatial and temporal brightness derivatives can be

computed:

$$y_{kmn+im+j} = E_{ijk} \qquad i = 0,\ldots,n-1; j = 0,\ldots,m-1; k = 0,1 \tag{10.4}$$

For the sake of simplicity we will assume a simple discrete derivative operator that can easily be replaced with a more elaborate one:

$$E_x(i,j) = \frac{E_{i,j+1,0} - E_{i,j-1,0}}{2\Delta x} \tag{10.5}$$

$$E_y(i,j) = \frac{E_{i+1,j,0} - E_{i-1,j,0}}{2\Delta y} \tag{10.6}$$

$$E_t(i,j) = \frac{E_{i,j,1} - E_{i,j,0}}{\Delta t} \tag{10.7}$$

where Δx, Δy denote the pixel distances and Δt is the time between frames.

With these choices we see that the measurement equation (10.2) is of the implicit type (4.12) $\mathbf{g}(\mathbf{x},\mathbf{y}) = \mathbf{0}$. The matrix $\mathbf{C} = \partial\mathbf{g}/\partial\mathbf{x}$ is diagonal with

$$C_{im+j,im+j} = -E_x(i,j)U - E_y(i,j)V + (xE_x(i,j) + yE_y(i,j))W. \tag{10.8}$$

The matrix $\mathbf{D} = \partial\mathbf{g}/\partial\mathbf{y}$ is sparse and banded with 6 bands:

$$D_{im+j,l} = \begin{cases} -\frac{1}{2\Delta x}((-U+xW)d_{ij} + Axy - B(1+x^2) + Cy) & l = im+j-1 \\ \frac{1}{2\Delta x}((-U+xW)d_{ij} + Axy - B(1+x^2) + Cy) & l = im+j+1 \\ -\frac{1}{2\Delta y}((-V+xW)d_{ij} + A(1+y^2) - Bxy - Cx) & l = (i-1)m+j \\ \frac{1}{2\Delta y}((-V+xW)d_{ij} + A(1+y^2) - Bxy - Cx) & l = (i+1)m+j \\ -\frac{1}{\Delta t} & l = im+j \\ \frac{1}{\Delta t} & l = im+j+nm \\ 0 & \text{otherwise} \end{cases} \tag{10.9}$$

If we assume our measurement values E to be identically distributed Gaussian random variables with variance σ_E^2 then the measurement covariance is given by $\mathbf{R} = \sigma_E^2\mathbf{I}$.

The above choices for the state $\mathbf{x}$, the measurement $\mathbf{y}$, the measurement matrices $\mathbf{C}$ and $\mathbf{D}$ as well as the measurement covariance $\mathbf{R}$ completely determine the update stage of the implicit Kalman filter and the update can be preformed by plugging these values into the equations (4.16), (4.17) (4.18). [1]

[1] Note that the update equations are slightly modified in the case of the implicit filter as described in section 4.4.

10.2 Alternative Formulation of the Update Algorithm

In designing the Kalman filter the choice of the depth as the state and the brightness as the measurement in the above formulation, leads to the implicit nonlinear filter. Other choices are possible and may lead to other properties of the estimation algorithm. One such alternative is presented here.

If we solve the brightness constancy constraint (10.2) for the depth $Z = 1/d$

$$Z = -\frac{\mathbf{s} \cdot \mathbf{t}}{\mathbf{v} \cdot \boldsymbol{\omega} + E_t} \tag{10.10}$$

we can precompute an estimate of the depth map values. Now we designate the depth values Z_{ij} to be not only the state $\mathbf{x}$

$$x_{im+j} = Z_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1 \tag{10.11}$$

but also the measurement $\mathbf{y}$

$$y_{im+j} = Z_{ij} \qquad i = 0, \ldots, n-1; j = 0, \ldots, m-1. \tag{10.12}$$

Now the relationship between state and measurement is given by the measurement matrix $\mathbf{C}$ which is simply the identity matrix. Through this choice of the measurement vector the filter actually becomes linear (4.4)!

The disadvantage of this formulation is that the covariance $\mathbf{R}$ of the measurement is more complex. To obtain an estimate of the variance in a given depth value Z we will propagate the noise in the brightness measurements E through equation (10.10). We assume that the brightness E at every pixel is corrupted by Gaussian noise n of variance σ_E^2 that is identically distributed at every pixel and mutually uncorrelated between pixels.

First we express the depth Z from (10.10) explicitly in terms of the brightness derivatives:

$$Z = \frac{aE_x + bE_y}{E_t + cE_x + dE_y} \tag{10.13}$$

where

$$a = fU - xW \tag{10.14}$$
$$b = fV - yW \tag{10.15}$$
$$c = (xyA - (f^2 + x^2)B)/f + yC \tag{10.16}$$
$$d = ((f^2 + y^2)A - xyB)/f - xC \tag{10.17}$$

Recall that f is the focal length, $\mathbf{t} = [U, V, W]^T$ and $\boldsymbol{\omega} = [A, B, C]^T$.

If the discrete approximations for the derivatives E_x, E_y, E_t suggested by Horn [44] are used, eight brightness values $E_i, i = 1, \ldots 8$ at the corners of a spatio-temporal cube contribute to the value of Z in (10.13)

$$Z = f(E_1, \ldots, E_8) \tag{10.18}$$

Under the assumption that the nonlinear function f can be locally approximated by the first-order terms of its Taylor series, the depth variance is given by

$$\sigma_Z^2 = \sigma_E^2 \sum_{i=1}^{8} (\frac{\partial Z}{\partial E_i})^2. \tag{10.19}$$

If we apply this formula to our expression (10.13) for the depth, the variance is found to be

$$\sigma_Z^2 = \frac{\sigma_E^2}{2} \left((\frac{1}{\Delta x})^2 (\frac{\partial Z}{\partial E_x})^2 + (\frac{1}{\Delta y})^2 (\frac{\partial Z}{\partial E_y})^2 + (\frac{1}{\Delta t})^2 (\frac{\partial Z}{\partial E_t})^2 \right) \tag{10.20}$$

where

$$\frac{\partial Z}{\partial E_x} = \frac{aE_t + (ad - bc)E_y}{(E_t + cE_x + dE_y)^2} \tag{10.21}$$

$$\frac{\partial Z}{\partial E_y} = \frac{bE_t - (ad - bc)E_x}{(E_t + cE_x + dE_y)^2} \tag{10.22}$$

$$\frac{\partial Z}{\partial E_y} = -\frac{aE_x + bE_y}{(E_t + cE_x + dE_y)^2}. \tag{10.23}$$

If we assume that depth map values are uncorrelated, the measurement covariance $\mathbf{R}$ is diagonal

$$R_{im+j,im+j} = \sigma_{ij}^2. \tag{10.24}$$

Actually, with the discrete derivative operator chosen above, the depth value Z_{ij} is correlated with its eight-connected neighbors. The correlation can be computed in the same way as the variance calculation shown here.

10.3 Experimental Evaluation

This section shows the results of experiments with the alternative implementation of the temporal reconstruction scheme for direct structure from motion. The same camera and experimental setup previously described in section 8.3 were used. The pepsi experiment below uses the same sequence of images introduced in that section, so that an immediate comparison of the two structure from motion techniques is possible. The thin plate model of surface smoothness was used with between 20 and 50 Gauss-Seidel iterations per frame.

10.3.1 Wave Experiment

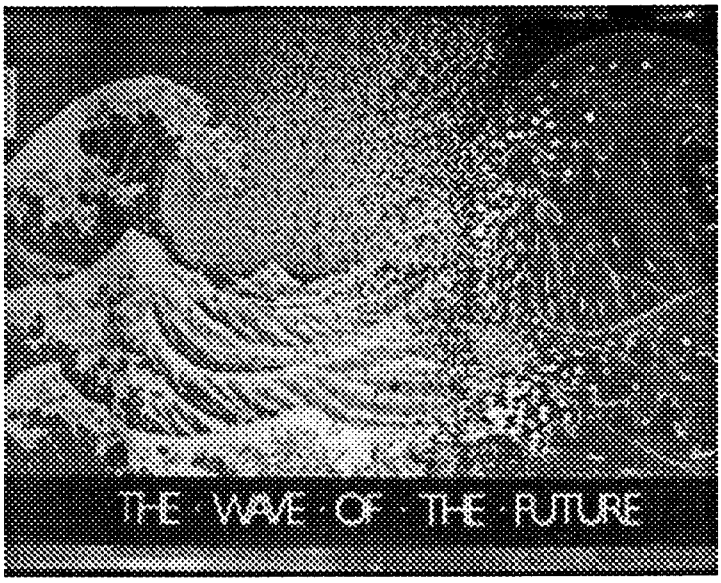

Figure 10.1: The wave experiment scene

This first experiment was designed to evaluate the performance of the temporal reconstruction scheme quantitatively. A simple planar scene was created by mounting a poster (to obtain the desired texture for motion) on a wall parallel to the camera's image plane at a distance of 1000 mm. The camera translated by $\mathbf{t} = [1.5, 0, 3.0]$ mm relative to the surface. The last of a sequence of 10 frames taken by this camera is show in figure 10.1 .

The recovered structure after 1 and 10 iterations of the temporal surface reconstruction algorithm is shown in figures 10.2. Since the thin-plate model of surface smoothness favors fronto-parallel surfaces and would therefore lead to a misrepresentation of the noise reduction effect achieved by temporal integration, it was turned off in this experiment. Figure 10.3 shows the development of the root mean squared error with respect to the ground truth as a function of the frame number.

10.3.2 Pepsi Experiment

This experiment uses identically the same images as the one described in section 8.3. The result of applying the temporal reconstruction algorithm directly to this motion sequence as opposed to computing the optical flow is shown in figure 10.4. A closeup look at the wireframe rendering of the structure obtained after the eighth iteration is shown in figure 10.5. It is noteworthy that each iteration of the the filter takes

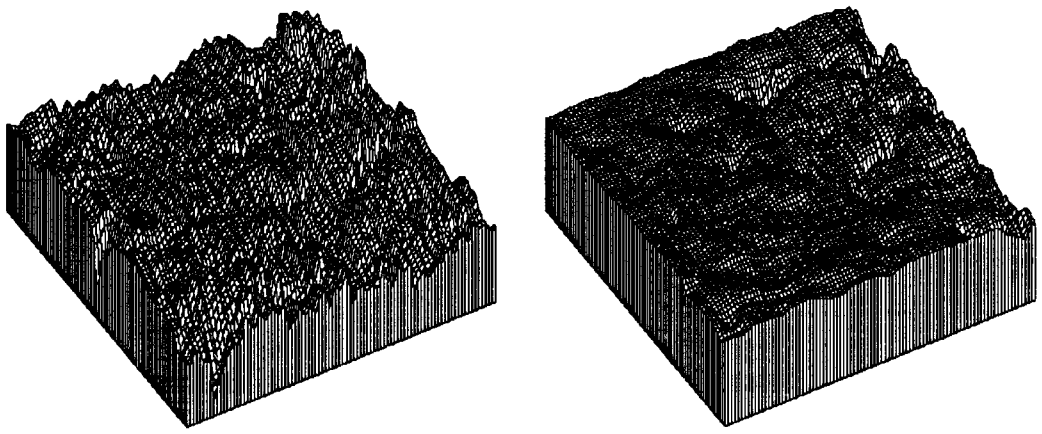

Figure 10.2: Wireframe renderings of the structure from the wave scene after 1 and 10 iterations of the filter.

approximately 30 seconds on the 200 by 200 images. This is the same time required by the optical flow based algorithm with the difference, that in the latter case, the optical flow must also be computed. Depending on the parameters of the optical flow algorithm, this may require up to 20 minutes!

10.3.3 Cup Experiment

This experiment was designed to evaluate the performance of the temporal reconstruction algorithm on a complex scene. A cup, a set of books and a staple-remover were placed on a table before a planar background on which a poster was mounted. A top view of the layout of the experiment is shown in figure 10.6. The camera translated $\mathbf{t} = [2, 0, 4]$ mm between frames to acquire the sequence of images shown in figure 10.7.

Figure 10.8 shows wire-frame renderings of the structure estimates after each one of the temporal iterations. A closeup of the structure after the ninth iteration is shown in figure 10.9. Note that what looks like a woman's face in the image is acutally a poster and appears flat in the depth map while a small spoon in the cup that is barely visible in the image shows up clearly in the structure rendering.

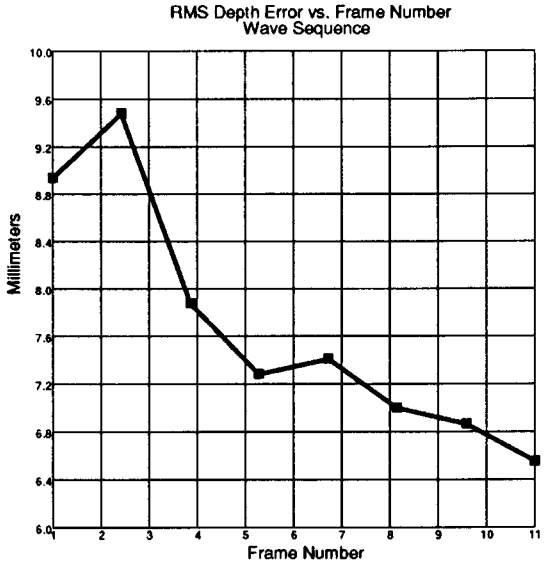

Figure 10.3: Development of root mean squared depth error as a function of the frame number.

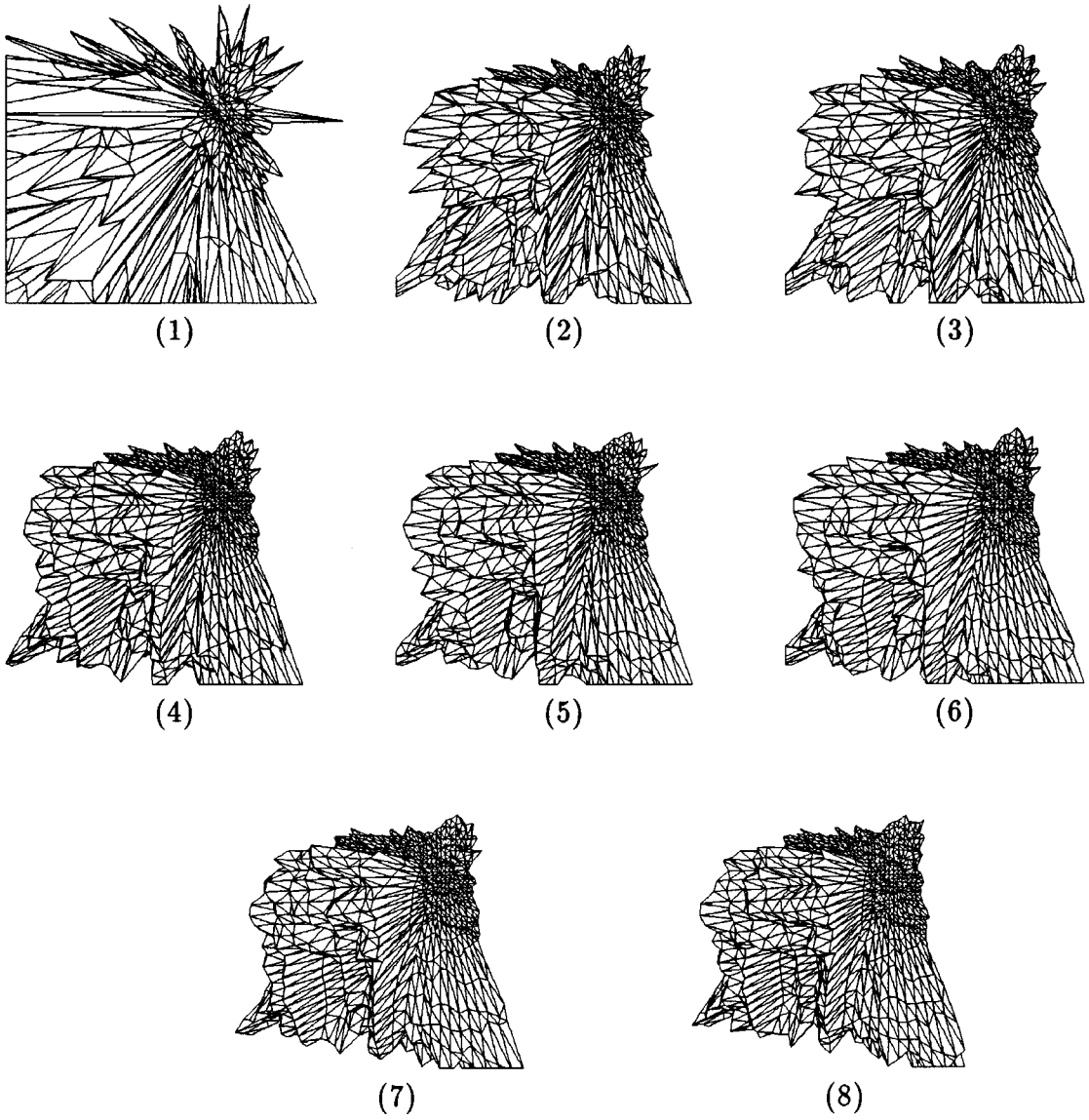

Figure 10.4: Wire frame renderings of the structure recovered after each of the first 8 iterations of the temporal surface reconstruction algorithm from the pepsi sequence.

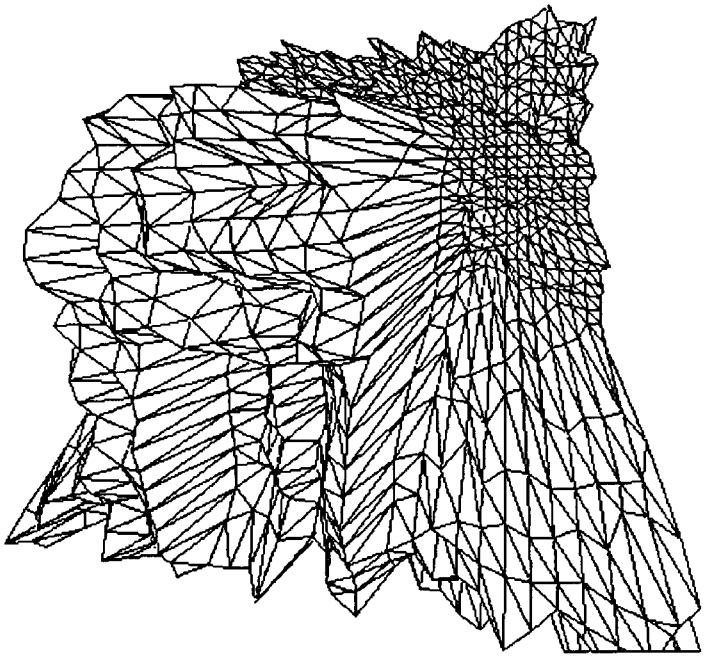

Figure 10.5: A closer look at the structure recovered after the 8th iteration of the temporal structure estimator using direct motion on the the pepsi sequence

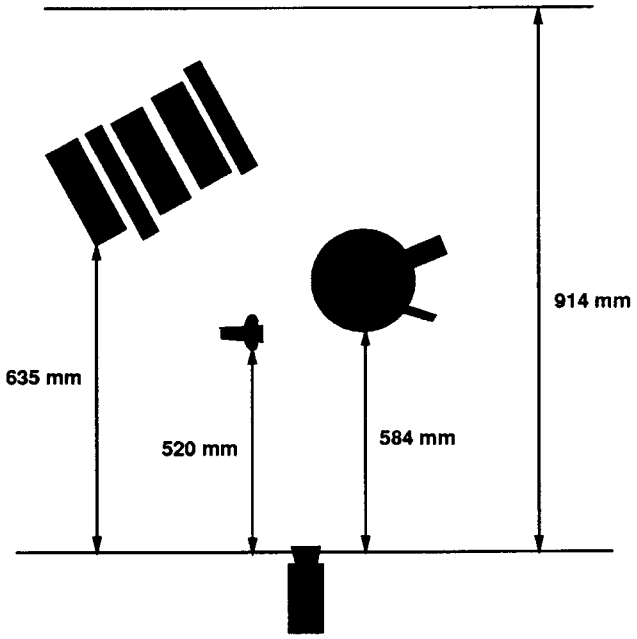

Figure 10.6: A top view of the scene layout for the cup experiment.

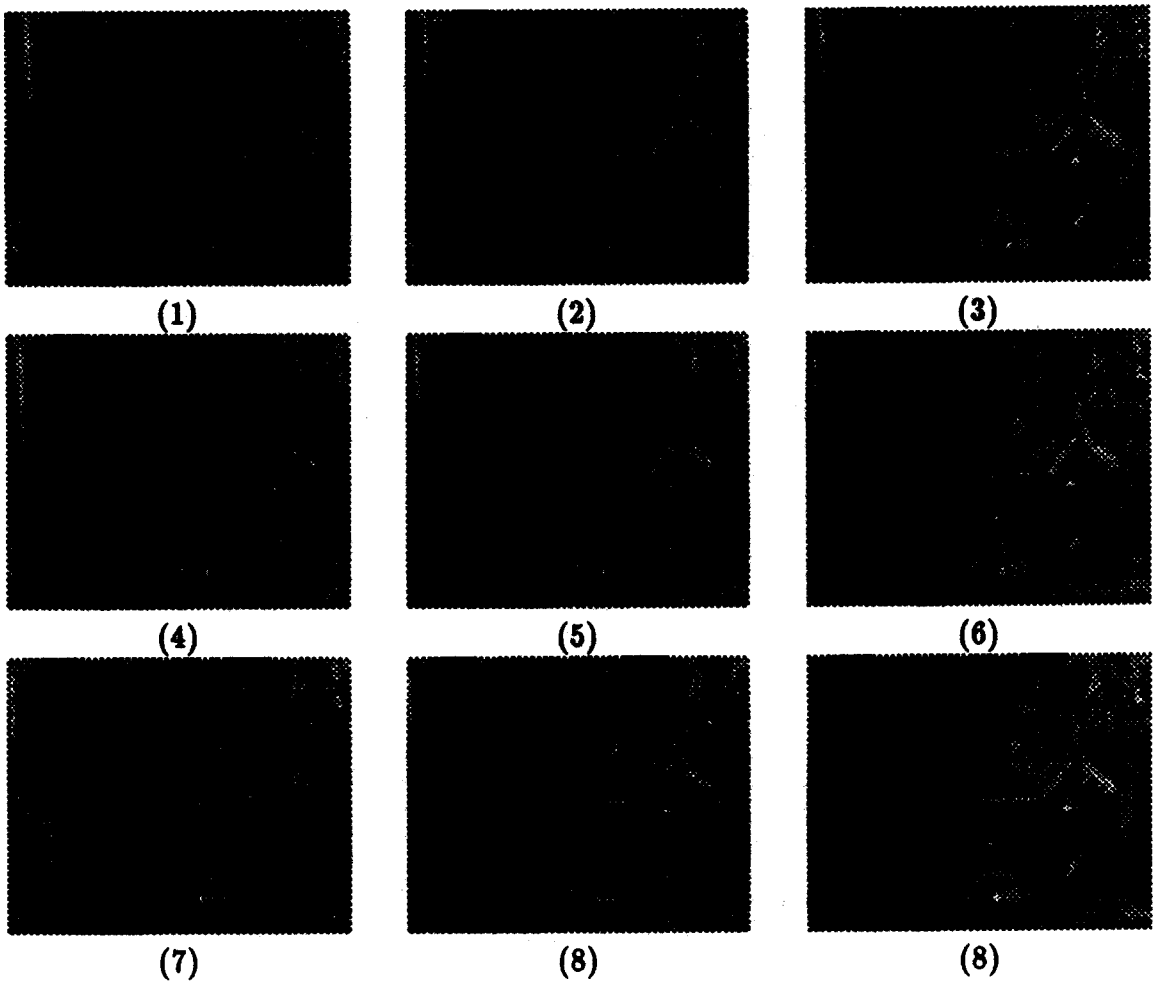

Figure 10.7: The first 9 images from the cup sequence.

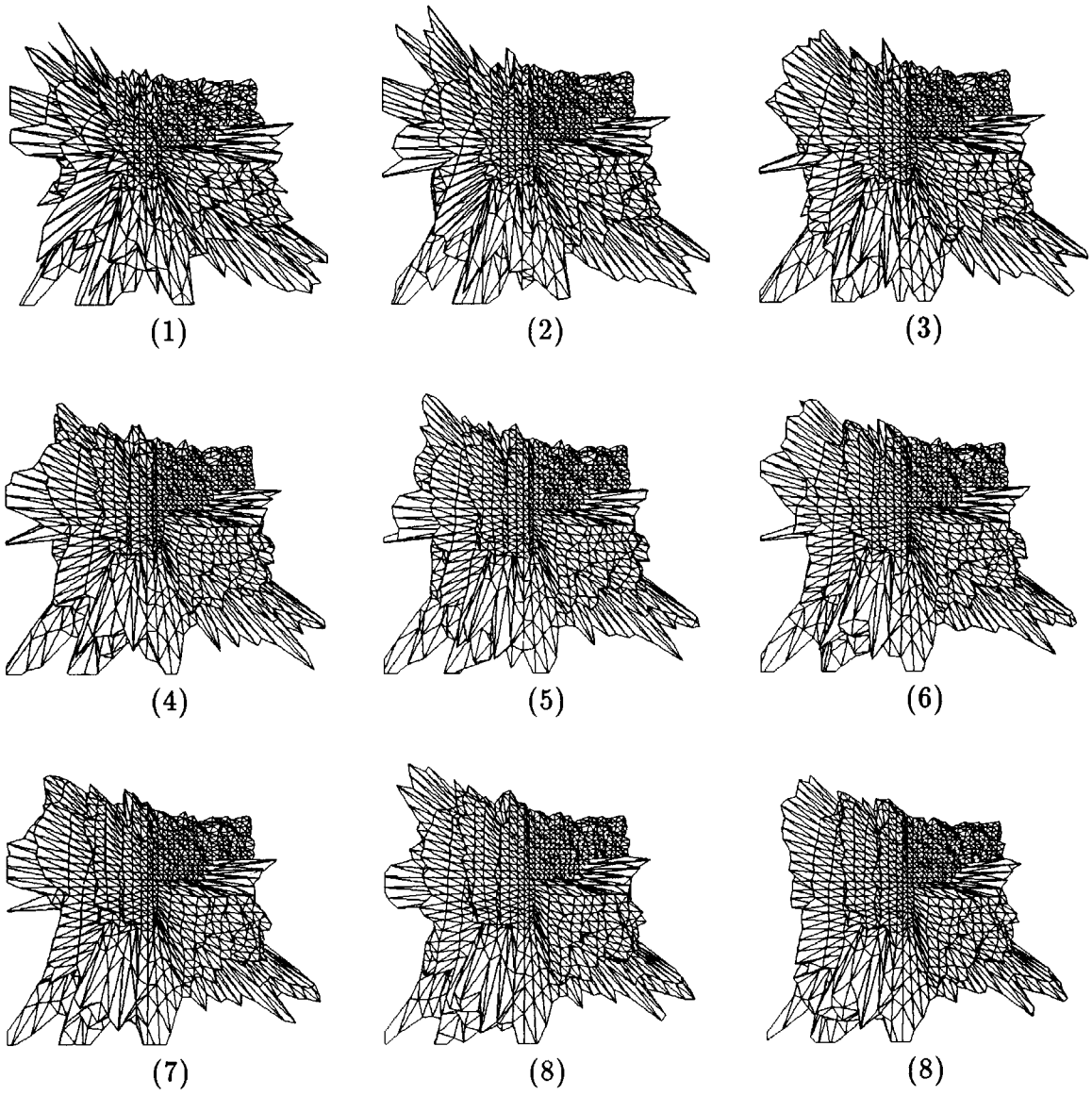

Figure 10.8: Wire frame rendering of the structure recovered from the cup sequence after each temporal iteration.

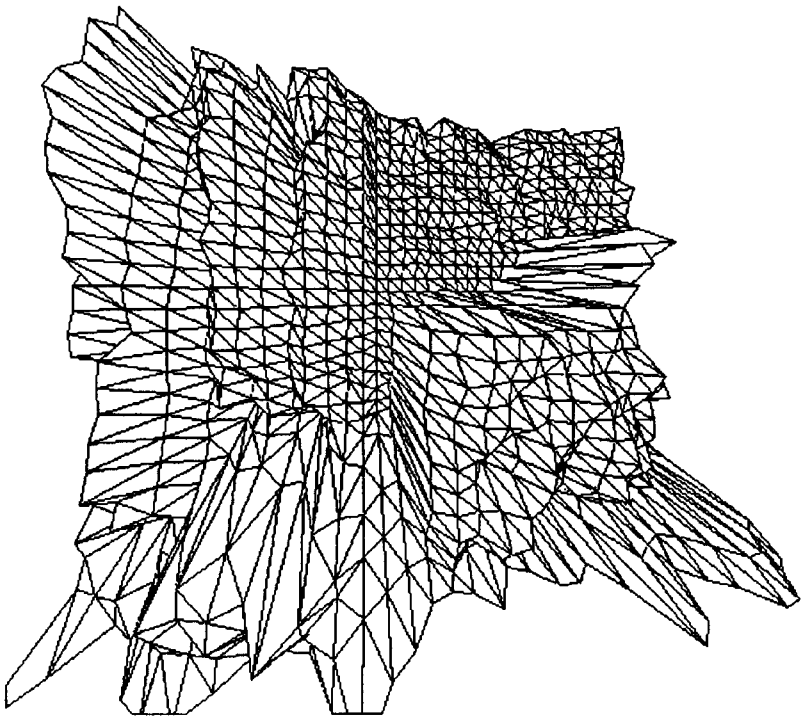

Figure 10.9: A closer look at the structure recovered after the 9th iteration of the temporal structure estimator using direct motion on the the cup sequence

Features and Faults

Chapter 11

This chapter discusses the temporal surface reconstruction scheme from several different perspectives. The first section is devoted to the analysis of complexity and run-time of the algorithm. The second investigates the possibility of implementing the algorithm on a parallel processor and analyzes the complexity of such an implementation. The third section summarizes the assumptions and approximations made in applying recursive estimation to the temporal surface reconstruction problem.

11.1 Computational Complexity and Run-Time

To analyze the serial complexity of the temporal reconstruction algorithm, let us consider the update and prediction stages separately and focus on a single iteration (time-step) of the filtering algorithm. As before we will assume that the image and the depth map have dimensions $n \times m$. The state and measurement vectors $\mathbf{x}, \mathbf{y}$ in the algorithms presented in this thesis contain $O(nm)$ values. The associated covariance matrices $\mathbf{R}, \mathbf{S}$ as well as the measurement matrix $\mathbf{C}$ have $O(n^2m^2)$ elements but due to their sparse nature, only $O(nm)$ are non-zero.

The update stage (4.16), (4.17), (4.18) requires multiplications and additions of the above vectors and matrices. Due to the sparse and banded nature of the matrices, this can be accomplished in time $O(nm)$. In addition, the matrix $\mathbf{S}$ must be inverted, which is achieved by the iterative Gauss-Seidel process (5.8). One iteration of this algorithm requires time $O(nm)$ so that the overall complexity of the update stage is $O(knm)$ where k is the number of iterations used in the Gauss-Seidel process. In our experiments, k was a small number, usually 20 to 50.

The prediction stage requires for each point in the depth map to be warped and then the depth map must be resampled. The warping processes each depth map entry once and therefore has complexity $O(nm)$. In the resampling step each of the $2(n-1)(m-1)$ warped triangular surface facets could in the worst case be intersected by all of the nm rays through grid point locations leading to a run time of $O(n^2m^2)$. This, however, is clearly a degenerate case that can be ruled out for small motions and real visual surfaces where a bounded small number of ray intersections

per triangulation facet can be safely assumed to reduce the expected run time to $O(nm)$. For convex surfaces, for example, a ray can intersect the surface at most two times. The approximative prediction algorithm presented in section 7.3 has a worst-case run-time of $O(nm)$.

In summary, the worst-case complexity of the temporal surface reconstruction algorithm is $O(n^2m^2)$. For all practical purposes, however, the expected run-time is only $O(nm)$ per time-step. Note that this complexity for a single measurement is the same as any one of the instantaneous surface reconstruction procedures from chapter 2.

In terms of the actual run-time for the implementations, the following results were obtained. The implementations were done on a Sun SPARCstation I. On an image of size 256×256, one iteration of the temporal surface reconstruction takes between 20 and 30 seconds depending on the visual mechanism used for the filter update. This time is almost evenly divided between the update and prediction stages where the time spent in the update stage is proportional to the number of Gauss-Seidel iterations. This is significant, as we have seen in the depth-from-shading example, in comparing the run-time with the repeated application of an instantaneous procedure. In the latter case, the Gauss-Seidel algorithm "starts from scratch" for each new measurement and may require a large number of iterations to converge. In the former case of temporal surface reconstruction, the Gauss-Seidel process in the update stage can be initialized with the predicted depth map from the previous time-step and will require considerably less iterations.

As a consequence, for all practical purposes, the temporal surface reconstruction algorithm is computationally *less expensive* than the repeated application of an instantaneous surface reconstruction.

11.2 Parallel Implementations

The temporal surface reconstruction procedure can benefit greatly from an implementation on a parallel processor as we will see in this section. We will investigate the complexity of an implementation on a SIMD processor such as the Connection Machine (TM).

The update stage can be implemented efficiently by arranging the processors in a two-dimensional grid and assigning one processor to each pixel of the image/depth map. Then the Jacobi method (see Golub and Van Loan [30]) can be used to solve the sparse matrix inversion problem (5.7). The update of a given pixel requires interaction with the four-connected neighbors. By our arrangement of processors, all values of the depth map can be updated at once, so that one Jacobi iteration can be computed in constant time. Therefore, the update stage of the temporal surface reconsruction takes only $O(k)$ time where k is the number of Jacobi iterations used.

For the prediction stage, we begin with the same arrangement of processors as above. Each processor computes the warping (7.3) of the point corresponding to its depth map entry. Obviously, this can be accomplished in a single time-step. Now the resulting depth value must be propagated to the new location (i, j) in the depth map for the resampling. Assuming that each processor can communicate with its four-connected neighbors, this may take time $O(n + m)$ (in the same worst-case scenario described for the serial case above) but for real surfaces and small motions this propagation will extend over only a small number of processors (pixels). After the propagation, each processor determines its new depth value by intersecting a ray through its location with the triangle facet given by the depth values that have been propagated to it (or by the simplified weighted scheme). This is again a constant-time operation.

In summary, a parallel SIMD implementation of the temporal surface reconstruction algorithm will require computation time proportional to the number of iterations k used in the update stage. I am aware of one effort to actually carry out such an implementation at the ETH Zürich. Most importantly, a SIMD implementation such as the one above is amenable to implementation directly on a single chip. Among the existing implementations of vision algorithms on single chips, the analog VLSI approach of Mead [69] is most easily extended to the recursive estimation task required for temporal surface reconstruction. The "artificial retina" can perform relaxation algorithms such as the one needed for the update procedure of the Kalman filter in a truly time-continuous fashion.

Another more common type of specialized hardware can effectively support the temporal surface reconstruction procedure. Graphics workstations such as the Silicon Graphics (TM) provide a fast storage called the Z-buffer that can be used to hold the depth map state vector and can be accessed quickly. The resampling step in the prediction stage is equivalent to the z-buffered rendering of a surface given as a triangular mesh. This is a common operation in these systems and is supported by special hardware.

Since the Kalman filter operations are linear, it is not surprising that the necessary computations can be implemented on a network-type architecture. Yeates [110] describes how a Kalman filtering algorithm can be implemented on a fully connected two-layer network by having the network use Newton's algorithm to perform the necessary matrix inversions. Network implementations are of particular interest because of the high speeds that can be achieved due to massively-parallel processing and because of the similarity to neuronal computation mechanisms found in humans.

11.3 Assumptions and Approximations

In applying the Kalman filter to the temporal surface reconstruction problem, we have made a number of assumptions and approximations. In some cases, these simplifications cause some of the useful properties of the filter (see section 4.6) to be lost. Additionally, it is important to understand these approximations in order to evaluate the temporal surface reconstruction scheme and its limitations. In what follows, the assumptions and approximations made throughout the thesis are compiled, evaluated and possible remedies are discussed.

1. *Extended Kalman filter:*
   Two of the visual mechanisms investigated in this thesis, structure from shading in chapter 9 and the first approach to structure from motion without optical flow 10, section 10.1 model the relationship between state and measurement as a non-linear one. This necessitates the use of the extended Kalman filter or the implicit Kalman filter. From section 4.6 we recall that these versions of the filter are only approximative and convergence is not guaranteed.

   In some cases, it is possible to formulate the filter in the desired linear form by making a different choice for the measurement vector $\mathbf{y}$ or the state $\mathbf{x}$. An example is the alternative temporal reconstruction scheme for direct structure from motion in section 10.2. By choosing the measurement vector to consist of the depth Z directly, the relationship to the state vector became linear and trivial. On the other hand, however, the derivation of the measurement covariance $\mathbf{R}$ requires another Taylor series approximation. In summary: we can trade off non-linearities in the measurement relationship for non-linearities in the measurement covariance.

2. *Correlation of measurements:*
   In determining the covariance matrix $\mathbf{R}$ of the measurement vector, we have neglected the effect of correlation between elements of the measurement vector in the case of structure from motion using optical flow chapter 8 and in one of the cases of structure from motion without optical flow chapter 10, section 10.2. This approximation simplifies the computation, since it ignores off-diagonal elements of the matrix $\mathbf{R}$ and makes the inversion of $\mathbf{R}$ required in the update stage of the filter (4.17) much easier. However, as we know from section 4.6, the convergence of the filter is no longer guaranteed under this approximation.

   There are two possible solutions to this problem. The first is to make a different choice for the elements of the measurement vector $\mathbf{y}$. This may lead to a much simpler covariance matrix as in the example of direct structure from motion chapter 10, section 10.1. This is the tradeoff discussed under the previous item. The second solution is to explicitly compute the covariances and carry them

through the computation. It would be interesting to investigate, whether this additional effort results in significantly improved estimates.

3. *Noise distribution:*
   The optimality of the linear Kalman filter requires the noise distribution to be Gaussian. In both of the linear filters described in this thesis (structure from motion using optical flow chapter 8 and structure form motion without optical flow chapter 10, section 10.2) the measurement quantities (optical flow and depth) will in general not have a Gaussian distribution, so that the optimality property is lost.

   As mentioned in section 4.6, the Gaussianity of measurement noise is necessary to show optimality of the Kalman filter among all possible estimators. If a Gaussian noise model is not applicable, there may be a non-linear estimator that outperforms the Kalman filter, but it remains optimal among linear estimators. This assumption is therefore not a crucial one.

4. *Prediction Resampling:*
   The resampling algorithm in the prediction stage (chapter 7) approximates the surface as being planar between sampling locations. This will introduce an error into the depth estimates, the magnitude of which depends on the frequency content of the surface. This will influence the convergence property of the filter, in particular it may prevent the estimate from converging completely to the ground truth.

   It is possible to improve the prediction stage to reduce the effect of this approximation. We could, for example, do a bilinear or bicubic approximation of the surface within a given facet. Such an approximation, although computationally more expensive, would also be compatible with the smoothness assumption on the surface structure. Whether or not a more accurate prediction stage is appropriate depends largely on the accuracy of the estimates obtainable with a given visual mechanism. The experimental results in this thesis indicate that for structure from motion, the errors as a result of the visual mechanism are far larger than any errors introduced by approximations within the filter while in structure from shading this may not be the case. Note that a small decrease in the certainty value of each depth estimate is used to explicitly represent the error introduced by the prediction stage.

5. *Certainty Prediction:*
   The prediction (4.20) of the certainty matrix $\mathbf{S}$ will in general not preserve the sparse and banded nature of this matrix. Since this property is crucial to maintain computational manageability, the change in off-diagonal elements was neglected by the separation in equation (6.8). This may cause the filter to lose its optimality and even its convergence property.

It will be very difficult to relax this assumption, since the computational feasibility depends on it. There are two arguments to support this approximation. First, it can be understood as a viewpoint-independent prior model of surface structure and therefore has some justification in a physical sense. Second, the experiments show that smooth surfaces can be recovered with this approximation. It may therefore be a tolerable one for practical purposes.

In addition to these assumptions and approximations made in the temporal reconstruction algorithm itself, there are two other sources of error that must be weighed carefully against the ones cited above in deciding where to begin with improvements.

The first is a systematic error in the modeling of the visual mechanism. In the structure from shading case (chapter 9) for example, the relationship between image brightness and surface gradients is only approximately modeled by the reflectance function (9.1) and does not follow this model exactly in real images. In the case of structure from motion without optical flow, the relationship between brightness gradients and depth is only approximately given by the brightness constancy assumption (10.2). These systematic errors depend on the particular image under consideration and can far outweigh the effect of the approximations made in constructing the filter algorithm.

The second is the approximate nature of the prior surface model. The ground truth surface will generally not obey a membrane or thin-plate model so that the smoothness constraint may actually drive the solution away from the true values. In particular, real surfaces contain depth discontinuities that cannot be modeled in this simple way and are therefore inaccurately recovered. While there exist methods for incorporating discontinuities into the prior surface model (see section 3.7), they are computationally more expensive and do not eliminate the adverse effect of an error in the prior model.

The analysis of these approximations and assumptions leads to one conclusion: theoretically derived properties of the temporal surface reconstruction scheme will be limited in their practical applicability. Experimentation must show how useful the method is for a given visual mechanism.

Conclusion

Chapter 12

The temporal surface reconstruction method presented in this thesis provides a way in which low-level "instantaneous" structure estimation procedures for various visual mechanisms can be embedded in a recursive estimation framework and thereby applied continuously to a sequence of frames. In the introduction chapter 1 we listed three requirements that a time-continuous structure estimation procedure should meet. Here we evaluate the temporal surface reconstruction method with respect to those criteria:

1. *Quality improvement:*
   The update stage of the filter incorporates each measurement in such a way, that the resulting state has minimal error variance. In the ideal filter the variance is guaranteed to always decrease over time.

2. *Motion transformations:*
   The prediction stage of the Kalman filter accounts for possible camera transformations between frames.

3. *Uncertainty representation:*
   Uncertainty is represented explicitly by covariance/certainty matrices and the update stage that combines old estimate and new measurement to produce a new estimate weights its inputs with their covariances to take this uncertainty into account.

4. *Computational simplicity:*
   The temporal surface reconstruction runs faster than the repeated application of instantaneous procedures while producing estimates of higher quality.

Although recursive estimation meets all of our initial requirements, we have seen that the nonlinearity of some visual mechanisms forces us to make assumptions that may cause some of the desirable theoretical properties of this scheme to be lost. Under such circumstances, experimentation plays an important part in verifying the

validity of assumptions and approximations and I believe the results presented here to be at least encouraging.

The appealing feature of this approach is the fact that it provides not only a uniform theoretical framework for temporal surface reconstruction but also yields algorithms of manageable complexity with robust results on real images. In addition, it constitutes a unifying theoretical framework for previous work in surface reconstruction from sequences of images. Previous work on incremental estimation of dense structure by Matthies, Szeliski and Kanade (see section 3.5) and Heel (see section 3.8) can be understood as the application of simplified versions of the temporal surface reconstruction scheme to specific visual mechanisms.

Future work in temporal structure estimation will proceed along five major lines:

1. Opportunities for relaxing the current set of assumptions and approximations necessary to apply recursive estimation to the temporal surface estimation problem (see section 11.3) will be explored. This can be achieved, for example, through new choices for the measurement vector, improved prediction schemes and enhanced modeling of uncertainty. Other sources of error such as the prior surface model and the modeling of the visual mechanism will also be the subject of improvement efforts.

2. Alternative representations for structure information will be investigated under the recursive estimation framework. One example is the voxel representation or the octree representation (see Szeliski [95], [94]). They can help overcome a major disadvantage of all of the previous representations: they can represent structure information that is not currently in view of the camera but was actually acquired much earlier in the image sequence and then disappeared from view. A second example are feature-based representations and how they can be understood as a sampling of a dense representation.

3. Other visual mechanisms will be embedded into the temporal framework and the models for exisiting visual mechanisms will be enhanced so that a reformulation of the embedding may become necessary. In particular, the problem of sensor fusion in which structure information is obtained from several visual mechanisms or non-visual sensors can be solved in an elegant way using the temporal reconstruction mechanism by simply expanding the measurement vector $\mathbf{y}$ to include all the measured quantities.

4. The temporal reconstruction scheme will find its way into implementations on parallel processors and special-purpose hardware. Ideally, the processing should be done immediately after the image acquisition and possibly on the same chip. All indications are that this task is feasible, but much more work is needed.

5. Ultimately, the temporal reconstruction scheme can only prove its usefulness when it becomes part of a functioning system. This will require that the structure information produced in a temporally continuous manner is utilized by some other algorithm that solves a particular application problem such as the navigation of a vehicle or the identification of a target. The navigation problem specifically is the subject of ongoing research by the author.

Hopefully, the theoretical and experimental results in this thesis will provide a basis for addressing these problems and contribute to a better understanding of the temporal nature of visual analysis.

The Implicit Kalman Filter

Appendix A

In this appendix I will derive the equations for the implicit Kalman filter introduced in section 4.3. The measurement equation of a dynamical system is given by

$$\mathbf{g}(\mathbf{y}_k - \mathbf{v}_k, \mathbf{x}_k) = \mathbf{0}. \tag{A.1}$$

where $\mathbf{v}_k \sim N(\mathbf{0}, \mathbf{R}_k)$. We postulate a linear update

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}) \tag{A.2}$$

where $\mathbf{K}_k$ will be chosen to minimize the expected length of the error vector $\boldsymbol{\epsilon}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k$. With the help of the state covariance $\mathbf{P}_k$

$$E[(\boldsymbol{\epsilon}_k^+)(\boldsymbol{\epsilon}_k^+)^T] = E[(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k)(\hat{\mathbf{x}}_k^+ - \mathbf{x}_k)^T] = \mathbf{P}_k^+ \tag{A.3}$$

we can formulate the optimization problem as follows: determine the matrix $\mathbf{K}_k$ that minimizes $J = trace(\mathbf{P}_k^+)$.

Under the postulated update equation (A.2) the estimation error is

$$\boldsymbol{\epsilon}_k^+ = \hat{\mathbf{x}}_k^+ - \mathbf{x}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}) - \mathbf{x}_k. \tag{A.4}$$

The error covariance becomes

$$\begin{aligned} \mathbf{P}_k^+ \;=\; & \mathbf{P}_k^- + E[\boldsymbol{\epsilon}_k^- \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}_k)]\mathbf{K}_k^T + \mathbf{K}_k E[\mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y})(\boldsymbol{\epsilon}_k^-)^T] + \\ & \mathbf{K}_k E[\mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y})\mathbf{g}^T(\hat{\mathbf{x}}_k^-, \mathbf{y})]\mathbf{K}_k^T \end{aligned} \tag{A.5}$$

To find the optimal $\mathbf{K}_k$ we differentiate J with respect to $\mathbf{K}_k$ and equate the result to zero. We obtain

$$\mathbf{K}_k = -E[\boldsymbol{\epsilon}_k^- \mathbf{g}^T(\hat{\mathbf{x}}_k^-, \mathbf{y}_k)](E[\mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y})\mathbf{g}^T(\hat{\mathbf{x}}_k^-, \mathbf{y})])^{-1} \tag{A.6}$$

This can be simplified by Taylor series expansion of $\mathbf{g}$:

$$\mathbf{g}(\mathbf{x}_k, \mathbf{y}_k - \mathbf{v}_k) = \mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}_k) + \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{x}}_k^-}(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) + \frac{\partial \mathbf{g}}{\partial \mathbf{y}_k}(-\mathbf{v}_k) + \ldots \tag{A.7}$$

Recall from (A.1) that the left hand side is zero and neglect the higher order terms to obtain

$$\mathbf{g}(\hat{\mathbf{x}}_k^-, \mathbf{y}_k) = \mathbf{C}_k(\hat{\mathbf{x}}_k^- - \mathbf{x}_k) + \mathbf{D}_k\mathbf{v}_k \tag{A.8}$$

where

$$\mathbf{C}_k = \frac{\partial \mathbf{g}}{\partial \hat{\mathbf{x}}_k^-} \quad \text{and} \quad \mathbf{D}_k = \frac{\partial \mathbf{g}}{\partial \mathbf{y}_k} \tag{A.9}$$

Now we can substitute the linear approximation (A.8) into the expression for the gain (A.6)

$$\mathbf{K}_k = -\mathbf{P}_k^+\mathbf{C}_k^T[\mathbf{C}_k\mathbf{P}_k^-\mathbf{C}_k^T + \mathbf{D}_k\mathbf{R}_k\mathbf{D}_k^T]^{-1} \tag{A.10}$$

where we have used the fact that $E[\boldsymbol{\epsilon}_k^-\mathbf{v}_k^T] = \mathbf{0}$ and $E[(\boldsymbol{\epsilon}_k^-)(\boldsymbol{\epsilon}_k^-)^T] = \mathbf{P}_k^-$. The linearized approximation also simplifies the expression (A.5) for the updated covariance

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k\mathbf{C}_k)\mathbf{P}_k^- \tag{A.11}$$

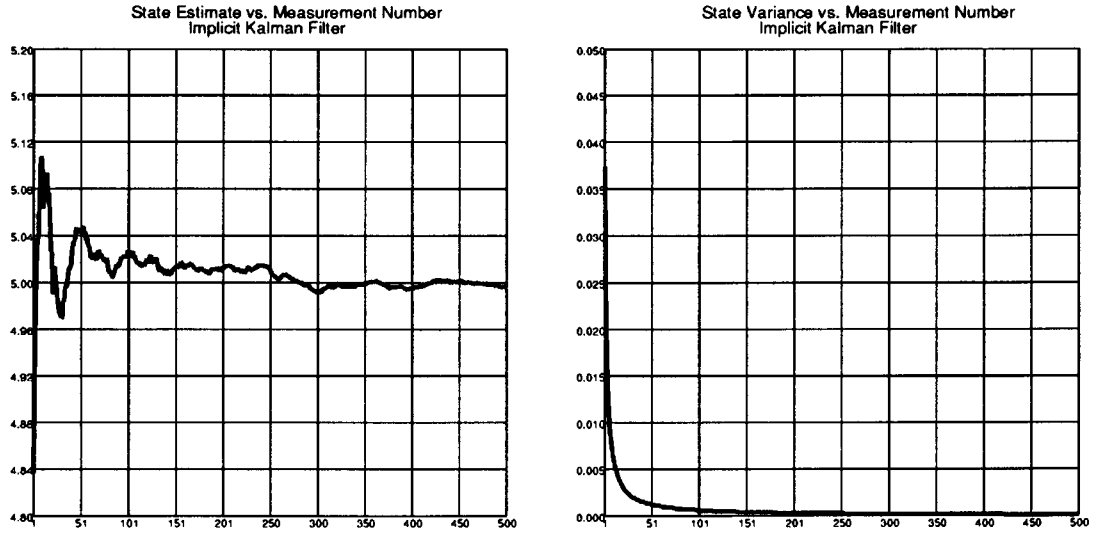

Figure A.1: State estimate and variance from the implicit Kalman filter experiment.

To illustrate the operation of the implicit Kalman filter, I have simulated the system

$$x_{k+1} = x_k \tag{A.12}$$

$$x_k y_k + C = 0 \tag{A.13}$$

with a true value $x = 5$, the constant $C = 10$ and a measurement noise of standard deviation $\sigma_y = 0.1$ (2 percent). The filter was initialized to $x_0 = 4$ and run for 500 iterations. The resulting state and variance estimates are shown in figure A.1.

Prediction of Estimate Covariances

Appendix B

The prediction stage of the Kalman filter for temporal surface reconstruction transforms the current estimate and its covariance from one time-step to the next and thereby accounts for interframe motions of the camera with respect to the scene. Chapter 7 describes in detail how the prediction of the state (depth values) can be accomplished, but gives only a simple approximation for the corresponding transformation of the covariances in section 7.2. Although this approximation is used in most of the experiments and is computationally much simpler, it is possible to determine the predicted values of the covariance/certainty more accurately by propagating them through the depth prediction equations. The derivation here takes advantage of the viewpoint-independent model of surface smoothness and therefore only treats the diagonal of the certainty matrix $\mathbf{S} = \mathbf{P}^{-1}$. Moreover, since most results about random variables are in terms of variances, the derivation is in terms of variances p which are the inverses of the diagonal entries in $\mathbf{S}$.

It remains to determine the variance values of the warped depth map. In essence, the warped values of Z are some function of the input values of Z i.e. the output value is a random variable which is some function of several input random variables. More formally: we interpret the given values of $Z(x_j, y_i)$ as normally distributed random variables with variances $p(x_j, y_i)$. What are the value of $p(x_j, y_i)$ after the warping?

B.1 Variance Propagation

Let us first establish some basic facts about propagation of variances. Let Z_1, Z_2 be two uncorrelated random variables with variances $\sigma^2_{Z1}, \sigma^2_{Z2}$. Then the random variable

$$Z = aZ_1 + bZ_2 \tag{B.1}$$

has the variance

$$\sigma^2_Z = a^2\sigma^2_{Z1} + b^2\sigma^2_{Z2} \tag{B.2}$$

The only assumption made here is that Z_1, Z_2 are uncorrelated.

In the more general case Z can be an arbitrary function of several random variables, say

$$Z = f(Z_1, Z_2). \tag{B.3}$$

In this case we approximate f by its Taylor series around the point of interest and neglect all but the first-order terms. Then we apply the above rule for a linear combination of random variables to obtain

$$\sigma_Z^2 = (\frac{\partial f}{\partial Z_1})^2 \sigma_{Z1}^2 + (\frac{\partial f}{\partial Z_2})^2 \sigma_{Z2}^2 \tag{B.4}$$

where the dertivatives must be evaluated at the particular point (Z_1, Z_2) of interest. This relationship is easily extended to n independent variables. The assumptions made here are zero correlation between Z_1 and Z_2 and the fact that $f()$ can be approximated by the first terms of its Taylor series near Z_1, Z_2.

B.2 Variances of the warped depth values

Having established how variances propagate through functions we need merely determine the functional relationship between input and output depth values and propagate the variances through these functions. Let us determine where in our algorithm we actually compute an output value of Z and how it is computed.

There are two ways in which an output value of Z can be computed. The first is by resampling in step 3 of the algorithm. There we compute the intersection of the ray through a pixel location (x, y) with a spatial triangle. The spatial triangle is determined by the 3D coordinates of the corner points (X_i', Y_i', Z_i') for $i = 1, 2, 3$. So our output value of Z is some function (which is determined by the interpolation procedure) of the corner coordinate values. Each corner coordinate is the result of warping one point (X_i, Y_i, Z_i) from the input surface. The warping function is simply a linear combination of the input point coordinates. Finally we note that the X and Y components of each original point are obtained by inverse perspective projection

$$X_i = \frac{x_i Z_i}{f} \quad \text{and} \quad Y_i = \frac{y_i Z_i}{f}. \tag{B.5}$$

Thus, each Z value obtained in step 3 of the algorithm is a function of three depth values Z_1, Z_2, Z_3 in the input depth map. We determine this functional relationship below.

The second possibility is when the depth value is obtained by extrapolating in step 4 of the algorithm. In this case a depth value Z is computed as the average of some depth values $Z_1, \ldots Z_k$ in its immediate neighborhood. This is a fairly simple linear relationship. The variance propagation for this case is also discussed in detail below.

B.2.1 Variances of interpolated depth values

We determined above that each output depth value Z is a function of three input depth values Z_1, Z_2, Z_3. Let us begin by determining the relationship between these values and the corresponding values (X_1', Y_1', Z_1'), (X_2', Y_2', Z_2'), (X_3', Y_3', Z_3') after warping. We combine the equations of motion

$$\begin{bmatrix} X_i' \\ Y_i' \\ Z_i' \end{bmatrix} = - \begin{bmatrix} U \\ V \\ W \end{bmatrix} - \begin{bmatrix} -1 & -C & B \\ C & -1 & -A \\ -B & A & -1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \tag{B.6}$$

with the inverse perspective projection

$$X_i = \frac{x_i Z_i}{f} \tag{B.7}$$

$$Y_i = \frac{y_i Z_i}{f} \tag{B.8}$$

to obtain

$$X_i' = -U + b_{xi} Z_i \tag{B.9}$$

$$Y_i' = -V + b_{yi} Z_i \tag{B.10}$$

$$Z_i' = -W + b_{zi} Z_i \tag{B.11}$$

where

$$b_{xi} = x/f + Cy/f - B \tag{B.12}$$

$$b_{yi} = -Cx/f + y/f + A \tag{B.13}$$

$$b_{zi} = Bx/f - Ay/f + 1. \tag{B.14}$$

After the motion warping, we compute the output depth by intersecting a ray through a grid point with a spatial triangle. As we recall, there are three cases by which Z can be obtained within the interpolation procedure.

1. There is exactly one point of intersection between the ray and the spatial triangle.

2. The ray lies in the same plane as the spatial triangle and it has at most one point in common with each edge of the spatial triangle.

3. The ray lies in the same plane as the spatial triangle and it coincides with one edge of the spatial triangle.

In the first case, the output Z value is computed as

$$Z = \lambda f = \frac{D_\lambda}{D} f \tag{B.15}$$

where

$$\begin{aligned} D &= (X_1 - X_3)[(Z_2 - Z_3)y - (Y_2 - Y_3)f] - \\ &\quad (Y_1 - Y_3)[(Z_2 - Z_3)x - (X_2 - X_3)f] + \\ &\quad (Z_1 - Z_3)[(Y_2 - Y_3)x - (X_2 - X_3)y] \end{aligned} \tag{B.16}$$

and

$$\begin{aligned} D_\lambda &= -X_3[(Y_1 - Y_3)(Z_2 - Z_3) - (Y_2 - Y_3)(Z_1 - Z_3)] \\ &\quad +Y_3[(X_1 - X_3)(Z_2 - Z_3) - (X_2 - X_3)(Z_1 - Z_3)] \\ &\quad -Z_3[(X_1 - X_3)(Y_2 - Y_3) - (X_2 - X_3)(Y_1 - Y_3)] \end{aligned} \tag{B.17}$$

where the primes have been omitted. If we use the above equations (B.9) - (B.11) to replace the variables in these expressions, we find

$$D = aZ_1Z_2 + bZ_1Z_3 + cZ_2Z_3 \tag{B.18}$$

where

$$a = b_{x1}(b_{z2}y - b_{y2}f) - b_{y1}(b_{z2}x - b_{x2}f) + b_{z1}(b_{y2}x - b_{x2}y) \tag{B.19}$$

$$b = b_{x1}(b_{y3}f - b_{z3}y) - b_{y1}(b_{x3}f - b_{z3}x) + b_{z1}(b_{x3}y - b_{y3}x) \tag{B.20}$$

$$c = -b_{x3}(b_{z2}y - b_{y2}f) + b_{y3}(b_{z2}x - b_{x2}f) - b_{z3}(b_{y2}x - b_{x2}y). \tag{B.21}$$

Further we have

$$D_\lambda = aZ_1Z_2 + bZ_1Z_3 + cZ_2Z_3 + dZ_1Z_2Z_3 + eZ_1Z_3^2 + fZ_2Z_3^2 \tag{B.22}$$

where

$$a = Ua_1 - Va_2 + Wa_3 \tag{B.23}$$

$$b = Ub_1 - Vb_2 + Wb_3 \tag{B.24}$$

$$c = Uc_1 - Vc_2 + Wc_3 \tag{B.25}$$

$$d = -b_{x3}a_1 + b_{y3}a_2 - b_{z3}a_3 \tag{B.26}$$

$$e = -b_{x3}b_1 + b_{y3}b_2 - b_{z3}b_3 \tag{B.27}$$

$$f = -b_{x3}c_1 + b_{y3}c_2 - b_{z3}c_3 \tag{B.28}$$

in which we have abbreviated

$$a_1 = b_{y1}b_{z2} - b_{y2}b_{z1} \tag{B.29}$$
$$b_1 = b_{z1}b_{y3} - b_{y1}b_{z3} \tag{B.30}$$
$$c_1 = b_{y2}b_{z3} - b_{y3}b_{z2} \tag{B.31}$$
$$\tag{B.32}$$
$$a_2 = b_{x1}b_{z2} - b_{x2}b_{z1}$$
$$b_2 = b_{x3}b_{z1} - b_{x1}b_{z3} \tag{B.33}$$
$$c_2 = b_{x2}b_{z3} - b_{x3}b_{z2} \tag{B.34}$$
$$\tag{B.35}$$
$$a_3 = b_{x1}b_{y2} - b_{x2}b_{y1}$$
$$b_3 = b_{x3}b_{y1} - b_{x1}b_{y3} \tag{B.36}$$
$$c_3 = b_{x2}b_{y3} - b_{y2}b_{x3}. \tag{B.37}$$

As a result of these tedious manipulations we are now able to express an interpolated value of Z as a function of three input depth values:

$$Z = Z(Z_1, Z_2, Z_3) = f\frac{D_\lambda(Z_1, Z_2, Z_3)}{D(Z_1, Z_2, Z_3)} \tag{B.38}$$

If the variances for the input depth values are p_1, p_2, p_3, we assume that they are uncorrelated and the above functional relationship can be approximated by the first terms of the Taylor series we can use the variance propagation (B.4)

$$p = (\frac{\partial Z}{\partial Z_1})^2 p_1 + (\frac{\partial Z}{\partial Z_2})^2 p_2 + (\frac{\partial Z}{\partial Z_3})^2 p_3 \tag{B.39}$$

to determine the output variance p. It remains to determine the partial derivatives

$$\frac{\partial Z}{\partial Z_i} = f\frac{\partial}{\partial Z_i}\frac{D_\lambda}{D} = \frac{\frac{\partial D_\lambda}{\partial Z_i}D - D_\lambda\frac{\partial D}{\partial Z_i}}{D^2}. \tag{B.40}$$

The partial derivatives of D are obtained easily from (B.18)

$$\frac{\partial D}{\partial Z_1} = aZ_2 + bZ_3 \tag{B.41}$$
$$\frac{\partial D}{\partial Z_2} = aZ_1 + cZ_3 \tag{B.42}$$
$$\frac{\partial D}{\partial Z_3} = bZ_1 + cZ_2 \tag{B.43}$$

and similarly those for D_λ from (B.22)

$$\frac{\partial D_\lambda}{\partial Z_1} = aZ_2 + bZ_3 + dZ_2Z_3 + eZ_3^2 \tag{B.44}$$

$$\frac{\partial D_\lambda}{\partial Z_2} = aZ_1 + cZ_3 + dZ_1Z_3 + fZ_3^2 \tag{B.45}$$

$$\frac{\partial D_\lambda}{\partial Z_3} = bZ_1 + cZ_2 + dZ_1Z_2 + 2eZ_1Z_3 + 2fZ_2Z_3. \tag{B.46}$$

Note that a, b and c are computed differently for D and D_λ. We must also take care not to confuse the coefficient f used in the expression for D_λ with the focal length f. This result enables us to compute the variance p of every output depth value Z obtained by interpolation in the case where the ray has exactly one point in common with the spatial triangle.

In the second case listed above we consider how a depth value is obtained if the interpolation ray is parallel to the spatial triangle but does not coincide with one of its edge segments. In this case we recall that Z is obtained by interpolating between the two endpoints (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) of the triangle edge segment:

$$Z = f\frac{d_\lambda}{d} \tag{B.47}$$

with

$$d = x(Y_1 - Y_2) - y(X_1 - X_2) \tag{B.48}$$

and

$$d_\lambda = (X_1 - X_2)Y_2 - (Y_1 - Y_2)X_2. \tag{B.49}$$

All coordinate components are after the motion warping although the primes have been omitted. We express both d and d_λ in terms of the depth values of the endpoints Z_1 and Z_2 before motion warping as done before:

$$d = aZ_1 + bZ_2 \tag{B.50}$$

where

$$a = xb_{y1} - yb_{x1} \tag{B.51}$$

$$b = yb_{x2} - xb_{y2} \tag{B.52}$$

and similarly

$$d_\lambda = aZ_1 + bZ_2 + cZ_1Z_2 \tag{B.53}$$

in which

$$a = b_{y1}U - b_{x1}V \tag{B.54}$$

$$b = b_{x2}V - b_{y2}U \tag{B.55}$$

$$c = b_{x1}b_{y2} - b_{y1}b_{x2}. \tag{B.56}$$

We compute the partial derivatives

$$\frac{\partial d}{\partial Z_1} = a \tag{B.57}$$

$$\frac{\partial d}{\partial Z_2} = b \tag{B.58}$$

and

$$\frac{\partial d_\lambda}{\partial Z_1} = a + cZ_2 \tag{B.59}$$

$$\frac{\partial d_\lambda}{\partial Z_2} = b + cZ_1 \tag{B.60}$$

which we need for

$$\frac{\partial Z}{\partial Z_1} = \frac{\frac{\partial d_\lambda}{\partial Z_1} d - d_\lambda \frac{\partial d}{\partial Z_1}}{d^2} \tag{B.61}$$

$$\frac{\partial Z}{\partial Z_2} = \frac{\frac{\partial d_\lambda}{\partial Z_2} d - d_\lambda \frac{\partial d}{\partial Z_2}}{d^2}. \tag{B.62}$$

Then the variance p of the output Z becomes

$$p = (\frac{\partial Z}{\partial Z_1})^2 p_1 + (\frac{\partial Z}{\partial Z_2})^2 p_2. \tag{B.63}$$

In the third and last case the interpolation ray coincides with at least one edge segment of the spatial triangle. Suppose the depth values of the endpoints are Z_1, Z_2 before motion warping. After motion warping they become

$$Z_1' = -W + b_{z1} Z_1 \quad \text{and} \quad Z_2' = -W + b_{z2} Z_2 \tag{B.64}$$

Recall that the algorithm assigned the smaller of the two values Z_1', Z_2' as the output value Z. Hence, if $Z_1' < Z_2'$ then the output variance would be

$$p = b_{z1}^2 p_1 \tag{B.65}$$

otherwise

$$p = b_{z2}^2 p_2. \tag{B.66}$$

From the point of view of implementation we see that the computation of variances cannot simply be added to the depth computation outlined in the previous section. A completely different approach is necessary in order to have the b_{xi}, b_{yi}, b_{zi} values available in the interpolation stage. Although the depth computation is identical the abstraction from the underlying geometry makes it less intuitive.

B.2.2 Variances of extrapolated depth values

Some of the output depth values are obtained through extrapolation in step 4 of the prediction algorithm. As described above we use a very simple extrapolation procedure. In which an unassigned depth value is set to the average of its assigned neighbors. This extrapolation was motivated by assuming smoothness of the surface. Nevertheless, the choice of variance at this location should indicate the fact, that no information was available there so that subsequent updates can contribute maximally to the value at this point. This is particularly important when objects enter the field of view that exhibit a depth discontinuity with respect to the previously visible objects. We therefore reset the variance value of extrapolated depth map entries to their initial values (see chapter 6).

An Implementation Example

Appendix C

This appendix describes the implementation of the temporal surface reconstruction algorithm for the structure from optical flow visual mechanism described in chapter 8. Although the update stage of the filter is completely specified by the choices of state, measurement and the associated covariances in that chapter, the execution of the update procedure (4.16), (4.17), (4.18) still requires some straightforward manipulations that are shown here for one example. Anyone trying to implement a temporal surface reconstruction algorithm will find the details presented here useful.

C.1 The Update Equations

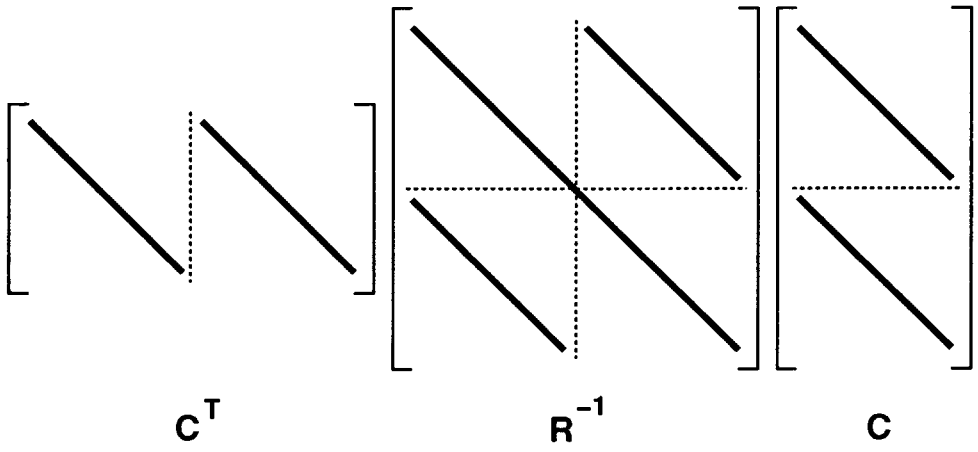

Figure C.1: The structure of the matrices in the filter update stage

We begin by simplifying the covariance/certainty update (4.16)

$$\mathbf{S}_k^+ = \mathbf{S}_k^- + \mathbf{C}_k^T \mathbf{R}_k^{-1} \mathbf{C}_k \tag{C.1}$$

Recall that the matrix $\mathbf{C}$ is a $2nm \times nm$ matrix and $\mathbf{R}$ is a $2nm \times 2nm$ matrix with the sparse banded structure shown in figure C.1. Because of the sparse nature of the matrix $\mathbf{R}$, the inverse $\mathbf{R}^{-1}$ is easy to obtain: it is sparse and banded with the same structure as $\mathbf{R}$. The elements are

$$\begin{aligned} r_{ii}^{-1} &= r_{i+nm,i+nm}/D_i && \text{for } 0 \le i < nm \\ r_{ii}^{-1} &= r_{i-nm,i-nm}/D_{i-nm} && \text{for } nm \le i < 2nm \\ r_{i,i+nm}^{-1} &= -r_{i,i+nm}/D_i && \text{for } 0 \le i < nm \\ r_{i+nm,i}^{-1} &= -r_{i+nm,i}/D_i && \text{for } 0 \le i < nm \end{aligned}$$

where $D_i = r_{ii}r_{i+nm,i+nm} - r_{i+nm,i}r_{i,i+nm}$. Note that $r_{i,i+nm} = r_{i+nm,i}$.

The matrix $\mathbf{C}^T\mathbf{R}^{-1}\mathbf{C}$ is a $nm \times nm$ *diagonal* matrix, so that only the diagonal elements of $\mathbf{S}$ are updated in (C.1):

$$s_{ii}^+ = s_{ii}^- + \frac{c_{ii}^2 r_{ii} + c_{i,i+nm}^2 r_{i+nm,i+nm} - 2c_{i+nm,i}c_{i,i+nm}r_{i,i+nm}}{r_{ii}r_{i+nm,i+nm} - r_{i,i+nm}r_{i+nm,i}} \tag{C.2}$$

The state update (4.17), (4.18)

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + (\mathbf{S}_k^+)^{-1}\mathbf{C}_k^T\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-) \tag{C.3}$$

is performed through the iterative Gauss-Seidel solution of (5.7)

$$\mathbf{p} = (\mathbf{S}_k^+)^{-1}\mathbf{q} \tag{C.4}$$

where $\mathbf{p} = \hat{\mathbf{x}}_k^+ - \hat{\mathbf{x}}_k^-$ and $\mathbf{q} = \mathbf{C}_k^T\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-)$. To compute $\mathbf{q}$, we begin with $\mathbf{v} = \mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^-$

$$\begin{aligned} v_i &= y_i - c_{ii}x_i && \text{for } 0 \le i < nm \\ v_i &= y_i - c_{i,i-nm}x_{i-nm} && \text{for } nm \le i < 2nm \end{aligned}$$

This vector $\mathbf{v}$ is multiplied from the right with $\mathbf{C}_k^T\mathbf{R}_k^{-1}$, a matrix that we have already determined above in the computation of the covariance update. The resulting vector $\mathbf{q}$ is

$$\begin{aligned} q_i = {} & [(c_{i+nm,i}r_{ii} - c_{ii}r_{i,i+nm})(y_{i+nm} - c_{i+nm,i}x_i) + \\ & (c_{ii}r_{i+nm,i+nm} - c_{i,i+nm}r_{i+nm,i})(y_i - c_{ii}x_i)]/(r_{ii}r_{i+nm,i+nm} - r_{i,i+nm}r_{i+nm,i}) \end{aligned} \tag{C.5}$$

Having obtained $\mathbf{q}$, we compute $\mathbf{p}$ by applying the Gauss-Seidel algorithm (5.8) to the relationship (C.4) and then the updated state vector is just

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{p}. \tag{C.6}$$

C.2 Hints and Hacks

In this section, four optional techniques are presented, that give you more control over the reconstruction process or improve the subjective quality of the results.

1. The covariance update (C.2) can result in negative values in the diagonal of the certainty matrix $\mathbf{S}_k^+$. This is perfectly normal if the full matrix $\mathbf{S}$ is maintained. However, we approximate the certainty matrix by a sparse banded version (6.8) in which only the diagonal varies. As a consequence, the diagonal values take the role of inverse variances and should be positive.

   A positive sign of the update certainty values can easily be insured by first computing the s_{ii}^+ as in (C.2) and if the result is negative, we set the measurement covariances $r_{i,i+nm}$, $r_{i+nm,i}$ to zero and recompute both s_{ii}^+ and q_i.

2. Instantaneous structure estimation algorithms (see chapter 2) allow to control the influence of the smoothness term on the reconstructed surface by means of a parameter λ. This explicit control can also be established in the temporal reconstruction algorithm. We begin by writing the separation (6.8) of the matrix $\mathbf{S}_k$ into the diagonal representation of uncertainty $\tilde{\mathbf{S}}_k$ and the representation of surface smoothness $\mathbf{S}$, the latter term now weighted with λ:

   $$\mathbf{S}_k = \tilde{\mathbf{S}}_k + \lambda \mathbf{S}. \tag{C.7}$$

   The Gauss-Seidel iterations for the update now have the form

   $$p_i^{(n+1)} = \frac{1}{\tilde{S}_{ii} + 4\lambda}(q_i + \lambda \bar{p}_i^{(n)}) \tag{C.8}$$

   where $\bar{p}_i^{(n)}$ is the sum of the four values of p that are nearest neighbors to $p_i^{(n)}$ on the depth map grid in the case of a membrane model of smoothness. This explicit control of the resulting surface smoothness is useful for subjective evaluations. Compare, for example, the result of the pepsi experiment in section 8.3 in which explicit smoothness control was used and the result on the same sequence in section 10.3 without smoothness control.

3. The depth Z is always positive and usually also bounded from above. The subjective quality of results can be improved, by introducing a limit check just after the state update of the filter which ensures that depth values are positive or within certain bounds. Two strategies are possible: One is simply to truncate depth values to bring them into the desired range. An alternative more compatible with surface smoothness is to fill in depth values outside the range with the average of neighbors that are within the range. In either case, the certainty value corresponding to a modified depth value must be reset to its initial value to indicate the complete uncertainty about the actual value there.

4. Very few images allow three-dimensional structure to be recovered everywhere with a given visual mechanism. The reason is that preconditions for the visual mechanism such as the applicability of a particular reflectance function or the brightness constancy assumption are not met. An example of such a situation is the surface of the soda can in the pepsi experiment which has large regions of uniform brightness, so that the brightness constancy constraint does not allow the extraction of information. In all these cases, it is impossible to recover structure data in certain image regions.

   It is, however, possible to subjectively improve the structure estimate in these regions. The key idea is that low quality depth estimates will be identified by low values in the corresponding certainty values. A useful strategy is to designate all structure estimates with certainty below a chosen threshold as "low quality" and to fill them in with neighboring depth values of higher quality.

Bibliography

[1] J. Aloimonos and J.-Y. Herve. Correspondenceless detection of depth and motion for a planar surface. Technical Report CAR-TR-357, Computer Vision Laboratory, Center for Automation Research, University of Maryland, April 1988.

[2] P. Anandan. Computing dense displacement fields with confidence measures in scenes containing occlusion. COINS Technical Report 84-32, University of Massachusetts, Amherst, December 1984.

[3] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2, 1989.

[4] N. Ayache and O. D. Faugeras. Maintaining representations of the environment of a mobile robot. Rapports de Recherche 789, INRIA, February 1988.

[5] R. Bajcsy and L. Lieberman. Texture gradient as a depth cue. *Computer Graphics and Image Processing*, 5(1), 1976.

[6] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3, 1989.

[7] J. Barron. Computing motion and structure from noisy, time-varying image velocity information. Technical Report RBCV-TR-88-24, University of Toronto, August 1988.

[8] S. Bharwani, E. Riseman, and A. Hanson. Refinement of environment depth maps over multiple frames. In *Proceedings of the Workshop on Motion: Representation and Analysis*, Charleston, S. C., May 1986.

[9] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. Research Report YALEU/DCS/RR-835, Yale University, 1990.

[10] A. Blake and A. Zisserman. *Visual Reconstruction.* MIT Press, 1987.

[11] R. C. Bolles and H. H. Baker. Epipolar-plane image analysis: A technique for analyzing motion sequences. In *IEEE Proceedings of the Third Workshop on Computer Vision: Representation and Control*, Bellaire, MI, October 1985.

[12] T. J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive estimation of 3-d kinematics and structure from a noisy monocular image sequence. *IEEE Transactions on Aerospace and Electronic Systems*, July 1990.

[13] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(1), January 1986.

[14] T. J. Broida and R. Chellappa. Kinematics and structure of a rigid object from a sequence of noisy images. In *Proceedings of the IEEE Workshop on Motion: Representation and Analysis*, Charleston, SC, May 1986.

[15] R. G. Brown. *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley & Sons, 1983.

[16] A. R. Bruss and B. K. P. Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21, 1983.

[17] H. Buelthoff and M. Fahle. Disparity gradients and depth scaling. AI Memo 1175, MIT Artificial Intelligence Laboratory, September 1989.

[18] S. Chandrashekhar and R. Chellappa. A two-step approach to passive navigation using a monocular image sequence. USC-SIPI Report 170, University of Southern California, 1991.

[19] J. Dengler. Estimation of discontinuous displacement vector fields with the minimum description length criterion. AI Memo 1265, MIT Artificial Intelligence Laboratory, October 1990.

[20] E. D. Dickmans. 4d-szenenanalyse mit integralen raum-/zeitlichen modellen. In *Mustererkennung 1987, 9. DAGM Symposium*, Braunschweig, W. Germany, September/October 1987.

[21] E. D. Dickmans. Subject-object discrimination in 4d-dynamic scene interpretation for machine vision. In *Proceedings Workshop on Visual Motion*, Irvine, CA, March 1989.

[22] O. D. Faugeras. 3-d shape representation. In O. D. Faugeras, editor, *Fundamentals in Computer Vision.* Cambridge University Press, 1983.

[23] O. D. Faugeras, N. Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *Proceedings of the IEEE Conference on Robotics and Automation*, San Francisco, CA, April 1986.

[24] O. D. Faugeras and F. Lustman. Motion and structure from motion in a piecewiese planar environment. Rapports de Recherche 856, INRIA, June 1988.

[25] W. O. Franzen. Natural representation of motion in space-time. In *Proceedings of the DARPA Image Understanding Workshop*, Boston, MA, April 1988.

[26] D. Geiger and F. Girosi. Parallel and deterministic algorithms for mrfs: surface reconstruction and integration. AI Memo 1114, MIT Artificial Intelligence Laboratory, June 1989.

[27] D. Geiger and F. Girosi. Parallel and deterministic algorithms for MRFs: surface reconstruction and integration. In O. D. Faugeras, editor, *Lecture Notes in Computer Science, Vol. 427: Computer Vision - ECCV 90.* Springer Verlag, Berlin, 1990.

[28] A. Gelb (Ed.). *Applied Optimal Estimation.* The MIT Press, Cambridge, MA, 1974.

[29] S. Geman and D. Geman. Stochastic realxation. gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6), 1984.

[30] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, 1983.

[31] W. E. L. Grimson. *From Images to Surfaces.* The MIT Press, 1981.

[32] W. E. L. Grimson. Computational experiments with a feature based stereo algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(1), 1985.

[33] D. J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4), 1987.

[34] D. J. Heeger and A. Jepson. Simple method for computing 3d motion and depth. In *ICCV*, Osaka, Japan, December 1990.

[35] J. Heel. Dynamical systems and motion vision. AI Memo 1037, MIT Artificial Intelligence Laboratory, April 1988.

[36] J. Heel. Dynamic motion vision. In *Proceedings SPIE Conference on Advances in Intelligent Robotics Systems*, Philadelphia, PA, November 1989.

[37] J. Heel. Dynamic motion vision. In *Proceedings of the DARPA Image Understanding Workshop*, Palo Alto, CA, May 1989.

[38] J. Heel. Direct dynamic motion vision. In *Proceedings of the IEEE Conference on Robotics and Automation*, Cincinnati, Ohio, May 1990.

[39] J. Heel. Direct estimation of structure and motion from multiple frames. AI Memo 1190, MIT Artificial Intelligence Laboratory, March 1990.

[40] J. Heel. Temporal surface reconstruction. In *Proceedings of the International Conference on Computer Vision*, Osaka, Japan, December 1990.

[41] J. Heel and S. Rao. Temporal integration of visual surface reconstruction. In *Proceedings of the DARPA Image Understanding Workshop*, Pittsburgh, PA, September 1990.

[42] E. C. Hildreth. Computations underlying the measurement of visual motion. *Artificial Intelligence*, 23, 1984.

[43] B. K. P. Horn. *Shape from Shading: a Method for Obtaining the Shape of a Smooth Opaque Object from One View.* PhD thesis, MIT, 1970.

[44] B. K. P. Horn. *Robot Vision.* The MIT Press, 1986.

[45] B. K. P. Horn. Height and gradient from shading. AI Memo 1105, MIT Artificial Intelligence Laboratory, May 1989.

[46] B. K. P. Horn and Brooks M. J. *Shape from Shading.* The MIT Press, 1989.

[47] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17, 1981.

[48] B. K. P. Horn and E. J. Weldon, Jr. Direct methods for recovering motion. *International Journal of Computer Vision*, 2, 1988.

[49] B.K.P. Horn, R. J. Woodham, and W. M. Silver. Determining shape and reflectance using multiple images. AI Memo 490, MIT Artificial Intelligence Laboratory, 490 1978.

[50] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1-3), 1981.

[51] S.-L. Iu and K. Wohn. Estimation of 3-d motion and structure based on a temporally-oriented approach with the method of regression. In *Proceedings Workshop on Visual Motion*, Irvine, CA, March 1989.

[52] J. L. Jezouin and N. Ayache. 3d structure from a monocular sequence of images. In *ICCV*, Osaka, Japan, December 1990.

[53] A. Kaelldahl. Simultaneous estimation of motion and shape from a sequence of feature point projections. Thesis LIU-TEK-LIC-1989, Linkoeping University, 1989.

[54] R. E. Kalman. A new approach to linear filtering and prediction problems. In *Transactions of the ASME - Journal of Basic Engineering*, volume 35-45, March 1960.

[55] K. Kanatani. Structure from motion without correspondence: General principle. In *Proceedings Image Understanding Workshop*, Miami, FL, December 1985.

[56] J. Kender. *Shape from Texture.* PhD thesis, CMU, Department of Computer Science, 1979.

[57] M. J. Korsten. *Three-Dimensional Body Parameter Estimation from Digital Images.* PhD thesis, University of Twente, Enschede, Netherlands, 1989.

[58] E. Krotkov. Focusing. *International Journal of Computer Vision*, 1, 1987.

[59] H. C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. In S. Ullman and W. Richards, editors, *Image Understanding 1984.* Ablex, 1984.

[60] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194, October 1976.

[61] D. Marr and T. Poggio. A theory of human stereo vision. AI Memo 451, MIT Artificial Intelligence Laboratory, November 1977.

[62] J. L. Marroquin. *Probabilistic Solution of Inverse Problems.* PhD thesis, Massachusetts Institute of Technology, 1985.

[63] J. L. Marroquin. Deterministic bayesian estimation of markovian random fields with applications to computational vision. In *Proceedings of the International Conference on Computer Vision*, London, England, June 1987.

[64] L. Matthies. *Dynamic Stereo Vision.* PhD thesis, Carnegie Mellon University, 1989.

[65] L. Matthies, R. Szeliski, and R. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. Technical Report CMU-CS-87-185, Computer Science Department, Carnegie-Mellon University, December 1987.

[66] L. Matthies, R. Szeliski, and R. Kanade. Incremental estimation of dense depth maps from image sequences. In *Proceedings Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 1988.

[67] L. Matthies, R. Szeliski, and R. Kanade. Kalman filter-based algorithms fo restimating depth from image sequences. *International Journal of Computer Vision*, 3, 1989.

[68] S. J. Maybank. Filter based estimates of depth. In *Proceedings of the British Machine Vision Conference*, Oxford, United Kingdom, 1990.

[69] C. Mead. *Analog VLSI and Neural Systems.* Addison-Wesley, 1989.

[70] A. Mitiche. On kineopsis and computation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8, 1986.

[71] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5), September 1986.

[72] N. Navab, R. Deriche, and Faugeras O. D. Recovering 3d motion and structure from stereo and 2d token tracking cooperation. In *ICCV*, Osaka, Japan, December 1990.

[73] S. K. Nayar and Y. Nakagawa. Shape from focus: An effective approach for rough surfaces. In *Proceedings of the IEEE conference on robotics and automation*, Cincinnati, Ohio, May 1990.

[74] S. Negahdaripour and B. K. P. Horn. A direct method for locating the focus of expansion. AI Memo 939, MIT Artificial Intelligence Laboratory, January 1987.

[75] S. Negahdaripour and B. K. P. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, January 1987.

[76] S. Negahdaripour and B. K. P. Horn. A direct method for locating the focus of expansion. *Computer Vision, Graphics and Image Processing*, Vol. 46(3), June 1989.

[77] A. Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(2), 1984.

[78] A. Pentland. A new sense for depth of field. *IEEE Transactions on Pattern* Computer Science Department, Carnegie-Mellon University, December 1987.

[79] T. Poggio, E. Gamble, and J. Little. Parallel integration of vision modules. *Science*, Vol. 242, October 1988.

[80] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035), September 1985.

[81] A. Rougee, B. C. Levy, and A. S. Willsky. An estimation-based approach to the reconstruction of optical flow. Technical Report LIDS-P-1663, MIT Laboratory for Information and Decision Systems, Cambridge, MA, April 1987.

[82] H. S. Sawhney and J. Oliensis. Image description and 3d interpretation from image trajectories under rotational motion. Technical Report COINS TR 89-90, University of Massachusetts, Amherst, September 1989.

[83] J.-P. Schott. *Three-Dimensional Motion Estimation Using Shading Information in Multiple Frames.* PhD thesis, Massachusetts Institute of Technology, 1989.

[84] I. K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1), January 87.

[85] H. Shariat and K. E. Price. How to use more than two frames to estimate motion. In *Proceedings of the Workshop on Motion: Representation and Analysis*, Charleston, S. C., May 1986.

[86] T. M. Sobh and K. Wohn. Recovery of 3-d motion and structure by temporal fusion. In *SPIE Conference on Sensor Fusion II*, Philadelphia, PA, 1989.

[87] M. Spetsakis and J. Aloimonos. A multi-frame approach to visual motion perception. Technical Report CAR-TR-407, Computer Vision Laboratory, Center for Automation Research, University of Maryland, November 1988.

[88] M. Spetsakis and J. Aloimonos. Optimal motion estimation. In *Proceedings Workshop on Visual Motion*, Irvine, CA, March 1989.

[89] K. A. Stevens. *Surface Perception from Local Analysis of Texture and Contour.* PhD thesis, MIT, Department of Psychology, 1979.

[90] J. Stuller and G. Krishnamurthy. Kalman filter formulation of low-level television motion estimation. *Computer Vision, Graphics and Image Processing*, 21(2), February 1983.

[91] M. Subbarao. Interpretation of image flow: Rigid curved surfaces in motion. *International Journal of Computer Vision*, 2, 1988.

[92] M. Subbarao. *Interpretation of Visual Motion: A Computational Study.* Morgan Kaufmann Publishers, Inc., 1988. Research Notes in Artificial Intelligence.

[93] R. S. Szeliski. Bayesian modeling of uncertainty in low-level vision. Technical Report CMU-CS-88-169, Computer Science Department, Carnegie Mellon University, August 1988.

[94] R. S. Szeliski. Real-time octree generation from rotating objects. Technical Report CRL 90/12, DEC Cambridge Research Laboratory, December 1990.

[95] R. S. Szeliski. Shape from rotation. Technical Report CRL 90/13, DEC Cambridge Research Laboratory, December 1990.

[96] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics and Image Processing*, 24, 1983.

[97] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(4), July 1986.

[98] W. B. Thompson and J. K. Kearney. Inexact vision. In *Proceedings Workshop on Motion: Representation and Analysis*, May 1986.

[99] C. Tomasi and T. Kanade. Shape and motion without depth. In *ICCV*, Osaka, Japan, December 1990.

[100] C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. Technical Report CMU-CS-91-105, Carnegie Mellon University, January 1991.

[101] R. Y. Tsai. Multiframe image point matching and 3-d surface reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2), March 1983.

[102] R. Y. Tsai and S. T. Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-29(6), December 1981.

[103] S. Ullman. The interpretation of structure from motion. *Proceedings Royal Society of London*, 203, 1979.

[104] S. Ullman. Maximizing rigidity: The incremental recovery of 3-d structure from rigid and rubbery motion. AI Memo 721, MIT Artificial Intelligence Laboratory, June 1983.

[105] A. M. Waxman and S. Ullman. Surface structure and three-dimensional motion from image flow kinematics. *International Journal of Robotics Research*, 4(3), 1985.

[106] D. Weinshall. Direct computation of qualitative 3d shape and motion invariants. In *International Conference on Computer Vision*, Osaka, Japan, December 1990.

[107] A. S. Willsky. *Digital signal processing and control and estimation theory.* The MIT Press, Cambridge, MA, 1979.

[108] J. J. Wu, R. E. Rink, T. M. Caelli, and G. Gourishankar. Recovery of the 3-d location and motion of a rigid object through camera image (an extended kalman filter approach). *International Journal of Computer Vision*, 3, 1988.

[109] M. Yamamoto. The image sequence analysis of three-dimensional dynamic scenes. Technical Report UDC 681.3.056, Electrotechnical Laboratory, Agency of Industrial Science and Technology, Japan, May 1988.

[110] M. C. Yeates. Neural networks can implement complex signal processing algorithms. *Neural Networks*, 1, 1988.

[111] A. Zapp. *Automatische Strassenfahrzeugfuehrung durch Rechnersehen.* PhD thesis, Universitaet der Bundeswehr Muenchen, Fakultaet fuer Luft- und Raumfahrttechnik, September 1988.

This blank page was inserted to preserve pagination.

CS-TR Scanning Project
Document Control Form

Date : 6 / 8 / 95

Report # AI-TR-1296

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: 147 (155-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): 2,10

Photographs/Tonal Material (by page number): 28,80,84,92,96,99,105,111

Other (note description/page number):

Description :	Page Number:
IMAGE MAP (1) UNNUMBERED TITLE PAGE	
(2-147) PAGES #'ED 2-147	
(148-152) SCANCONTROL, COVER, SPINE, DOD(2)	
(153-155) TRGTS (3)	

Scanning Agent Signoff:
Date Received: 6/8/95 Date Scanned: 6/12/95 Date Returned: 6/15/95

Scanning Agent Signature: Michael W. Cook

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1991	technical report

4. TITLE AND SUBTITLE
Temporal Surface Reconstruction

5. FUNDING NUMBERS
N00014-85-K-00124
N00014-86-K-0685

6. AUTHOR(S)
Joachim Heel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER
AI-TR 1296

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Information Systems
Arlington, Virginia 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AD-A259494

11. SUPPLEMENTARY NOTES
None

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution of this document is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis investigates the problem of estimating the three-dimensional structure of a scene from a sequence of images. Structure information can be recovered from images through a number of *visual mechanisms* such as shading, motion and stereo. Image information is commonly available in a time-continuous fashion and this work proposes a method for estimating structure information in a *temporally continuous* manner for a variety of visual mechanisms.

Structural information about a scene is represented in a dense *depth map* in which the distance to the scene is stored for each pixel location in the image. In addition, uncertainty about the structure values is represented explicitly by the estimate covariance. This representation is maintained over time by a stochastic recursive estimator, the *Kalman filter*. The estimator consists of two stages which are repeated

(continued on back)

14. SUBJECT TERMS (key words)
3D reconstruction, structure estimation, temporal vision, Kalman filter, surface reconstruction

15. NUMBER OF PAGES
149

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED

Block 13 continued:

for each new arriving image. The update stage improves the current depth estimate by incorporating the latest image measurement. It depends on the particular visual mechanism being employed and amounts to an iterative relaxation algorithm similar to conventional single-frame algorithms. The prediction stage transforms the current depth estimate into the next time-step to account for changes in the depth values that may occur if the camera moves relative to the (rigid) scene during the acquisition of the sequence. This step requires a three-dimensional transformation (translation and rotation) of each depth map entry followed by a resampling operation to maintain the regular map representation.

The temporal reconstruction algorithm is described in detail for the recovery of structure from motion with and without optical flow and for structure from shading. Extensive experimental evaluation shows that the temporal algorithm not only improves the quality of estimates significantly over time but also requires orders of magnitude less time per image than previous "instantaneous" techniques.

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.**