Methods for Structure from Motion

Henrik Aanæs

Informatics and Mathematical Modelling
Technical University of Denmark
Ph.D. Thesis No. 122
Kgs. Lyngby 2003

IMM

This dissertation is submitted to Informatics and Mathematical Modeling at the Technical University of Denmark in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

The work has been supervised by Associate Professor Rasmus Larsen.

Kgs. Lyngby, September , 2003

Henrik Aanæs

Resumé

Structure from motion problematikken beskæftiger sig med at estimere 3D struktur fra 2D afbildninger heraf. Denne problemstilling er en af de mest populære og velstuderede inden for coputer vision. Dette skyldes tildels, at den er akademisk interessant, men også at den har et stort kommercielt potientiale.

Denne afhandling er rapporteringen af et studie inden for dette område, structure from motion. Dette studie har reulteret i udviklingen af nye metoder til at imødekomme nogle af de problemer, der er inden for området. Hovedsagligt drejer dette sig om at gøre de såkaldte faktoriserings–metoder mere robuste, at undersøge hvorledes stivheds–antagelsen kan blødes op samt at undersøge alternative måder at løse overflade–estimerings– problemet på.

Abstract

Structure from motion, the problem of estimating 3D structure from 2D images hereof, is one of the most popular and well studied problems within computer vision. In part because it is academically interesting, but also because it holds a wealth of commercially very interesting prospects, e.g. within entertainment, reverse engineering and architecture.

This thesis is a study within this area of structure from motion. The result of the work, which this thesis represents is the development of new methods for addressing some of the problems within the field. Mainly in robustifying the factorization approach, relaxing the rigidity constrains, and in considering alternative ways of solving the surface estimation problem.

Acknowledgments

Writing acknowledgments, such as this, is always a treacherous task, in that it is all too easy to forget someone. In case I have done this, I hope you will accept my deepest apologies and my reassurance that no malice was intended.

It is my belief, that research such as this is best carried out as a communal project. As such I would like to start out by thanking all my co-authors[1] – Rune Fisker, Kalle Åström, Jens Michael Carstensen, Hans Bruun Nielsen, Nicolas Guilbert, Rasmus Larsen, J. Andreas Bærentzen, Jan Erik Solem, Anders Heyden, Fredrik Kahl and Charlotte Svensson – for their cooperation and invaluable input to our common work.

Secondly I would like to thank the image group here at informatics and mathematical modeling (IMM) and the vision group in Lund for supplying an inspiring, productive and fun atmosphere for doing this work. Special thanks goes out to my advisor Rasmus Larsen for his support, and to Fredrik Kahl. I would also like to thank Kalle Åström and Anders Heyden for making the stay in Lund possible.

On a broader scale, I would like to thank the IMM for supplying the larger framework for this work, and for an open atmosphere where ample advice is given across professional divides. I would especially like to thank Hans Bruun Nielsen for much help and many interesting discussions. But Per Christian Hansen, Poul Thyregod and Finn Kuno Christensen should also be acknowledged for their aid and help.

As with most Ph.D. projects, the time and effort put into this on has often exceeded normal office hours – by quite a lot actually. This has naturally drawn on the people closest to me for their patients and understanding. In this regard a special thanks goes to my girlfriend Solveig.

Lastly, I would like to thank DTU for funding this Ph.D. project Henrik Öjelund for interesting discussions on the subject, and Knut Conradsen for taking personal responsibility for getting me started right within the field.

[1] Mentioned in order of appearance

Contents

CHAPTER 1

Introduction

As the title indicates the subject of this thesis is structure from motion, and my work on the subject in conjunction with my Ph.D. work. My efforts in this relation has focused on developing new methods for this structure from motion problem.

This thesis is structured such, that in Part I an introductory overview of structure and motion is given, which also attempts to set my contributions in perspective. As such the introduction of the subject has been referred to this first part.

Part II contains the contributions of this thesis presented as individual self contained papers. The reason for this is two–fold, firstly many of these papers are extremely worked through, more than would be realistic here. Secondly the papers are rather orthogonal to each other, hence allowing the reader to only read the part of this thesis, which is of particular interest to him. All but the latest of the included papers have been accepted through a peer reviewed process and published internationally. Below a summary of all main publications in relation to this thesis are presented, however when more then one publication exist on the same work, only the most extensive is included, although this might not be the one accepted for publication due to time. ("*" Indicates the one comprising the chapter)

Chapter 8

* H. Aanaes, R. Fisker, K. Aström, and J.M. Carstensen. Robust factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1215–1225, 2002.

H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Factorization with erroneous data. In *Photogrametric Computer Vision*, 2002.

H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Factorization with contaminated data. In *Presentation and abstract at Eleventh International Workshop on*

Matrices and Statistics, 2002.

H. Aanæs, R. Fisker, and J. M. Carstensen. Robust structure and motion. In *The 9th Danish Conference on Pattern Recognition and Image Analysis, Aalborg*, pages 1–9, 2000.

Chapter 9

* H. B. Nielsen and H. Aanæs. Separation of structure and motion by data modification. Technical report, IMM, DTU, preliminary versions.

Chapter 10

* N. Guilbert, H. Aanæs, and R. Larsen. Integrating prior knowledge and structure from motion. In *Proceedings of the Scandinavian Image Analysis (SCIA'01)*, pages 477–481, Bergen, Norway, 2001.

Chapter 11

* H. Aanæs, R. Larsen, and J.A. Bærentzen. Pde based surface estimation for structure from motion. In *Scandinavian Conference on Image Analysis 2003*, 2003.

Chapter 12

* H. Aanæs and J. A. Bærentzen. Pseudo–normals for signed distance computation. In *Vision, Modeling, and visualization 2003, Munich, Germany*, 2003.

J. A. Bærentzen and H. Aanæs. Computing discrete signed distance fields from triangle meshes. Technical Report 21, IMM, DTU, 2002.

Chapter 13

* J. E. Solem, H. Aanæs, and A. Heyden. PDE based shape from specularities. In *Scale Space, Isle of Skye, UK*, 2003.

Chapter 14

* H. Aanæs and F. Kahl. Deformable structure and motion. *working paper*, 2003.

H. Aanæs and F. Kahl. Estimation of deformable structure and motion. In *Vision and Modelling of Dynamic Scenes*, 2002.

H. Aanæs and F. Kahl. A factorization approach for deformable structure from motion. In *SSAB*, 2002.

H. Aanæs and F. Kahl. Estimation of deformable structure and motion. Technical report, Centre for Mathematical Sciences, Lund University, January 2002.

Chapter 15

- * C. Svensson, H. Aanæs, and F. Kahl. Structure estimation and surface triangulation of deformable objects. In *Scandinavian Conference on Image Analysis 2003*, 2003.

Part I

Structure from Motion Overview

CHAPTER 2

Structure from Motion

Structure from motion is one of the most popular problems within computer vision, and has received tremendous attention over the last decade or so. It deals with the estimation of 3D structure from 2D images, and as such can be seen as an automation and extension of photogrammetry. To attempt a more formal definition structure from motion is: *The estimation of the 3D structure of a usually rigid object and relative camera motion from 2D images hereof, when the external camera parameters are unknown but translating*. It is noted, that sometimes the internal camera parameters are also unknown, and that the rigidity constraint can be relaxed.

Structure from motion can be seen as the simultaneous solving of two dual and well known problems, namely the surveying a an unknown structure from known camera positions, and the determining of ones position or camera motion from known fix–points, cf. Figure 2.1. However, this combination of estimation problems has the effect, that the solution can only be determined up to an Euclidean similarity transform, i.e. scale, rotation and translation. This ambiguity, does not render the results useless, as it still has many applications within entertainment, reverse engineering and architecture, to mention a few.

The standard approach to constructing a structure from motion system is illustrated in Figure 2.2. Here more or less salient features – usually the high curvature points – are extracted and tracked through the images, cf. Chapter 3, upon which the structure from motion is calculated based on these features applying multiple view geometry, cf. Chapter 4. As illustrated in Figure 2.2 there is an interaction between the multiple view geometry estimation and the feature tracking, this consists of the multiple view relationships being used to regularize the feature tracking. Following the structure estimation based on the features, an estimate of the camera parameters (external and if needed internal), and the 3D structure of the extracted features are at hand. If, as is usually the case, the extracted features are image

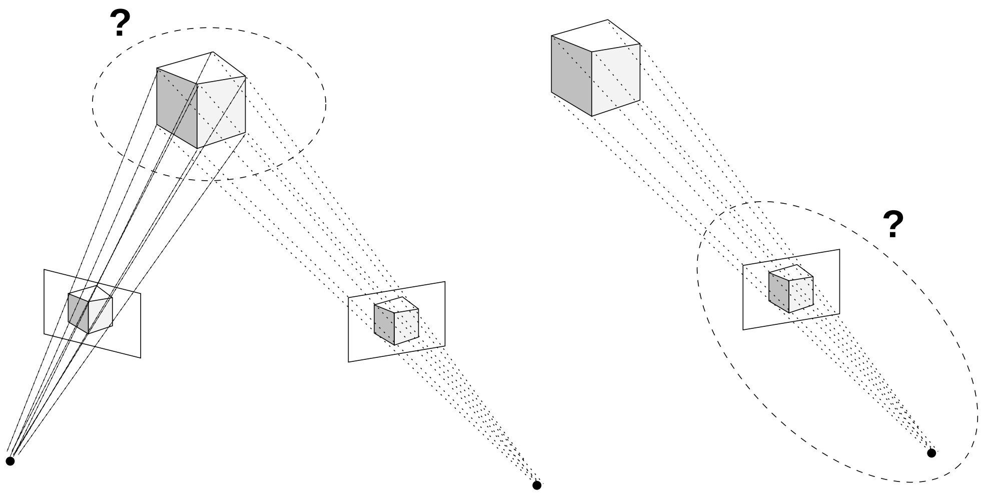


Figure 2.1: Structure from motion can be seen as solving two well known and dual problems simultaneously.

points, then the estimated 3D structure is a 3D point cloud. In that an estimate of the whole structure, and not just of some salient features, is sought, it is customary to proceed with a multiple view stereo algorithm, where the estimated camera parameters are assumed fixed, cf. Chapter 5.

An interpretation of the above described standard approach to structure from motion, is that it works by model or problem approximation. That is, attempting to estimate the full surface model and the camera parameter straight on is a futile task[1]. Hence the model/ problem is simplified to something that can be estimated/solved and the solution to this simplified problem is then used as an initial guess for the original problem. Here the structure is reduced to points, whereupon an estimate of the camera parameters and 3D point structure can be achieved. This camera structure – and possibly the 3D point structure – can then be used to estimate a full structure. Only recently was it proposed estimating the surface and the camera parameters together, following the estimation of the surface *based on the previously estimated camera parameters*, thus solving the original problem [218]. All in all structure from motion estimation utilizes layers of approximated models or problems, even more so, since solving of the structure from motion problem with points also uses hierarchies of models, e.g. factorization methods.

A further perspective on structure from motion is that it can be seen as the ’academic spearhead’ for a wide variety of 3D reconstruction techniques, i.e. 3D reconstruction from image data techniques that are made easier by construction but is very interesting from an application point of view. Examples hereof are: Laser scanners cf. [124], where the camera motion is known, and the feature tracking is made easier by projecting light onto the object;

[1] To the best of my knowledge.

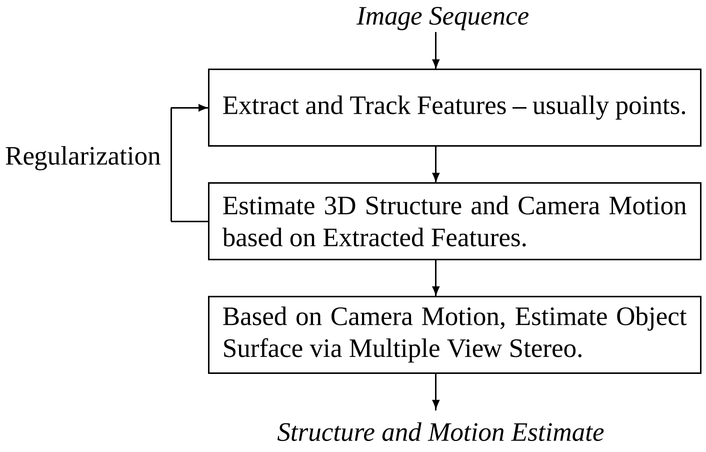


Image Sequence
Extract and Track Features – usually points.
Regularization
Estimate 3D Structure and Camera Motion based on Extracted Features.
Based on Camera Motion, Estimate Object Surface via Multiple View Stereo.
Structure and Motion Estimate

Figure 2.2: Schematic overview of proposed scheme. The dotted line denotes a feedback loop which should be included in future work.

the work of Debevec et al. [53, 52], which is a full functioning structure from motion system except for a user having to annotate all major structures; lastly there is the multiple view stereo work of Narayanna, Kanade and others [141, 140, 163] used to create virtual views of a sporting event, among other the NFL super–bowl.

In the following an overview and short discussion of the different parts of structure from motion is given. This is accompanied by a presentation of the contributions of this thesis within the respective areas.

CHAPTER 3

Feature Tracking

As the vigilant reader might have noticed, feature tracking is not part of the immediate research aim of this thesis. However, for completeness a short overview will be given here. An attempt at defining Feature tracking is; *Find the correspondence between two – or more – images. Understood as determining where the same physical entities are depicted in the different images in question.* See Figure 3.1.

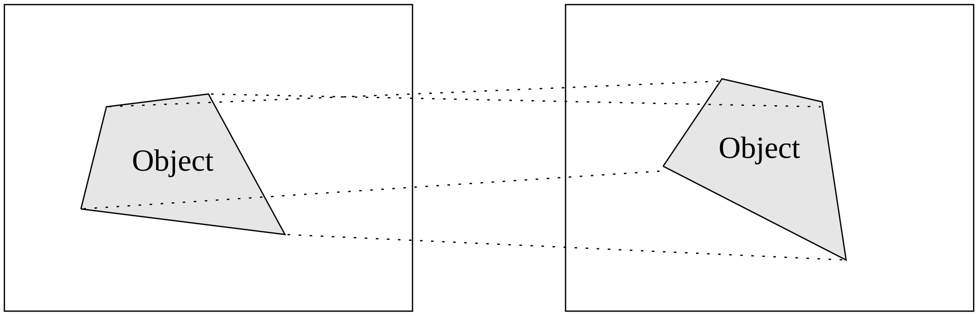


Figure 3.1: The aim of feature tracking.

The general problem of feature tracking is one of the the great problems of image interpretation. As such it is known under different names depending on the field of use and exact formulation, e.g. the correspondence problem, registration or optical flow to mention some of the most common. Hence there is a wealth of literature on the subject, and most any introductory image analysis text book will deal with the it, e.g. [75]. For a recent in depth survey aimed at 3D reconstruction, the reader is referred to Chapter 3 of [199]. A more dated but recommendable survey of the general registration problem is given in [29].

3.1 Tracking by Similarity

With today's technology computers are unable to perform higher order semantic inference in general. As such it is *not* possible for computers to recognize an object in two or more images, and from that higher order information deduce the solution to the tracking problem. General feature tracking algorithms are therefore confined to tracking by similarity of appearance, e.g. which corner in image 2. looks most similar to corner A. in image 1.

The measure or norm of similarity varies among proposed methods, cf. [199]. An obvious and commonly used measure is that of correlating image patches. As an example assume, that the features that should be tracked are points. Then the correlation measure can be used by correlating patches centered at the features, as illustrated in Figure 3.2.

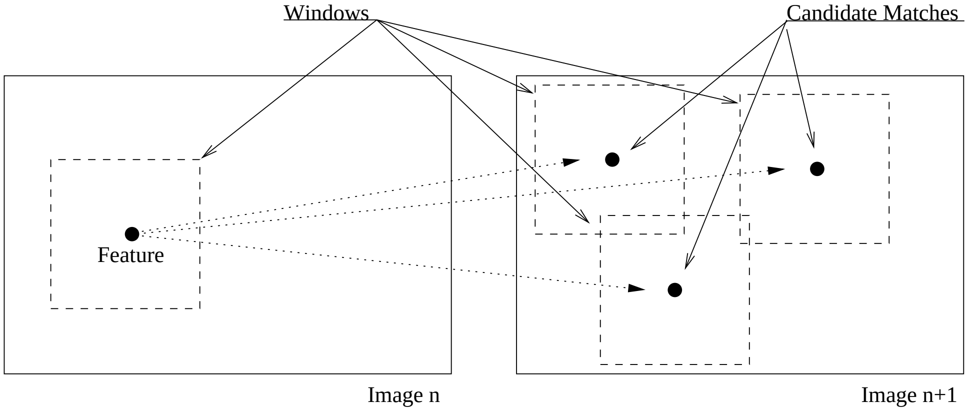


Figure 3.2: Matching features by finding correlation of surrounding windows.

An issue with tracking by similarity is that it implicitly assumes that changes between images are no bigger than the same physical entities still look 'the same'. As an example, consider the building in Figure 3.3(a). The assumption seems to hold well if the next image were Figure 3.3(b), however if it were Figure 3.3(c) it is more dubious if the assumption holds. As such many tracking algorithms would have difficulty tracing between the images of Figure 3.3(a) and Figure 3.3(c).

The above considerations is one of the main reasons why feature tracking is still a fundamentally hard problem to solve. Even though there is a wealth of uses for it and there has been enormous amounts of research done in the area, to my knowledge, a general full purpose algorithm does not exist. This does, however, not imply that no reliable algorithms for feature tracking exist, just that the ones that do are not general. Things like abiding by the 'things do not change to much' assumption will usually give good results. Limiting the

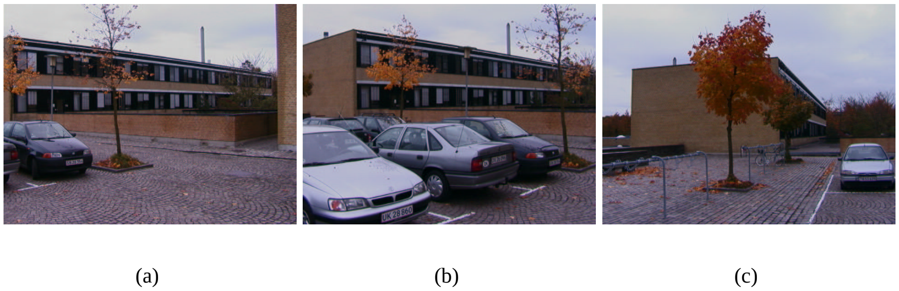


Figure 3.3: Different views of a building here at DTU.

object domain to e.g. only considering license plates, can also prove feasible. A convincing tracker is [21].

3.2 Aperture Problem

In the above we have found the correspondence between features in two images. Denote by the flow of a feature, the displacement vector between the feature and its correspondent[1]. Consider every pixel in an image as a feature, and determine the correspondence and flow vector for every such pixel. Then the vector field consisting of the flow vectors is termed the *optical flow*. Refer to [15] for a good but somewhat dated overview of optical flow estimation techniques.

A inherent limitation when solving the correspondence problem, is the so called aperture problem. The basic problem is illustrated in Figure 3.4. The aperture problem is that flow can only be determined across image gradients. And with noise it is seen that the higher the image gradient is in a given direction, the better the flow can be estimated. Hence the certainty of the optical flow is proportional to the image gradient.

The aperture problem is good to have in mind when features should be selected for matching. That is if explicit features should be matched i.e. not the optical flow. The direct implication is that features with high image gradients in all direction should be chosen. This is the same as a large second order derivative. Hence a good approach is to find the places in the image where the second order derivative is large, and then choose the corresponding physical entities as the feature to be matched. In structure from motion the most popular feature extractor is the Harris Corner Detector [86]. As an example consider Figures 3.5(a) and 3.5(b). Here it is definitely advisable to choose as features the corners of the drawings. For an in depth discussion of feature extraction the reader is referred to [171].

[1] this could be measured in change in pixel coordinates.

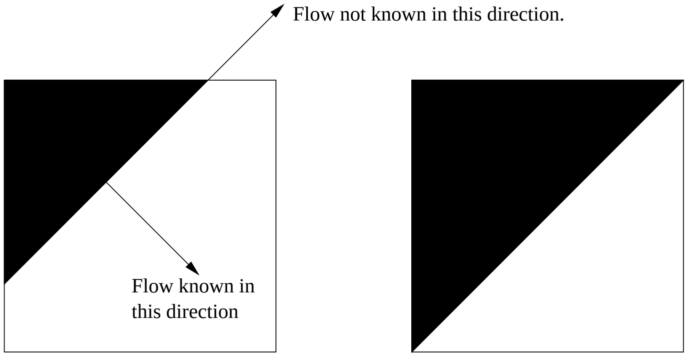

Flow not known in this direction.
Flow known in this direction

Figure 3.4: It is seen that the flow of a given feature can only be determined along the image gradient. This is an illustration of the aperture problem.

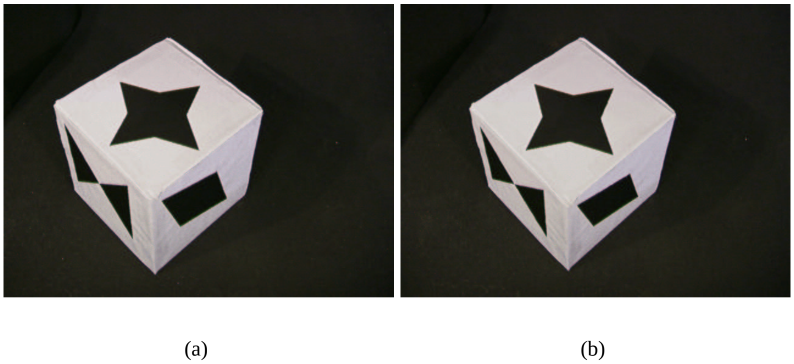


Figure 3.5: Different views of a cardboard box.

3.3 Constrains in Search Space

An enhancement to the basic approach of finding features in both images and comparing them all to each other, is to constrain the search space. This is another way of saying, that only some features in one image can match a given feature in the other. There are two main reasons for this, firstly to limit the number of computations required. Secondly this is a way to incorporate prior knowledge or assumptions of how the images are formed.

A very popular constraint is on the numerical size of the optical flow, implying that things have not moved to much between the images. Another constraint, popular within computer vision, is assuming rigid bodies and using the epipolar geometry, see Figure 3.6, where the search space for a feature in one image can be constrained to a line. Constraining with the epipolar geometry was proposed in [205, 207, 208, 222, 223]. Generally, it is the regularization with the epipolar geometry that reduce the mis–matching of features enough for the data to be applicable further in a structure from motion system. See Chapter 4 for more on epipolar geometry.

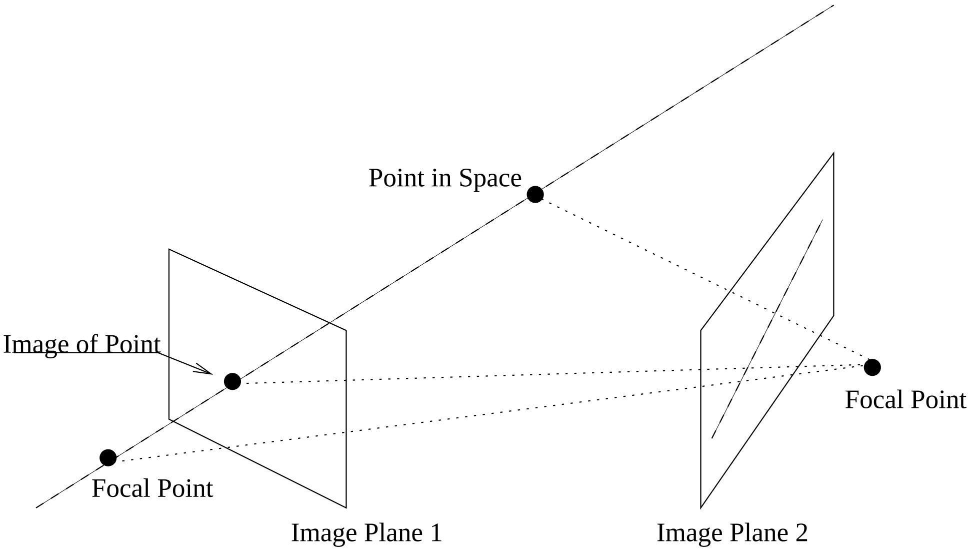


Figure 3.6: Assume that the object viewed by the two cameras does not change, e.g. the two images are taken at the same time, and the camera geometry is known. Then it is seen that a feature in one image constrains the location of its correspondence in the other image to a line.

3.4 Implementation Outline

Looking at the number of people working with structure from motion, there is naturally a myriad of ways in which features are tracked in this setting. However, in order to be more specific an outline of a 'standard' approach for matching images between two images will be given here.

1. Extract Features from the images, e.g. via Harris corner detector [86].
2. Match neighborhoods of features as illustrated in Figure 3.2, e.g. via correlation. Possibly constraining the search space.
3. Get one to one matches by solving the linear assignment problem, see below.
4. Robustly estimate the epipolar geometry, cf. [205, 207, 208, 222, 223]. Perhaps taking degenerate cases into account cf. [206].

5. If not stop goto 2.

Linear Assignment Problem

When all the features in one image have been compared to the features in the other, most features are likely to have more than one candidate for a match. However, since the features are seen as physical entities they should only match to one feature. Solving this problem such that the combined similarity is greates (or some cost the least) is done by posing it as a linear assignment problem. An efficient method for solving this is proposed in [108].

3.5 Summary

Here I have tried to convey the overall considerations in association with feature tracking in conjunction with structure from motion. Since I have tried to come with some general comments in a rather limited space, there are bound to be some generalization errors, for which I apologize, but overall I believe the conveyed picture to be rather accurate.

CHAPTER 4

Multiple View Geometry

The geometry of the viewing or imaging process from multiple views is by no mean a new area of research, and it is at the core of photogrammetry. In its modern sense[1], multiple view geometry has been a continually active area of research since the early part of the last century. The interested reader is referred to the manual by Slama [185], which also includes a historical overview. It should be noted that multiple view geometry is sometimes seen as synonymous with structure from motion. This is however not the nomenclature of this thesis.

Although multiple view geometry is a vintage academic discipline, the 1990's sparked a revolutionary development within the frame work of computer vision. The development has been such, that solutions to most of the considered issues could be presented in the excellent books by Hartley and Zisserman [89] in 2000 and Faugeras, Luong and Papadopoulo [69] in 2001, less then a decade later.

On a broad view, the focus of the computer vision community, in regards to multiple view geometry, centered on relaxing the constrains on, and automating the 3D estimation process. Results concerning relaxing the constrains include the factorization algorithms e.g. [111, 195, 204] allowing for the structure from motion problem to be solved without an initial guess needed, as in the non–linear bundle adjustment approaches e.g. [185, 211]. Together with the eight–point methods for estimating the fundamental matrix [90, 125] the factorization methods are sometimes referred to as direct methods. Another limitation surmounted is the ability to perform 3D reconstruction without known internal camera parameters, i.e. auto–calibration, see e.g. [68, 88, 95, 144, 159, 160]. However, relaxing the constrains also increases the number of ambiguities and degenerate configurations. Ambiguities within structure from motion have also been identified and studied, see e.g. [13, 31, 33, 109, 194].

As for automating the 3D reconstruction process, the main problem faced is that of the

[1] not considering geometry

unreliability of the feature tracking, as mentioned in the last chapter. So much effort has gone into developing robust statistical methods for estimating multiple view relations, such that outliers of the feature tracking could be identified and dealt with. To mention a few, there is the issue of robustly estimating the epipolar geometry [206, 207, 223], which sparked the popularity of the RANSAC algorithm [70] and incited the development of extending the idea to three and four view geometry [63, 87, 93, 177, 210].

However, with the myriad of work within the field, the above consideration are inevitably broad generalizations, and the citations somewhat haphazard. For a more just survey of the field the reader is referred to [89].

This thesis contributions to multiple view geometry is primarily the work on how the factorization methods can be made robust towards outliers and erroneous tracked features. This is described in Chapter 8. This spawned the development of a new numerical algorithm for weighted subspace estimation, which is found in Chapter 9. The merits of this work is that it allows the factorization algorithm to function with the shortcomings of the feature tracking, and as such allowing for a more automated approach.

In Chapter 10, work is presented introducing the idea of using prior shape knowledge – here planar structures – to regularize the structure from motion estimation. This has the effect that even with poor data good results can be obtained, assuming that the prior is true.

As for the immediate challenges of multiple view geometry, it is always hard to predict what thoughts will be on the agenda tomorrow. I, however, think that it is fair to say that most of the work in the past decade has focused on developing basic methods for 3D reconstruction, and hence taking a rather theoretical approach. As such a period with a more experimental approach is called for, addressing the questions of what approach(es) work best on real and and different data sets of considerable size, and uncovering what new issues arise in these cases. Along the same lines, I am unaware of a *major* study investigating the accuracy of structure from motion. Along the lines of this forecast – and in all due fairness – there is work being done constructing full structure from motion systems, and uncovering the issues when testing them on real data, e.g. [21, 71, 144, 155, 157, 161, 165, 176, 226].

As a courtesy to readers less familiar with multiple view geometry, the rest of this chapter is formed as an introduction to multiple view geometry, as basic knowledge in this field is required for most of this thesis. As mentioned before, excellent books on the subject exist and it would be futile to try and better this work here.

4.1 The Basic Camera Model

Multiple view geometry takes its offset in images of the world. Hence a mathematical model of a camera is an obvious way to begin. More abstractly, this is our observation model, since it is through the camera that our observation of the world are formed. Normal cameras can be approximated well by the model shown in Figure 4.1. As depicted, it is beneficial to introduce a coordinate system with origo at the camera center, with the x–y plane parallel to the image plane and the z-axis along the optical axis, and scaling the system such that the distance from the camera center to the image plane is 1.

The camera model projected into the x–z plane is seen in Figure 4.2. From this figure it

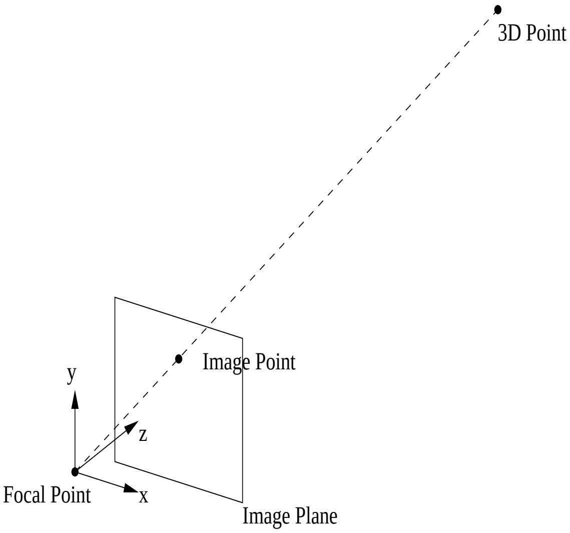


Figure 4.1: Model of a camera.

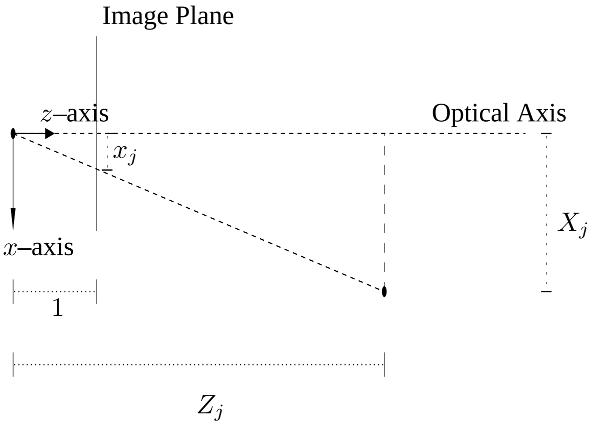


Figure 4.2: Figure 4.1 projected into the x–z plane.

can be seen that

$$x_j = \frac{x_j}{1} = \frac{X_j}{Z_j} \ \ , \tag{4.1}$$

and likewise in the y–z plane yielding the combined camera model:

$$\left[\begin{array}{c} x_j \\ y_j \end{array} \right] = \frac{1}{Z_j} \left[\begin{array}{c} X_j \\ Y_j \end{array} \right] \ \ . \tag{4.2}$$

This model (4.2) assumes that the $z-$ axis is identical to the optical axis, and that the focal length has been set to unity. Obtaining this is called an internal calibration of the camera. This is obtained by translating the image coordinates such that the optical axis passes through the origo, and scaling the coordinate system such that the focal lengths are eliminated. For further information the reader is referred to [34], which also covers subjects such as radial distortion and skewness.

4.2 Homogeneous Coordinates

The homogeneous representation of 2D points is made by adding a third ’dimension’ to the points which is an arbitrary scale factor. The 2D image point $\begin{bmatrix} x & y \end{bmatrix}^T$ is written as $\begin{bmatrix} sx & sy & s \end{bmatrix}^T$, where s is an arbitrary scale factor. As such the 2D point $\begin{bmatrix} 2 & 3 \end{bmatrix}^T$ is represented as both:

$$\left[\begin{array}{c} 1 \cdot 2 \\ 1 \cdot 3 \\ 1 \end{array} \right] = \left[\begin{array}{c} 2 \\ 3 \\ 1 \end{array} \right] \quad \text{and} \quad \left[\begin{array}{c} 3 \cdot 2 \\ 3 \cdot 3 \\ 3 \end{array} \right] = \left[\begin{array}{c} 6 \\ 9 \\ 3 \end{array} \right] \ \ .$$

The reason for using this seemingly silly representation, where a given 2D coordinate has many equivalent representations is, that it makes the representation of certain geometrical entities simpler.

For example a line consist of all points, $\begin{bmatrix} x & y \end{bmatrix}^T$, for which it holds true that:

$$a \cdot x + b \cdot y + c = 0 \Leftrightarrow s \cdot a \cdot x + s \cdot b \cdot y + s \cdot c = 0 \ \ ,$$

where s is an arbitrary scale factor. Since a line is invariant up to a scale factor, it can be represented as an inner product in homogeneous coordinates:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}^T \begin{bmatrix} s \ \cdot x \\ s \cdot y \\ s \end{bmatrix} = l^T p = 0 \ \ ,$$

where l is the vector associated with the line and p is the homogeneous coordinate. As such, with the aid of homogeneous coordinates, a line can be expressed as an inner product.

Another example is in the expression of the camera model (4.2)

$$s_j \cdot p_j = \begin{bmatrix} s_j \cdot x_j \\ s_j \cdot y_j \\ s_j \end{bmatrix} = \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} \quad , \tag{4.3}$$

where it is noted that s_j is set equal to Z_j

4.3 The Camera Model Revisited

As mentioned, the camera model in (4.2) and (4.3) is only valid if the whole world is viewed in a coordinate system which is scaled according to the focal length and translated such that the origo of the image coordinate system is located on the optical axis. If this coordinate system is not scaled and translated, this can be incorporated into (4.3) as follows:

$$\begin{bmatrix} s_j \cdot x_j \\ s_j \cdot y_j \\ s_j \end{bmatrix} = \begin{bmatrix} f & 0 & \Delta_x \\ 0 & f & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} \quad , \tag{4.4}$$

where f is the focal length and (Δ_x, Δ_y) is the coordinate of the optical axis in the image coordinate system, equaling the translation. The matrix is often referred to as the calibration matrix, and its entities the internal parameters.

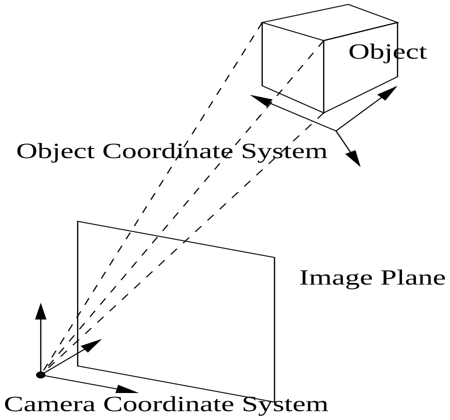


Figure 4.3: The camera and object coordinate systems do often not coincide.

Often the world is not aligned with the camera coordinate system, which will be inherently true with more than one camera, see Figure 4.3. As such the world coordinates have to be - temporarily - transformed into the camera coordinate system. This transformation will in general consist of a rotation $R_{3\times 3}$ and a translation $t_{3\times 1}$, transforming (4.4) into:

$$\begin{bmatrix} s_j \cdot x_j \\ s_j \cdot y_j \\ s_j \end{bmatrix} = \begin{bmatrix} f & 0 & \Delta_x \\ 0 & f & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \left(R \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix} + t \right) \quad . \tag{4.5}$$

Denoting the 3D point $\begin{bmatrix} X_j & Y_j & Z_j & 1 \end{bmatrix}^T$ in homogeneous coordinates – corresponding to 3D space – equation (4.5) can be written as:

$$\begin{bmatrix} s_j \cdot x_j \\ s_j \cdot y_j \\ s_j \end{bmatrix} = \begin{bmatrix} f & 0 & \Delta_x \\ 0 & f & \Delta_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix} \quad . \tag{4.6}$$

This is also the final camera model presented here, which is also the most common model used and is often referred to as the perspective camera model. It is noted, that sometimes the calibration matrix is omitted, because it is assumed that the image points have been altered accordingly.

4.4 Epipolar Geometry

After the perspective camera model one of the most fundamental concepts of multiple view geometry is the relations there exists between two projections or images of an object. This relationship is denoted the epipolar geometry, see e.g. [125]. This relationship is captured by the essential matrix or fundamental matrix depending on whether the calibration matrix is known or not.

4.4.1 The Fundamental Matrix

The fundamental matrix represents a relationship between two images with known relative position, concerning the location of a feature in both images. This relationship, as seen in Figure 4.4, is that the projection of a 3D feature (with unknown position) onto a 2D image plane (image 1) restricts the location of this 3D feature to a line in 3D, which again can be projected into another image plane (image 2). Hence a 2D feature in image 2 corresponding to a given 2D feature in image 1, is restricted to a line. This line is called the epipolar line of the 2D feature in image 1. This line is naturally only a half line, in that the 3D feature cannot be located behind the camera.

This relationship is expressed by the fundamental matrix. This can be derived by denoting the 3D point by P and its projection into image j by p_j, hence the - assumed - perspective camera model can be written as (see (4.5) for comparison):

$$s_j p_j = s_j \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} = A \left(R \cdot P_j + t \right) \quad , \tag{4.7}$$

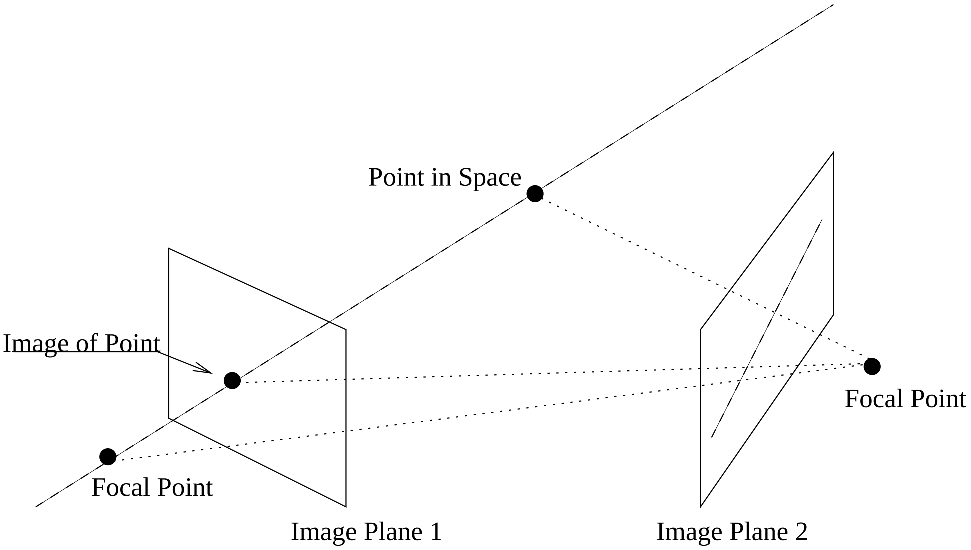


Figure 4.4: A geometric relation between two images.

where A represents the internal camera parameters (focal length, translation, skew etc.), R and t the external parameters, incorporating the rotation and translation from the object coordinate system to the camera coordinate system. Note that the 2D projection is formulated in 3D with a fixed z-coordinate representing its location on the image plane.

Given two images with projections of the same 3D feature, the object coordinate system can be aligned with the coordinate system of the first image, hence the observation models can be written as:

$$s_1 p_1 = A_1 P \qquad s_2 p_2 = A_2(RP + t) \ ,$$

where $R = R_2 R_1^T$ and $t = t_2 - R_2 R_1^T t_1$ relative to (4.7). Assuming – reasonably – that the $A's$ can be inverted, combining these two equations yields:

$$P = s_1 A_1^{-1} p_1 \ \Rightarrow \ p_2 = s A_2 R A_1^{-1} p_1 + A_2 t \frac{1}{s_2} \ ,$$

which is seen to be the epipolar line in image 2, where s is defined as $s = \frac{s_1}{s_2}$. This can be modified to (× denotes the cross product):

$$
\begin{aligned}
&A_2^{-1}p_2 = sRA_1^{-1}p_1 + t\frac{1}{s_2} \;\Rightarrow \\
&t \times A_2^{-1}p_2 = t \times sRA_1^{-1}p_1 \Rightarrow \\
&0 = (t \times A_2^{-1}p_2) \times (t \times sRA_1^{-1}p_1) \Rightarrow \\
&0 = (t \times A_2^{-1}p_2) \times (t \times RA_1^{-1}p_1) = \\
&A_2^{-1}p_2(t^T(t \times RA_1^{-1}p_1)) - t((A_2^{-1}p_2)^T(t \times RA_1^{-1}p_1)) = \\
&-t((A_2^{-1}p_2)^T(t \times RA_1^{-1}p_1)) = \\
&-t(p_2^T A_2^{-T} TRA_1^{-1}p_1) \Rightarrow \\
&t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \;\text{ or }\; p_2^T A_2^{-T} TRA_1^{-1}p_1 = 0 \;\;,
\end{aligned}
$$

here T is the matrix defining the cross product with t, as in $\forall x \; Tx = t \times x$ – see Section 4.8. If $t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$ then T will be a 3 by 3 matrix with all zeroes, implying:

$$p_2^T A_2^{-T} TRA_1^{-1}p_1 = 0 \;\; .$$

Thus defining the fundamental matrix as $F = A_2^{-T} TRA_1^{-1}$ and giving the relation:

$$p_2^T F p_1 = 0 \;\; . \tag{4.8}$$

It is seen, that the 3 by 3 fundamental matrix can be parametized by only 7 parameters. First, it does not have full rank, since T does not have full rank, and as such $det(F) = 0$. Secondly, it is only defined up to a scale, seen by multiplying (4.8) by an arbitrary scalar. It is seen that if t is non–zero, then the rank of T is 2 and the rank of R and A is 3 or full, so the rank of F is 2.

This entity can also be derived geometrically, by viewing the relation as seen in Figure 4.5. Here it is seen, that the point $A_2^{-1}p_2$ must be located in the plane spanned by the vector connecting the two epipoles, t, and the vector connecting the epipole in image 2 and the 3D features projection in image 1, $RA_1^T p_1 + t$.

Mathematically this can be expressed as:

$$(A_2^{-1}p_2)^T \left(t \times (RA_1^T p_1 + t)\right) = 0 \;\; ,$$

where $t \times (RA_1^T p_1 + t)$ is the normal vector to the plane. This can be reformulated as:

$$
\begin{aligned}
(A_2^{-1}p_2)^T \left(t \times (RA_1^T p_1 + t)\right) &= \\
(A_2^{-1}p_2)^T (TRA_1^T p_1) \;=\; p_2^T A_2^{-T} TRA_1^{-1}p_1 &= 0 \;\; ,
\end{aligned}
$$

yielding the formulation of the fundamental matrix once again.

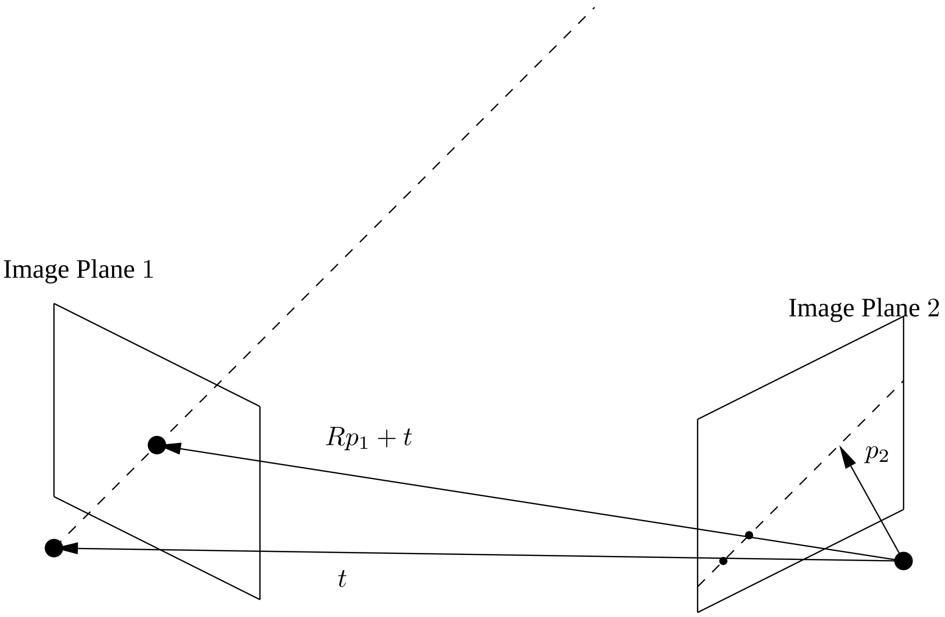


Figure 4.5: A geometric relation between two images.

4.4.2 The Essential Matrix

The fundamental matrix is closely related to another relation called the essential matrix, E. This is the relation among two images if they have been normalized. That is that the 2D features have been altered such that the A's are identity matrices. Hence the relationship of the fundamental matrix becomes:

$$0 = p_2^T T R p_1 = p_2^T E p_1 \quad , \tag{4.9}$$

where E denotes the essential matrix. As such the relation between the essential and the fundamental matrices is:

$$A_2^{-T} E A_1^{-1} = F \quad . \tag{4.10}$$

This relation, which temporally precedes the concept of the fundamental matrix, and some of its central properties are presented in [100]. The reader is referred to [89, 128] for a review.

One important property of E is that it has two singular values which are equal and one which is zero. That is, a singular value decomposition (SVD) of E can be written as:

$$E = U \Sigma V^T \quad ,$$

where

$$\Sigma = \begin{bmatrix} \sigma & 0 & 0 \\ 0 & \sigma & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad .$$

This is a necessary and sufficient condition for a 3×3 matrix to be an essential matrix. For a proof see [100].

4.5 Estimating the Fundamental Matrix

From the above, it is clear that the fundamental matrix can be calculated from the camera if the internal, A_i, and external, R_i, t_i, camera parameters are known. These are however not always at hand, and as such achieving it by other means becomes interesting. A useful way of doing this, is by estimating the fundamental matrix from point correspondences (p_1, p_2). This is an extensive subject, and an overview is presented here. The reader is referred to [89, 222] for a thorough and complete presentation.

Denote the fundamental matrix by its elements:

$$F = \left[\begin{array}{ccc} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{array} \right] \quad , \tag{4.11}$$

then it is noted, that a corresponding point pair, (p_1, p_2), which per definition satisfies the epipolar constraint, supplies a linear constraint on the elements of F. I.e.

$$\begin{aligned} 0 = p_2^T F p_1 = & \qquad (4.12) \\ \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix} \left[\begin{array}{ccc} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{array} \right] \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = & \\ \begin{bmatrix} x_2x_1, x_2y_1, x_2, y_2x_1, y_2y_1, y_2, x_1, y_1, 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = & \\ b^T \mathbf{f} & \end{aligned}$$

where $\mathbf{f}$ is the vector containing the elements of F, and b is the linear constraint corresponding to an image pair.

As mentioned above, the fundamental matrix has 7 degrees of freedom, and hence at least 7 point pairs and corresponding constrains, $b^T \mathbf{f} = 0$, are needed.

4.5.1 The Eight Point Algorithm

The two constraints on a general 3×3 matrix, with 9 degrees of freedom, that makes it a fundamental matrix, is the indifference of scale and $det(F) = 0$. The later constraint is not linear in the elements of F, and hence is cumbersome to enforce. Due to this, this constraint, $det(F) = 0$, is sometimes ignored, leaving 8 free parameters to be determined, thus yielding the *eight point algorithm*, in that 8 constraints (= point pairs) are needed.

This algorithm is easy to implement, because it is linear, but it does not yield optimal results, among others, because a necessary constraint is not enforced, for better algorithms refer to [89]. This is not to say, that the results are useless, since this is far from the case, and the eight point algorithm is often used to initialize the more sophisticated approaches.

More specifically the eight point algorithm works by having eight or more constraints , b_j, which are arranged in a matrix $B = [b_1, \ldots b_n]^T, \ n \geq 8$. Then from (4.12) it is seen that:

$$B\mathbf{f} = 0 \ . \tag{4.13}$$

In the presence of noise this can not be achieved in general, and $\mathbf{f}$, and hence F, is found via:

$$\min_{\mathbf{f}} ||B\mathbf{f}||_2^2 = \min_{\mathbf{f}} \ \mathbf{f}^T B^T B\mathbf{f}, \quad ||\mathbf{f}||_2^2 = 1 \ . \tag{4.14}$$

The extra condition $||\mathbf{f}||_2^2 = 1$ is added to avoid the trivial – and useless – null solution. Numerically (4.14) can be solved by finding the eigen vector of $B^T B$ with the smallest eigen value. In practice however this is done via singular value decomposition, SVD.

Numerical Aspects

In conjunction with the eight point algorithm there are some aspects, that drastically will improve the result. The first one is to use far more than 8 point pairs, in order to reduce noise in the location of the points. A recommended number is of course hard to give, and is highly dependent on the image noise, but 15-30 usually works well in the presence of modest noise on the features.

In order to achieve numerical stability it is *highly* recommended that the point locations are normalized before the eight point algorithm. The result of this normalization should be, that for each image the mean value should be $(0, 0)$, and that the mean distance from the origin – $(0, 0)$ – to the points is $\sqrt{2}$. See [90] for further discussion.

4.6 Reconstructing 3D Structure and Motion

Moving further than the two view geometry, to multiple views of the same rigid body. Then the main task becomes estimating the structure and motion. This is mainly done by finding the structure P_j and motion R_i, t_i that best fit the data p_j, given the perspective camera model (4.6). More formally this can be expressed as; given $i \in [1 \ldots m]$ views of $j \in [1 \ldots n]$ feature points describing a rigid body, find the P_j, R_i and t_i minimize:

$$\min_{P_j, R_i, t_i} \sum_{i=1}^{m} \sum_{j=1}^{n} ||s_{ij} p_{ij} - A_i(R_i P_j + t_i)||^2 \ \ , \tag{4.15}$$

where the calibration matrix A_i is usually given.

The solution to (4.15) is easily obtained using a non–linear optimization algorithm, e.g. the Marquart method [2]. In order to achieve a good result, an initialization, or start guess is required. For more information on solving (4.15) refer to [185, 211].

However, it is only possible to solve the structure and motion problem up to an Euclidean similarity transformation, in that the cameras have no way of knowing the origo or the orientation of the 'true' coordinate system. Secondly the whole system is only known up to a scale. A good example of this is the "Star Wars" movie where the "death star" looks to be the size of a small planet. I hope I do not break any illusions here, but it is not. The background for this optical trick is, that cameras only 'know' directions and not size.

A note of warning, the ambiguity of the Euclidean similarity transformation has implications when optimizing (4.15). In that the fixing of this ambiguity also effects the variance structure of the structure and motion estimate. For an introduction to this issue the reader is referred to [43, 44], and for a more thorough treatment see [14, 132, 193].

4.6.1 Factorization Algorithms

Factorization algorithms supply an approximate or initial guess for the non–linear solution to (4.15). These methods were first proposed by [204], and work by linearizing the perspective camera model, i.e. it is assumed, that (4.6) can be written as:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = M_i P_j + t_i \quad , \tag{4.16}$$

where M_i is a 2 by 3 matrix

$$M_i = \begin{bmatrix} M_i^x \\ M_i^y \end{bmatrix} \quad . \tag{4.17}$$

Assuming this camera model (as will be done in the rest of this subsection) the t_i are temporarily removed from the equation by forming:

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} - \frac{1}{n} \sum_{j=1}^{n} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \quad .$$

Hereby it can be assumed that

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = M_i P_j \quad ,$$

which is equivalent to assuming, that the origo of the world coordinate system is at the center of mass of the rigid object.

The mean corrected observations can then be combined to:

$$\mathbf{W} = \mathbf{MS} \quad , \tag{4.18}$$

[2] The reader is referred to `www.imm.dtu.dk/˜hbn` for further details in non–linear optimization

where

$$\mathbf{W} = \begin{bmatrix} \tilde{x}_{11} & \cdots & \tilde{x}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{x}_{m1} & \cdots & \tilde{x}_{mn} \\ \tilde{y}_{11} & \cdots & \tilde{y}_{1n} \\ \vdots & \ddots & \vdots \\ \tilde{y}_{m1} & \cdots & \tilde{y}_{mn} \end{bmatrix} \quad ,$$

$$\mathbf{M} = \begin{bmatrix} M_1^x \\ \vdots \\ M_m^x \\ M_1^y \\ \vdots \\ M_m^y \end{bmatrix}$$

and

$$\mathbf{S} = \begin{bmatrix} P_1 & \cdots & P_n \end{bmatrix} \quad .$$

Taking the Singular value decomposition of $\mathbf{W}$ yields:

$$\mathbf{W} = \mathbf{U\Sigma V}^T \quad ,$$

where it can be shown that the least squares solution fit is equivalent to:

$$\mathbf{M} = \tilde{\mathbf{U}}\sqrt{\tilde{\mathbf{\Sigma}}} \quad , \quad \mathbf{S} = \sqrt{\tilde{\mathbf{\Sigma}}}\tilde{\mathbf{V}}^T \quad , \tag{4.19}$$

where $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ are the three first rows of $\mathbf{U}$ and $\mathbf{V}$ respectively, and $\tilde{\mathbf{\Sigma}}$ is the top–left 3 by 3 submatrix of $\mathbf{\Sigma}$.

This solution (4.19) is not unique, in that for any invertible 3 by 3 matrix $\mathbf{B}$, it holds that

$$\mathbf{W} = \mathbf{MS} = \mathbf{MBB}^{-1}\mathbf{S} \quad .$$

This $\mathbf{B}$ is usually found up to scale by requiring that the rows of M_iB be as orthonormal as possible. This then gives a solution for the structure $\mathbf{B}^{-1}\mathbf{S}$ and an approximation for the motion.

For a more thorough review of factorization methods and or how to retrieve the motion parameters refer to [111].

4.7 Constraint Feature Tracking

As mentioned in the previous chapter, the two view geometry is typically used to regularize the feature tracking. That is, if it is assumed that a rigid body is depicted, then the epipolar constrains hold between any two images. This is thus a constraint on the constellation of tracked features. The problem is, however, that the epipolar geometry is not known in general, and tracked features are needed to estimate it.

The great solution of [207, 223], is to robustly fit the epipolar geometry to partly erroneous tracked features. Hereby outlier detection of the tracked features is performed, by regularizing with the epipolar geometry. This again induces a fundamental improvement of the quality of the tracked features on many real data–sets. The robust statistical method of [207] is RANSAC [70], which has obtained a high degree of popularity. The general approach described here has also been extended to three– and N–view relationships, but a treatment of these issues is beyond the scope of this introduction, cf. [89].

4.8 The Cross Product as an Operator

As an appendix to this introduction it is shown that the cross product of a given vector with any other, can be formulated as a matrix multiplication. That is, a matrix A can be constructed such that $a \times b = A \cdot b$, where $\times$ denotes the cross product. This is seen by:

$$a \times b = \left[\begin{array}{c} a_1 \\ a_2 \\ a_3 \end{array}\right] \times \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array}\right] = \left[\begin{array}{c} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{array}\right] =$$

$$\left[\begin{array}{ccc} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{array}\right] \cdot \left[\begin{array}{c} b_1 \\ b_2 \\ b_3 \end{array}\right] = A \cdot b \ .$$

Hereby defining A. This skew symmetric matrix A can also be seen as a linear operator.

CHAPTER 5

Multiple View Stereo

As seen in Chapter 2, following the feature tracking and the structure from motion of the extracted points the surface of the object should be estimated. Compared to feature tracking and multiple view geometry, there are many more open questions here, and as such a ’standard’ approach can not be offered. The most common surface reconstruction approaches for structure from motion will be considered and discussed and the contribution of this thesis will be set in perspective. It should be noted that, the aim is to give an overview and not an in depth survey of the literature.

5.1 Approaches to Structure Estimation

There are many cues to the 3D structure of a depicted object in the respective image(s) hereof. Apart from stereo, where the structure can be inferred by triangulation, we humans are also able to infer much of the 3D structure from a single image. Some of these single view visual cues have also been applied in computer vision for structure estimation, e.g. shape from shading cf. e.g. [28, 122, 154, 221] shape from texture cf. e.g. [114, 191, 214] and recently some very interesting work on photometric stereo has been presented in [92]. The latter requires a set of training images.

In the structure from motion setting, stereo or multiple view stereo is the obvious choice. As far as I know, stereo is also the single visual cue that gives the best result in general. But if stereo were not to be used there was no reason for structure from motion. Hence this is what will be considered here and what has been considered in the literature in relation to structure from motion. However, since surface estimation is a hard problem all visual cues should ideally be utilized, as previously proposed in similar settings e.g. [37, 54, 76, 149].

An area of research closely related to surface estimation is new view synthesis where some of the approaches, e.g. [11, 79], only model the plenoptic function, i.e. the intensity, color and direction of light rays at all points. These approaches however do not explicitly model the surface, and as such are not applicable for a structure from motion approach as considered here.

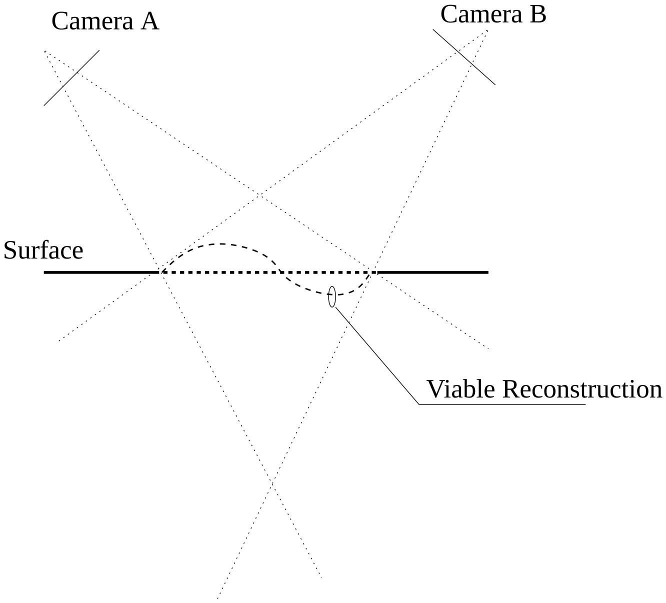


Figure 5.1: Consider the reconstruction of the dashed part of the surface, assumed to have uniform color, i.e. every part of it looks the same in both Cameras A and B. Then it is seen that any surface located within the depicted rectangle will produce the exact same image in the two cameras, and hence be just as good a fit to the data (images) as the true surface. This is seen to be an ambiguity of the surface reconstruction problem.

5.2 The General Stereo Problem

Stereo and multiple view stereo[1], *the estimation of a depicted surface from known cameras*, is a hard problem. In essence it resembles the feature tracking problem, described in Chapter 3, and can be viewed as finding the correspondences for all points on the surface. To see this, note that *if* the correspondences of all points on the surface are known, i.e. all pixels in the image, then it would be straight forward to estimate the surface shape via triangulation.

This relation to feature tracking implies, that the aperture problem has taken on a new form, as pointed out in the insightful work of Kutulakos and Seitz [119, 118]. That is, as

[1] stereo with more than two views

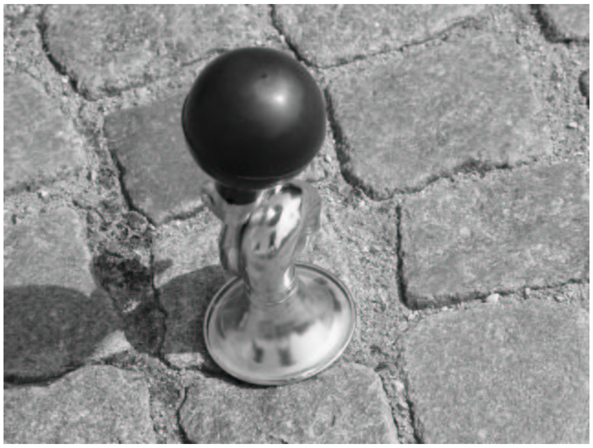


Figure 5.2: A typical every day object, not adhering to the Lambertian assumption partly due to the reflective surface.

illustrated in Figure 5.1, that there is an ambiguity of the surface shape. The largest possible surface consistent with the images is know as the photo hull. For a great overview of photo hulls in particular and surface estimation in general refer to [183].

There is jet another challenge of surface estimation it shares with feature tracking. That is that most algorithms assume that the same entity looks similar in all images. This is very closely related to the Lambertian surface assumption, i.e. that a point on the surface irradiates the same light in all directions (intensity and cromasity). This surface assumption is made by most stereo algorithms, although there has been some work on getting around this – see below. However, many or most real objects do not adhere to this assumption, causing problems for most stereo approaches, see e.g. Figure 5.2.

The ambiguity of the stereo problem, and the general difficulty of the problem, e.g. specularities, ushers in a need for regularization in stereo algorithms, whereby prior knowledge or assumptions can be used to aid the process and make it well defined. The stereo problem has spawned a lot of various algorithms. In the following the two – by far – most common approaches to stereo in the structure from motion setting are presented (Section 5.3 and Section 5.4).

5.3 Two View Stereo

Two view stereo – or just stereo – is one of the more studied problems within computer vision, a reason for it's popularity is that it so resembles the human visual system, i.e. two eyes, and that two is the minimal amount of views needed for stereo. A good overview and dissection of stereo algorithms is found in the work of Scharstein and Szeliski [170], to which the interested reader is referred. However a sample approach is presented in the following to convey the general ideas.

5.3.1 A Popular Approach to Stereo

A popular approach to stereo, which is the one utilized in [157] and the one I implemented for insight, is centered around the algorithm of Cox et al. [47]. This has the advantage that some more practical optimization/ implementation issues are considered in [61, 62, 225]. As an overview a brief description of a setup using this stereo algorithm is given here.

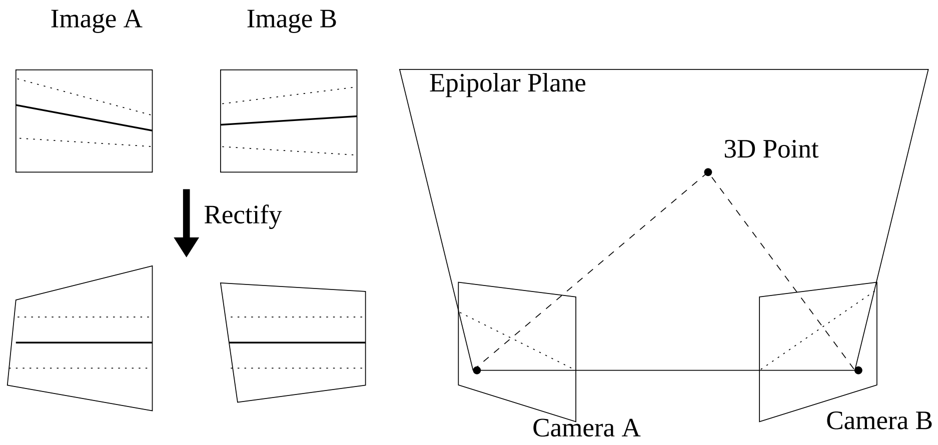


Figure 5.3: As a preprocessing of the images they are rectified – via a warp – such that corresponding epipolar lines are paired in scan lines. To the right it is seen, that a plane going through the two optical centers of the cameras, intersect the image planes in corresponding epipolar lines. That is, any 3D point located on this epipolar plane will be depicted to these intersecting lines. Hence these lines can be paired as described.

At the outset, the two images are rectified, as illustrated in Figure 5.3, such that the scan lines of each of the two images form pairs. These pairs are such that for all pixels the correspondence is found in the paired scan line and vice versa. The theory behind why this can be done, is the epipolar geometry described in Chapter 4. Methods for doing this are presented in [89] and for more general camera configurations in [158].

Following the rectification of the images, it is seen that, the stereo problem can be reduced to just matching paired lines. Cox et al. [47] propose a method for doing this using dynamic programming. This is however under the assumption, that there is a monotonic ordering of the correspondences, i.e. if pixel i in Image A is matched to pixel j in Image B then pixel $i + 1$ can not match to pixel $j - 1$.

The output of this algorithm is a disparity map, which is how much a given pixel in Image A should be moved along the scan line to find its corresponding pixel. In other words the disparity equals the optical flow, where only a scalar is needed, since the direction is given, i.e. along the scan line. As in the correspondence problem the measure of similarity of matching cost can vary, but usually a correlation[2] or covariance measure is used.

Once the disparities, and hence the correspondences, have been found, it is ’straight forward’ to calculate the depth of the depicted entities and hence the 3D surface.[3] As an illustration of the described setup, an example is presented in Figure 5.4. It is run on two images from a longer sequence from which the camera calibration is achieved via structure from motion.

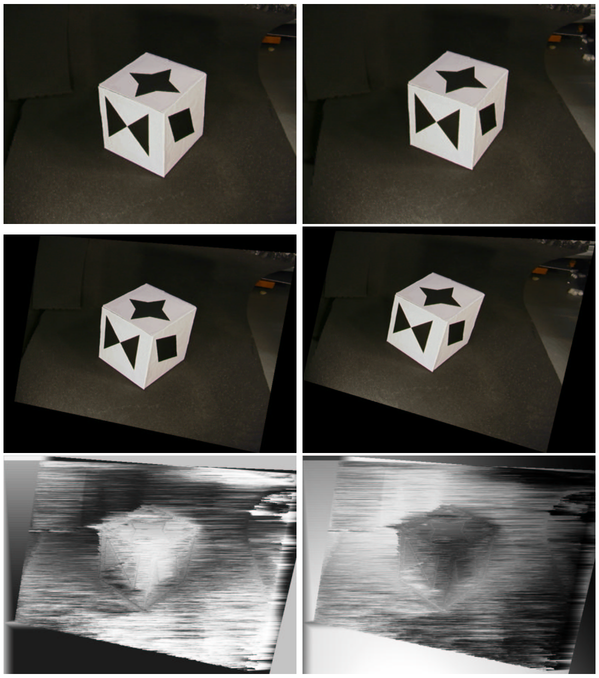


Figure 5.4: An example of the described two view stereo setup: Original image pair (top), rectified image pair (middle) and diparity map (lower left) depth map, i.e. distance from the first camera to the surface (lower right). The reconstruction quality is quite poor due to noise in that camera calibration.

5.3.2 Extension to Multiple Views

In relation to the structure from motion problem where more than two views typically exist, there has been some work on directly extending the work on two view stereo to multiple

[2] normalized covariance.

[3] A word of warning, a common error in estimating the depth or 3D position is to find the 3D lines associated with the observed pixels, and then find the 3D point which is closest to them. With noise, a perfect fit is unlikely to exists and as such this procedure does not minimize the correct error, in that it implicitly assumes a linearized camera model. Instead it is necessary to make a miniature bundle adjustment, cf. Chapter 4.

cameras. One approach has been viewing the results from a two view stereo algorithm as a single measurement and then finding the best fit to these, e.g. [145, 157, 166, 197], i.e. to run a two view stereo algorithm on some or all pairs of images, getting a (possible partial) estimation of the surface to be estimated, and then merge these surfaces to one estimate. The latter could e.g. be done via the techniques of [36, 188, 213], although the above cited methods employ other methods.

Another approach to extending to multiple views directly address the *depth–baseline dilemma* described below, e.g. [47]. This is done by finding the disparity between two images with a depth to baseline ratio closest to one, by matching across a sequence of intermediate images. The surface is then estimated based on the image pair with the depth to baseline ratio closest to one.

One of the main problems with the extensions of two view stereo to multiple views, in my opinion, is related to error minimization, or what Hartley and Zisserman [89] refer to as obtaining a gold standard solution. That is, since the image observations are noisy, the surface estimation approach should also take this uncertainty into account in a statistically sound way. It is noted, that a straight forward averaging of the two view stereo estimates is suboptimal. This is so, since the uncertainty varies, both due to the different baseline to depth ratios but also due to occlusions. In the latter case some parts of the two view surface estimate can have an infinite uncertainty.

A possible way to handle the uncertainty of the different two view estimates, is to propagate the uncertainty of the individual two view stereo estimates to the merging algorithm. But this will make the formulation of the underlying object function minimized less clear. Other points of criticism of the combined two view stereo methods are found in [41], where 3 criteria for “true image matching techniques” are proposed.

Depth–Baseline Dilemma

When designing a two view stereo set up there is a dilemma in choosing the baseline to dept ratio, cf. Figure 5.5. On one side, as discussed in Chapter 3, images should look similar in order for matching algorithms to work well, which will typically be the case if the baseline is small relative to the depth. On the other side, as illustrated in Figure 5.6, a baseline to depth ratio close to one will make the measurements much less sensitive to noise, giving a more accurate reconstruction. A rule of thumb is that the depth to baseline ratio should be between $\frac{1}{3}$ and 3 to get reliable results [34].

5.4 Direct Surface Optimization

A different and later approach to surface estimation, compared to generalized two view stereo, is what I choose to term "direct surface optimization". That is the surface estimation is formulated as a data fitting problem where the model, i.e. the surface, is optimized such that it gives the best possible explanation of the data, i.e. the images. This differs from two view stereo where the model is crystalized out after several estimation processes, and uses a non–object centered representation. The variability between direct surface opti-

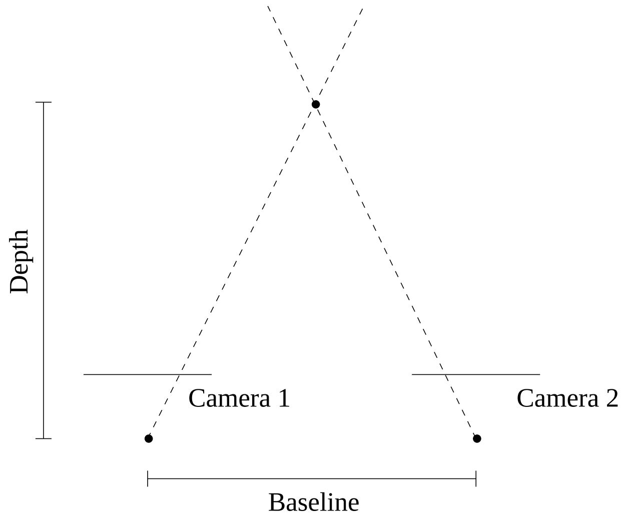


Figure 5.5: An illustration of the baseline and depth of two cameras relative to an object or point.

mization methods is mainly in relation to the surface representation, i.e. triangular mesh[4], voxelization of 3D space or as a signed distance field, and in the objective function used. The objective function, is usually correlation or covariance, but in principal depends on the assumed surface and observation model. The surface model is usually assumed Lambertian.

5.4.1 Representations

The **mesh** representation of surfaces is the standard within computer graphics [74, 216], and has among others been used for surface estimation in [76, 103, 137, 164]. Isidoro and Sclaroff [103], however, only use sporadic stochastic samples off the surface to evaluate the surface fit. Morris and Kanade [137] propose a method aimed directly at structure and motion, by triangulating the 3D points found in structure from motion as best to explain the images. This is in analogy to Delauney triangulation, see e.g. [83], except that the objective function is changed from encouraging well formed triangles to best fitting the image data. One of the great advantages of using triangular meshes is their close relation to computer graphics, whereby the graphics hardware can be used for acceleration, it also eases visualization. Actually the signed distance field and voxel grid are usually converted to meshes – via marching cubes [126] – when they should be visualized.

The representation of a surface via a **signed distance field** is an implicit one. In that the surface is represented by a discrete field where each element or voxel contains the signed distance to the surface. The sign denotes whether the voxel is inside or outside of the volume

[4] will be referred to simply as "mesh"

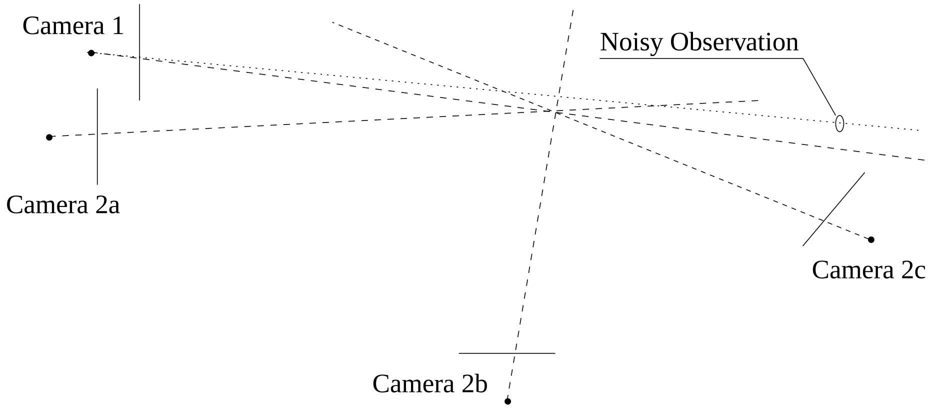


Figure 5.6: When estimating the position of a 3D point between Camera 1 and one of the Camera 2's it is seen that using Camera 2b will give the most accurate results. This is due to the angle between the intersecting ray, which are directly related to the baseline to depth ratio.

boarded by the surface. Hence the surface is represented as the 0^{th}–level set of this distance field. This gives name to the popular methods for manipulating these surfaces, namely the level set method [146, 147, 174]. The level set framework is a partial differential equation (PDE) solver, and as such the objective function needs to be formulated as a such. Hence the Euler–Lagrange differential equation become handy. The use of the level set methods for surface estimation was first proposed by Faugeras and Keriven [64, 66] and has been received well by the computer vision community. It has among others been extended to non Lambertian surfaces by Jin et al. in [106, 105].

The **voxel grid** representation of a surface or the volume the surface encapsules is very similar to the signed distance field representation, in that it can be seen as a binary version thereof. The voxel grid is a discrete 3D grid where the voxel is 1 or present if inside the surface and 0 if outside. This corresponds to the sign of the signed distance field. The surface estimation techniques utilizing the voxel grid is the so called space carving cf. [60, 119, 118, 173, 183]. The analogy is that the technique starts with all the voxels present (set to 1) and then carves them away like a sculpture from a block of stone.

5.4.2 Comparison

The advantage of the signed distance field and voxel grid representation is, that they are highly flexible structures, seamlessly allowing large deformations of the surface including the change of topology. In regard to this, a mesh structure would typically have to be remeshed – i.e. a new set of triangle facets would have to be constructed – to represent the surface effectively, if deformations were to large.

As to whether the space carving or level set methods should be used, is mainly a question of regularization. Kutulakos and Seitz [118] argue that their method – space carving – has the advantage that it runs without priors, contrary to the level set method of Faugeras and Keriven [64, 66], which as a minimum has a smoothness prior implicit. The rationale behind the argument that a lack of prior is advantageous, is: a prior biases the estimate towards ones prior knowledge. On the other hand, the signed distance field representation is a much more natural framework for expressing many of the priors for surfaces, in that the level set method is intertwined with differential geometry. Concerning the ambiguity of the surface estimation problem, the space carving approach addresses this by reconstructing the photo hull, cf. Figure 5.1.

However, the choice of the photo hull as a solution can also be seen as sort of regularization, using a prior of "make the object as large as possible". So in other words regularization is needed, and making the object as large as possible does not seem to be the best model of objects, i.e. prior model, we can come up with. It is noted, that space carving methods applying other priors have been proposed e.g. [56, 184]. As a note, the usual prior in the level set frame work is a smoothness prior, formulated as a penalty of the total surface area. The PDE equivalent is penalizing the total area is the curvature flow.

An advantage of the direct surface optimization techniques described here is, that they have a very direct, and hence intuitive, incorporation of the modelling framework into the algorithms, making interpretation an construction easier. This includes a clear uncertainty model. Another advantage, compared to the two view approach, is that the surface or reflectance model can have a higher degree of freedom, in that there is more observations. In other words, in the two view stereo approach the reflectance model can only have one parameter, typically the diffuse color, or else the problem will become under constrained. With the more observations of the direct surface estimation approach more advanced reflectance models can be handled.

An issue with these direct methods – the level set method in particular [183] – is the computational time needed. From my experience with the level set method for surface estimation half a day is by no means uncommon[5]. However there has been some work to address this issue, by using using hierarchical scales or a multi–grid[6] framework [42, 117]. See also Chapter 11 discussed below.

5.5 Surface Models

For the most part of this chapter, it has been more or less ambiguous what is implied by a surface model. To be more formal, the surface model consists of a 3D surface shape and the reflectance properties of the individual points on the surface. The latter includes texture, i.e. color, but also specularities of the surface. The shape of the surface is what most of all the above mentioned techniques deal with. The likely underlying assumption is; that once the shape has been estimated the texture can be found via back projecting one or more images.

The reflectance model however has implications for the matching or objective function, and vice versa. This function quantifies how likely it is for two parts of the image to originate

[5] In a C++ framework.

[6] See e.g. [27]

from the same 3D point, and hence deals with how the same point on the surface looks from different directions, i.e. in different images. In most surface estimation algorithms some form of covariance score is used. This has the implication that a given point on the surface looks the same from all directions. This corresponds to the Lambertian reflectance model, and assumes that no reflections exist. This assumption far from always holds as illustrated in Figure 5.2.

There has been some work done on extending surface estimation beyond the Lambertian surface model, by Jin et al. [106, 105]. In [106] they handle reflectance by applying robust statistics and treating them as outliers – i.e. disregarding them. In [105] it a full framework for estimating reflective surfaces is proposed, by allowing a surface's appearance to span a 1D subspace. However, there still seems room for work to be done here.

Within the computer graphics literature, there is a well developed theory of surface reflections, and the lighting of the scene. Much of this will probably have to be taken into account when solving the structure from motion problem in the future. As a general reference see [74] and [58], but to mention a few: Ramamoorthi [162] has done some interesting work on formulating the reflectance of light of a surface in a convolution framework. This allows for much of the highly developed signal analysis framework for estimation to be employed. Also the work of Yu et al. [219] on estimating global illumination deserves mentioning.

5.6 Discussion

My evaluation of the presented methods, which are the ones I have seen used in conjunction with structure from motion, is that the direct surface optimization approaches are preferable at the present stage of development. The reason behind this is, that the object function is more intuitive, and it allows for more complex surface models. The latter I believe is an area of development which has to be included. However these methods are rather slow – especially the level sets.

As to whether a mesh or a grid representation is preferable, at present the best results come from methods based on the grid representation. This is partly due to the great flexibility of the grid based methods. But it might also be due to these having been highly popular recently, and as such the forefront of development has been done here. An advantage of mesh based approaches is, that this is the representation used in the computer graphics hardware – present on most computers today – allowing for a potential speedup by hardware usage.

The speed problem of the level set approach to stereo, can be addressed by using the faster mesh based methods as initialization. This is proposed in Chapter 11, where the speedup is considerable. Initializing the surface estimation via apparent contours (cf. [110]) has been proposed in [48], but it could also be interesting to apply the extended two view methods in this way. It is noted, that more data dependent regularization techniques were also briefly investigated in this work, i.e. Chapter 11.

As part of doing this, another contribution of this thesis, is proposing a new and simple method for converting meshes to signed distance fields, cf. Chapter 12. This is naturally required for applying the mesh based method as initialization for a signed distance field based method. This work also yielded some nice results about the angle weighted pseudo normal [190, 201]. The last contribution of this thesis, in the area of surface estimation, is presented in Chapter 13. Here a method for using specular reflections *and* estimated 3D feature points for surface estimation is proposed. Thus extending beyond the Lambertian model.

There are two main unsolved problems I believe will receive much attention in the nearer future. One is the extension to more elaborate reflectance models, as is already being done by Jin et al. [106, 105], such that more real life objects can be handled. The other is on the operational side, mainly speed, but also the development of more robust statistical techniques, and other things needed for a more efficient and reliable system.

5.7 Summary

Here an overview of possible ways of estimating the surface in a structure from motion setting has been presented and discussed, along side this thesis contribution and like paths for future development. However, as noted above this is not a complete survey of the literature so the selection of methods mentioned has been selective and somewhat haphazard. Keeping the subjective nature of the undertaking I believe to have conveyed an accurate picture of the field.

CHAPTER 6

Relaxing the Rigidity Constraint

In the previous part of this thesis, and in most of the work done on structure from motion, the assumption of the object considered being rigid is held. However, it is feasible to relax this constraint, as proposed by Bregler et al. [25] in 2000. This was done by proposing a factorization method for solving the non–rigid structure from motion problem, as the branch was dubbed. Here the structure of a non rigid rigid object is estimated from tracked features. There were however some slight miscalculations in this, as pointed out by the same group in [209], where a new approach was also presented. Simultaneously Brand [22] proposed another factorization algorithm, along side an approach to feature tracking [24, 23]. In 2002 Kahl and I considered how the estimation multiple view geometric procedure should be extended beyond the factorization approach. An extended version of this work is presented in Chapter 14. Following this Svenson, Kahl and I investigated extending multiple view stereo techniques to the non rigid case, cf. Chapter 15.

Previously, it had been proposed estimating non–rigid motion using a Kalman filter with physics based priors on the dynamics [133, 153]. However, the deforming objects need to be properly initialized.

6.1 Linear Subspace Constraint

On a more operational level the extension to non–rigid structure, in the above mentioned approaches, works by extending the model for the structure. This extension is to a linear subspace. That is, instead of representing 3D point j as vector:

$$P_j \quad ,$$

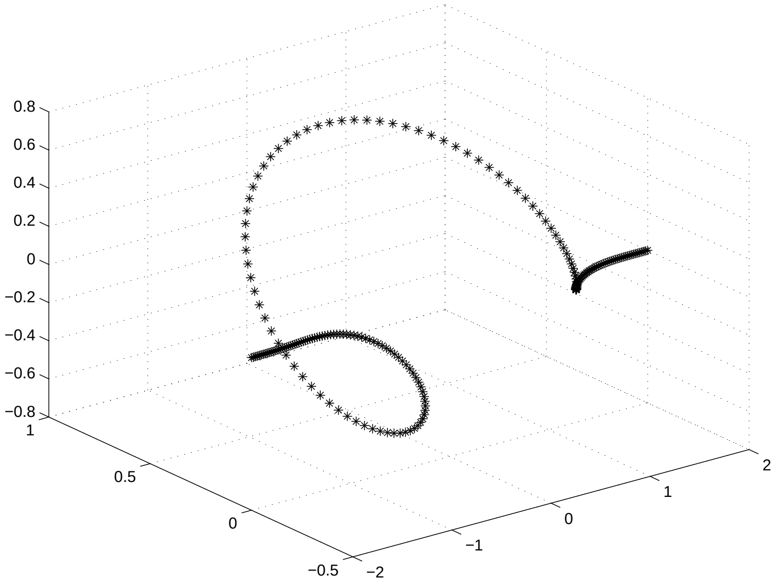


Figure 6.1: The depicted points will need a 3D–(sub)space if they should be represented effectivly in a linear subspace. It is, however,fair to say, that they are all located in a 1D–subspace, which is highly non–linear.

it is represented as a *linear* combination of r different vectors or modes, i.e.

$$\tilde{P_{ij}} = \sum_{k=1}^{r} \beta_{ik} P_{jk} \quad , \tag{6.1}$$

where $\tilde{P_{ij}}$ denotes the combined 3D point, the β_{ik} are scalars and the index i indicates the time instance or frame number.

Even though the extension to non–rigid structure induces a lot of degrees of freedom, there are typically still plenty of constraints left. Let $j \in \{1 \ldots m\}$, then it is seen the structure is a $3m$–dimensional variable. Usually $r << 3m$, r being the number of modes, so at the outset this does not render the problem under–constrained. However, these extra degrees of freedom induce some additional ambiguities in the solution, as described in Chapter 14.

The assumption, that the non–rigid structure is described well by a linear model is the well known and popular one used in Principal Component Analysis (PCA) framework [98, 151]. However, it is by no means clear that this is an effective representation of the model and captures the true underlying physical deformations, cf. [200]. As an illustrative example of this refer to Figure 6.1. So contrary to rigid structure from motion, where all the models describe the underlying physics very well, the added modelling framework is less exact in that regards. This more ’black box’ nature of the added framework is also illustrated by the number of modes, r, being unknown a priori. Hence the model order is also part of the estimation problem, and draws on the field of model selection, cf. e.g. [12, 91, 97, 134, 172].

6.2 Discussion

From a brief glance around our everyday environment, it is clear that we by no means live in a purely rigid world. As such, it is natural to extend the structure from motion framework in this direction, as has been commenced. As described above, the proposed schemes are less dependent on the underlying physics. This diminished dependence on physics also implies the need for a rigorous test of the approach, in order to test the validity and applicability of the modelling approach. But first the theoretical framework has to be more well understood.

In this regards it should be noted, that it is unlikely that a general physical dependent methods will appear in the nearer future, since computers are nowhere near capable of object recognition, without which the underlying physics is unknown.

CHAPTER 7

Discussion and Conclusion

As seen in from the preceding chapters, this thesis has contributed by addressing some of the subproblems within structure from motion. These contributions have mainly dealt with robust factorization approaches, relaxing the rigidity constrains, and considering alternative ways of solving the surface estimation problem. An exact description of these contribution is found in the following.

As for the future challenges of structure from motion. My guess is that in the nearer future two issues will be dominant. Firstly, the further development of multiple view stereo techniques, cf. Chapter 5. Secondly, the further integration of the various developed techniques, such that the *whole* system can be studied as has e.g. been done in [71, 144, 157, 161, 165, 226]. The latter will allow a better evaluation of how the myriad of methods developed actually apply, and get a better understanding of what parts of the problem are still unsolved.

Another line of thought is using structure from motion as a part of a bigger whole, e.g. using the 3D models estimated via structure from motion to do higher level inference. Here I think object recognition from image streams could become very fruitful, i.e. form an image sequence, reconstruct the surface of a given object, and try to recognize it based on this data. In this regard, there has been some work on object recognition from 3D models generated from laser scanners, e.g. [138, 192]. There has also been some very interesting developments within object recognition which it could be very interesting to extend to a structure from motion setting, e.g. [115, 152, 180, 179].

Part II

Contributions

CHAPTER 8

Robust Factorization

by: Henrik Aanæs, Rune Fisker, Kalle Åström and Jens Michael Carstensen

Abstract

Factorization algorithms for recovering structure and motion from an image stream have many advantages, but they usually require a set of well tracked features. Such a set is in general not available in practical applications. There is thus a need for making factorization algorithms deal effectively with errors in the tracked features.

We propose a new and computationally efficient algorithm for applying an arbitrary error function in the factorization scheme. This algorithm enables the use of robust statistical techniques and arbitrary noise models for the individual features. These techniques and models enable the factorization scheme to deal effectively with mismatched features, missing features and noise on the individual features. The proposed approach further includes a new method for Euclidean reconstruction that significantly improves convergence of the factorization algorithms.

The proposed algorithm has been implemented as a modification of the Christy–Horaud factorization scheme, which yields a perspective reconstruction. Based on this implementation a considerable increase in error tolerance is demonstrated on real and synthetic data. The proposed scheme can however be applied to most other factorization algorithms.

keywords: Robust statistics, feature tracking, perspective reconstruction, Euclidean reconstruction, structure from motion.

8.1 Introduction

The reconstruction of structure and motion of a rigid object from an image stream is one of the most studied problems within computer vision. A popular way of addressing this problem is to extract and track features through the image sequence and then limit the problem to estimating the structure and motion of these tracked features. A family of effective and popular algorithms for solving this estimation problem are the so called factorization algorithms, see e.g. [39, 46, 102, 104, 111, 136, 156, 204].

These factorization algorithms work by linearizing the camera observation model and give good results rapidly and without an initial guess for the solution. Hence the factorization algorithms are good candidates for solving the structure and motion problem, either as a full solution or as initialization to other algorithms such as bundle adjustment, see e.g. [185, 211].

The factorization algorithms assume that the correspondence or feature tracking problem has been solved. The correspondence problem is, however, one of the most difficult fundamental problems within computer vision. No perfect and truly general solution has yet been presented. For most practical purposes one must deal with erroneously tracked features as input to the factorization algorithm. This fact poses a considerable challenge to factorization algorithms, since they implicitly assume independent identical distributed Gaussian noise on the 2D features (the 2–norm is used as error function on the 2D features). This noise assumption based on the 2–norm is known to perform rather poorly in the presence of outliers induced by such erroneous data. These errors typically arise from mismatching 2D features or from a 2D feature being absent due to occlusion. It is common for badly tracked features to disturb the estimation of structure and motion considerably.

Previous attempts have been made at addressing this problem. Irani and Anandan [102] assumes that the noise is separable in a 3D feature point contribution and a frame contribution. In other words if a 3D feature point has a relatively high uncertainty in one frame it is assumed that it has a similar high uncertainty in all other frames. However, large differences in the variance of the individual 2D feature points is critical to the implementation of robust statistical techniques that can deal with feature point noise, missing features, and feature mismatch in single frames. Morris and Kanade [136] propose a bilinear minimization method as an improvement on top of a standard factorization. The bilinear minimization incorporates directional uncertainty models in the solution. However, the method does not implement robust statistical techniques. Tomasi and Kanade [204] and Jacobs [104] address the problem of missing data points by the use of heuristics. Attempts at solving similar linear problems, in the presence of missing and erroneous data, has also been made, e.g. [181].

Here we propose a combined approach that deals effectively with missing features and is robust towards errors in the matching of the 2D features in a factorization framework. This is achieved by allowing for an arbitrary noise model on the 2D features – i.e. we are not restricted to a Gaussian model. Hereby the proposed approach is capable of dealing effectively with mismatched features or outliers by the use of robust statistics. Arbitrary noise models also deal with missing 2D features by emulating them as being located arbitrarily in the image with very high noise variance.

The proposed approach is implemented as an improvement to the factorization algorithm of Christy and Horaud [39]. The Christy–Horaud algorithm has the advantage, that it assumes a perspective camera model as opposed to a linearized version, e.g. [204]. The Christy–Horaud algorithm with the proposed approach incorporated deals efficiently with real and simulated data containing large feature errors.

The presentation is organized by giving an overview of the factorization algorithms in Section 8.2 followed by a discussion of how to deal with erroneous data in Section 8.3. In Section 8.4 a new numerical algorithm for estimating the optimal subspace of a matrix with weighted entries is proposed. In Sections 8.5 and 8.6 schemes improving the robustness of the factorization approach are presented, followed by experimental results in Section 8.7.

8.2 Factorization Overview

As a courtesy to the reader and to introduce notation a short overview of the factorization algorithm is presented. For a more thorough introduction the reader is referred to [38]. All the factorization methods cited utilize some linearization of the pinhole camera with known intrinsic parameters:

$$s_{ij}\begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_i^T & t_i^x \\ \mathbf{b}_i^T & t_i^y \\ \mathbf{c}_i^T & t_i^z \end{bmatrix} \begin{bmatrix} \mathbf{P}_j \\ 1 \end{bmatrix} \quad , \tag{8.1}$$

where the 3D feature, $\mathbf{P}_j$, is projected in frame i as (x_{ij}, y_{ij}), $\mathbf{t}_i = [t_i^x, t_i^y, t_i^z]^T$ is the appropriate translation vector and $\mathbf{a}_i^T$,$\mathbf{b}_i^T$ and $\mathbf{c}_i^T$ are the three rows vectors of the rotation matrix. The used/approximated observation model can thus be written as:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \mathbf{M}_i \mathbf{P}_j \quad , \tag{8.2}$$

where $\mathbf{M}_i$ is the 2×3 'linearized motion' matrix associated with frame i.

When n features have been tracked in k frames, $i \in [1 \ldots k]$ and $j \in [1 \ldots n]$, the observations from (8.2) can be combined to:

$$\mathbf{S} = \mathbf{M}\mathbf{P} \quad , \tag{8.3}$$

where $\mathbf{M}$ is a $2k \times 3$ matrix composed of the $\mathbf{M}_i$ and $\mathbf{P}$ is a $3 \times n$ matrix composed of the $\mathbf{P}_j$. Thus the elements of $\mathbf{S}$ are given by:

$$\mathbf{S} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{k1} & \cdots & x_{kn} \\ y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{k1} & \cdots & y_{kn} \end{bmatrix} \quad .$$

The solution to this linearized problem is then found as the $\mathbf{M}$ and $\mathbf{P}$ that minimize:

$$\mathbf{N} = \mathbf{S} - \mathbf{M}\mathbf{P} \quad , \tag{8.4}$$

where $\mathbf{N}$ is the residual between model, $\mathbf{MP}$, and the data, $\mathbf{S}$. The residuals, $\mathbf{N}$, are usually minimized using the Frobenius norm. This is equivalent to minimizing the squared Euclidean norm of the reprojection error, i.e. the error between the measured 2D features and the corresponding reprojected 3D feature. Thus the objective function is:

$$\min_{\mathbf{M},\mathbf{P}} ||\mathbf{S}-\mathbf{MP}||_F^2 = \min_{\mathbf{M},\mathbf{P}} \sum_{j=1}^{n} ||\mathbf{S}_j - \mathbf{MP}_j||_2^2 \ , \tag{8.5}$$

where $\mathbf{S}_j$ and $\mathbf{P}_j$ denote the j^{th} column of $\mathbf{S}$ and $\mathbf{P}$, respectively. In this case the solution to $\mathbf{M}$ and $\mathbf{P}$ can be found through the singular value decomposition, SVD, of $\mathbf{S}$.

It is noted, that for any invertible 3×3 matrix, $\mathbf{A}$:

$$\mathbf{MP} = \mathbf{MAA}^{-1}\mathbf{P} = \tilde{\mathbf{M}}\tilde{\mathbf{P}} \ . \tag{8.6}$$

Hence the solution is only defined up to an affine transformation. In [39] an Euclidean reconstruction is achieved by estimation of an $\mathbf{A}$, such that the rotation matrices, $[\mathbf{a}_i \ \mathbf{b}_i \ \mathbf{c}_i]^T$, are as orthonormal as possible. Further details and an improved approach is presented in Section 8.6.

8.2.1 Christy–Horaud Factorization

The factorization algorithm that the proposed method extends on is that of Christy and Horaud [39], see Figure 8.1. The Christy–Horaud algorithm is an extension to the work in [156, 204] partly by incorporating the work of [55]. The algorithm has the advantage of iteratively achieving a solution to the original non–linearized problem, thus achieving perspective reconstruction. This is achieved by iteratively solving a linearized version of the problem, and then modifying the data to approach the perspective camera. The update formula for the data, i.e. $\mathbf{S}$ or x_{ij}, y_{ij} is [39]:

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \left(\begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} - \begin{bmatrix} x_{o_i} \\ y_{o_i} \end{bmatrix} \right) (1 - \epsilon_{ij}) \ , \tag{8.7}$$

where $(\tilde{x}_{ij}, \tilde{y}_{ij})$ is the updated data, (x_{o_i}, y_{o_i}) is the object origin projected onto frame i and ϵ_{ij} is the scaled depth defined as:

$$\epsilon_{ij} = \frac{\mathbf{c}_i^T \cdot P_j}{t_i^z} \ . \tag{8.8}$$

For further details the reader is referred to [38, 39].

8.3 Dealing with Imperfect Data

8.3.1 Types of Errors

A framework for dealing with imperfect data should be geared towards the types of errors expected. Three types of errors have been identified, the first two originating from [223].

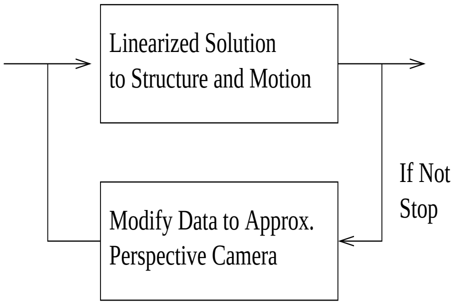


Figure 8.1: Overview of the Christy–Horaud factorization algorithm. A linearized – e.g. paraperspective – solution is iterated into a perspective.

- **Bad Feature Locations.** The locations of a 2D feature is distorted. If this distortion is anisotropic and/or the covariance structure varies the Frobenius norm is sub–optimal. This is the problem addressed in [102, 136].
- **False Matches.** There has been a mismatch of 2D features, i.e. two 2D feature originating from *different* 3D features have been matched as originating from the *same* 3D feature.
- **Missing Feature.** The projection of a 3D feature can not be found or matched, e.g. due to occlusion or simply by 'breakdown' of the feature matching algorithm. This is addressed in [104, 204].

Our proposed method deals with all these types of errors by attaching an uncertainty to the individual 2D features. It is implemented by associating a weighting structure to $\mathbf{S}$. This mainly effects the solution of (8.5), but in the case of Christy–Horaud it also effects the object frame origin (see Section 8.5). When weights are introduced (8.5) becomes:

$$\min_{\mathbf{M},\mathbf{P}} \sum_{j=1}^{n} ||\mathbf{V}_j(\mathbf{S}_j - \mathbf{M}\mathbf{P}_j)||_2^2 \ , \tag{8.9}$$

where $\mathbf{V}_j$ is an $2k \times 2k$ weighting matrix representing the weights of the j^{th} column of $\mathbf{S}$. The minimization of (8.9) is the subject of Section 8.4. The $\mathbf{V}_j^T\mathbf{V}_j$ is seen to be the inverse covariance structure of $\mathbf{S}_j$ and (8.9) is equivalent to minimizing the Mahalanobis distance. The variance, $\mathbf{\Sigma}_j$, of the noise on S_j is incorporated by:

$$\mathbf{\Sigma}_j^{-1} = \mathbf{V}_j^T\mathbf{V}_j \ . \tag{8.10}$$

An approach with this uncertainty formulation is seen to deal with the three types of identified errors:

- **Bad Feature Locations.** Assuming Gaussian noise, the weights can be constructed to incorporate the uncertainty structure of the 2D features as shown above. This approach can even deal with arbitrary Gaussian noise. With the presented formulation of the weights - one V_j matrix per column - covariance between the x and y coordinates can be expressed. An extension to general covariance between all features would require a 3D tensor formulation of the weights.

- **False Matches.** A mismatched 2D feature can be down weighted, even approaching zero, leaving it out of the optimization to any desired extent.
- **Missing Features.** Is equivalent to predicting that the missing 2D feature is located some where in the image, but that the uncertainty of the prediction is very high.

The information of missing features should be apparent from the feature matching algorithm and should be directly expressed in $\mathbf{\Sigma}_j$. If prior knowledge about the distribution of bad feature location is present, this can also be expressed in $\mathbf{\Sigma}_j$. However this not a requirement. *False Matches* are almost by nature unknown a priori so robust statistical techniques are employed to deal with these errors. Note that in the absence of prior knowledge, the $\mathbf{\Sigma}_j$ is initialized as the identity matrix.

8.3.2 Adaptive Weighting

As described in [89, 208, 223] among others, false matches give rise to outliers in the data. Here outliers are understood as 2D features where the residual, i.e. the distance between the original 2D feature and it's reprojected counterpart is large. If the percentage of false matches is relatively low (considerably lower than 50%) false matches can be dealt with effectively by diminishing the effect of outliers. In practice, this is done via a robust error function. Popular robust functions are the truncated quadratic and Huber's M–estimator (sometimes called Huber Norm), see Figure 8.2. For a detailed discussion the reader is referred to [18]. The use of robust error functions efficiently deals with the problem of false matches.

These error functions are implemented via Iteratively Reweighted Least Squares (IRLS). IRLS works by iteratively fitting the model to the data by minimizing the weighted least squares of the residuals. These weights are then altered such that the residuals are reweighted according to the desired error function and not the 2-norm. In this approach, the weights, w_{ij}, from the IRLS are collected in:

$$\mathbf{W}_j = \begin{bmatrix} w_{1j} & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & w_{nk} & & & \\ & & & w_{1j} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & w_{nk} \end{bmatrix} \tag{8.11}$$

and incorporated by letting:

$$\mathbf{V}_j^T\mathbf{V}_j = \mathbf{W}_j^T\mathbf{W}_j \quad .$$

In case of a Gaussian prior on the 2D features with covariance $\mathbf{\Sigma}_j$, the V_j are given by:

$$\mathbf{V}_j^T\mathbf{V}_j = \mathbf{W}_j^T\mathbf{\Sigma}_j^{-1}\mathbf{W}_j \quad .$$

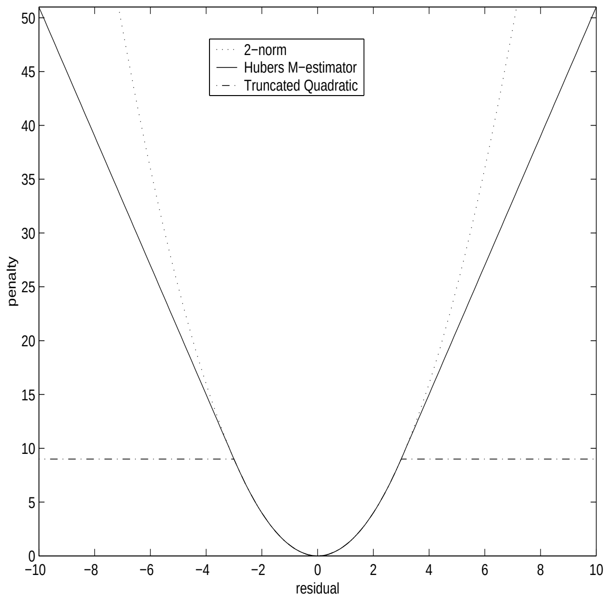


Figure 8.2: Three popular error functions with $k = 3$ for the truncated quadratic and the Huber's M–estimator.

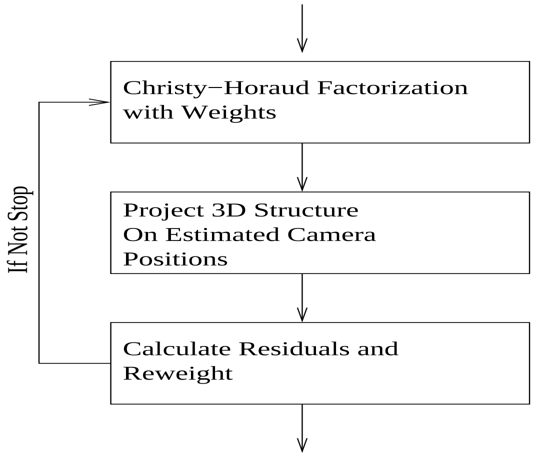

Christy−Horaud Factorization with Weights
Project 3D Structure On Estimated Camera Positions
Calculate Residuals and Reweight
If Not Stop

Figure 8.3: Overview of the proposed reweighting scheme. This is employed to deal effectively with false matches

As an example, the reweighting formula for the truncated quadratic is:

$$w_{ij} = \begin{cases} 1 & ||N_{ij}||_{\Sigma_{ij}} < k \\ \sqrt{\frac{k^2}{N_{ij}^2}} & ||N_{ij}||_{\Sigma_{ij}} > k \end{cases} , \tag{8.12}$$

where N_{ij} is the residual on datum ij, w_{ij} is the corresponding weight and k is a user defined constant relating to the image noise. Here $||\cdot||_{\Sigma_{ij}}$ denotes the Mahalanobis distance induced by $\mathbf{\Sigma}_j$. If no prior is available the 2-norm is used. The parameter k is an indication of the general image noise in the image. For a detailed discussion of how to choose k refer to [85]. It is noted, that the experimental results (see Figure 8.20) show that the proposed method is rather robust towards the choice of k.

It is also noted, that almost arbitrary error functions can be implemented via the IRLS approach. *This allows an arbitrary noise model on the bad feature locations* by enabling the implementation of the induced error function. The scheme for combining IRLS with the weighted factorization is illustrated in Figure 8.3.

8.4 Separation with Weights

The main computational problem in the proposed approach is to determine a solution to the weighted least squares problem (8.9). The solution to (8.9) is $\mathbf{M}$ and $\mathbf{P}$ given $\mathbf{S}$ and $\mathbf{V}_j$. Note that a SVD can not be applied as for (8.5). To solve (8.9), a method similar to the idea in the Christy-Horaud factorization algorithm [39] is proposed. This method is generally known as surrogate modeling, see e.g. [20]. Surrogate modeling works by applying a computationally 'simpler' model to iteratively approximate the original 'hard' problem.

The best known example of surrogate modeling is probably the Newton optimization method. Here a 2^{nd} order polynomial is approximated to the objective function in each iteration and a temporary optimum is achieved. This temporary optimum is then used to make a new 2^{nd} order approximation, and thus a new temporary optimum. This is continued until convergence is achieved.

Here (8.5) is used to iteratively approximate (8.9) getting a temporary optimum, which in turn can be used to make a new approximation. The approximation is performed by modifying the original data, $\mathbf{S}$, such that the solution to (8.5) with the modified data, $\tilde{\mathbf{S}}$, is the same as (8.9) with the original data. By letting ˜ denoting modified data, the goal is to obtain:

$$\begin{aligned} \min_{M,P} \sum_{j=1}^{n} ||\mathbf{V}_j(\mathbf{S}_j - \mathbf{M}\mathbf{P}_j)||_2^2 &= \\ \min_{M,P} \sum_{j=1}^{n} \mathbf{N}_j^T \mathbf{V}_j^T \mathbf{V}_j \mathbf{N}_j &\overset{def}{=} \\ \min_{M,P} \sum_{j=1}^{n} \tilde{\mathbf{N}}_j^T \tilde{\mathbf{N}}_j &= \\ \min_{M,P} \sum_{j=1}^{n} ||\tilde{\mathbf{S}}_j - \mathbf{M}\mathbf{P}_j||_2^2 &\ , \end{aligned} \tag{8.13}$$

where $\mathbf{N} = [\mathbf{N}_1 \ldots \mathbf{N}_n]$ denotes the residuals:

$$\mathbf{N}_j = \mathbf{S}_j - \mathbf{M}\mathbf{P}_j \ .$$

Hereby the subspace, $\mathbf{M}$, can be found via SVD and $\mathbf{P}$ via the normal equations once $\mathbf{M}$ is known. Let q denote the iteration number, then the algorithm goes as follows:

1. **Initialize.** $\tilde{\mathbf{S}}^0 = \mathbf{S}$, $q = 1$.

2. **Estimate Model.** Get $\mathbf{M}^q$ by the singular vectors corresponding to the three largest singular values of $\tilde{\mathbf{S}}^{q-1}$, via SVD. Get $\mathbf{P}^q$ from

$$\forall j: \quad \mathbf{P}_j^q = \left[\mathbf{M}^{qT}\mathbf{V}_j^T\mathbf{V}_j\mathbf{M}^q\right]^{-1} \mathbf{M}^{qT}\mathbf{V}_j^T\mathbf{V}_j \cdot \mathbf{S}_j \ .$$

3. **Calculate Residuals.** $\mathbf{N}^q = \mathbf{S} - \mathbf{M}^q \cdot \mathbf{P}^q$.

4. **Modify Data.**

$$\begin{aligned} \forall j: \ \tilde{\mathbf{N}}_j^q &= \mathbf{V}_j\mathbf{N}_j^q \\ \tilde{\mathbf{S}}^q &= \mathbf{M}^q\mathbf{P}^q + \tilde{\mathbf{N}}^q \ . \end{aligned}$$

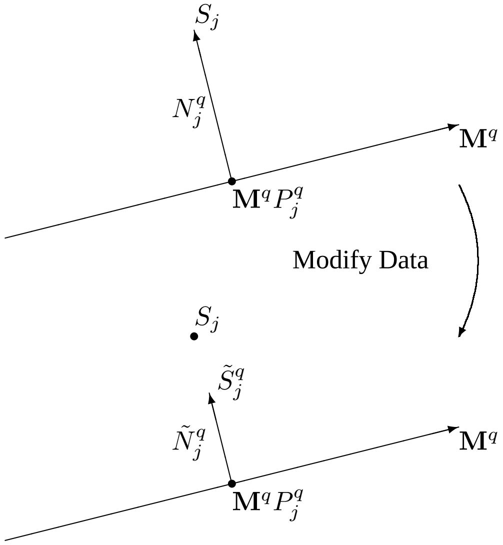


Figure 8.4: A geometric illustration of how the data is modified in steps 3. and 4. of the proposed algorithm for separation with weights.

5. **If Not Stop.** $q = q + 1$, goto 2. The stop criteria is

$$||\mathbf{N}^q - \mathbf{N}^{q-1}||_\infty < tolerance \ \ .$$

As illustrated in Figure 8.4 the data, $\mathbf{S}$, is modified such that the Frobenius norm of the modified residuals, $\tilde{\mathbf{N}}_j^q$, are equal to norm of the original residuals, $\mathbf{N}_j^q$, in the norm induced by the weights, $\mathbf{V}_j$. The last part of step 2. ensures that the residual, $\mathbf{N}_j$, is orthogonal to $\mathbf{M}$ in the induced norm, since $\mathbf{M}^q\mathbf{P}_j^q$ is the projection of $\mathbf{S}_j$ onto $\mathbf{M}^q$ in the induced norm.

8.4.1 Separation Performance with Given Weights

A quasi–Newton method could also be used to solve (8.9). One of the most effective quasi–Newton methods is the Broyden,Fletcher,Goldfarb & Shanno method (BFGS) [30, 73, 78, 175]. However this is not recommended, since the proposed method is faster and more reliable. Partly because with a ’standard’ optimization method the problem is very likely to become ill-conditioned due to the potentially large differences in weights.

To illustrate this, the proposed method and the BFGS were tested against each other, see Table 8.1. The $\mathbf{S}$ matrix was formed by (8.3) with noise added from a compound Gaussian distribution. The compound distribution consisted of two Gaussian distributions, one with a standard deviation 10 times larger than the other. The fraction of the larger varying Gaussian is the *Noise Level*. It is seen, that the proposed method performs better than BFGS, and that the BFGS approach did not converge for $\mathbf{S} = 40 \times 40$ and *Noise Level*=0.5, due to ill-conditioning.

S	Noise	Proposed	BFGS	Flop
$k \times n$	Level	Method		Ratio
20x40	0.02	1.20e+07	2.32e+08	19.33
20x40	0.10	1.58e+07	5.81e+08	36.73
20x40	0.50	5.50e+07	4.22e+08	7.67
40x40	0.02	7.20e+07	1.99e+09	27.58
40x40	0.10	1.15e+08	3.64e+09	31.73
40x40	0.50	3.59e+08	–	–
80x40	0.02	5.17e+08	1.78e+10	34.41
80x40	0.10	8.00e+08	7.08e+10	88.52
80x40	0.50	2.30e+09	8.74e+10	37.93

Table 8.1: Computational time comparison of the proposed algorithm with MatLab's BFGS (fminu()), – denotes that the optimization did not converge due to ill-conditioning. The computational time is measured in Flops.

A formal proof of convergence seems to be infeasible. However this is a common problem for most numerical optimization schemes. The convergence has been followed closely during the tests of the algorithm. These empirical results show that the convergence has been very good.

A thorough investigation of the numerical properties of the algorithm is presented in [143]. Here the proposed method for separation is also compared to other alternatives besides BFGS with favorable results. Among these are the types of algorithms where $\mathbf{M}$ is estimated given $\mathbf{P}$ and vice versa in an iterative manner, like the one presented in [181].

8.5 The Object Frame Origin

In order to achieve robust factorization, the calculation scheme of the object frame origin, (x_{o_i}, y_{o_i}) applied in the Christy–Horaud algorithm needed to be improved. The object frame origin is the point around which the camera model is linearized, and errors here will propagate to the rest of the estimation. Making the estimation origin robust is essential to deal efficiently with erroneous data. In the Christy–Horaud algorithm, the 2D features corresponding to an arbitrary 3D feature is chosen. This scheme is error prone with erroneously tracked features, e.g. if the chosen feature corresponds to a false feature match.

A natural alternative, would be to iteratively calculate the center of mass from the estimated structure and reproject it onto the estimated camera positions. This however turns out to be *unstable*.

We still choose to use the center of mass, but calculated from the 2D features and the reprojected 3D features in a similar manner as (8.7). Combining 2D features and the reprojected 3D features allows us to diminish the influence of the 2D features with high uncertainty, while stabilizing the estimation with the reliable 2D features. The formula for the object frame origin of frame i is given by:

$$\begin{bmatrix} x_{o_i} \\ y_{o_i} \end{bmatrix} = \frac{\sum_{j=1}^{n} \begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} (1+\epsilon_{ij})}{n + \sum_{j=1}^{n} \epsilon_{ij}} \quad , \tag{8.14}$$

where ϵ_{ij} is defined in (8.8) and $(\tilde{x}_{ij}, \tilde{y}_{ij})$ is a weighted mean between the 2D feature and the reprojected 3D feature defined by.

$$\begin{bmatrix} \tilde{x}_{ij} \\ \tilde{y}_{ij} \end{bmatrix} = \gamma_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} + (1 - \gamma_{ij}) \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \end{bmatrix} \quad . \tag{8.15}$$

Here γ_{ij} denotes the weight. This weighted mean ensures robustness towards outliers, since outliers are partly replaced by the estimate of the model. If $\mathbf{V}_j^T \mathbf{V}_j$ is *diagonal* then the weight γ_{ij} is given by:

$$\gamma_{ij} = \frac{1}{\pi} \arctan(\alpha(v_{ij} - \frac{1}{k'})) + \frac{1}{2} \quad , \tag{8.16}$$

where v_{ij} is the i^{th} diagonal element of $\mathbf{V}_j$. The 'cutoff' value k' serves the same function as k in the truncated quadratic and the Huber's M–estimator. It is seen that (8.16) is a smooth version of the step function around $\frac{1}{k'}$ where α controls how smooth this approximation is.

If $\mathbf{V}_j^T \mathbf{V}_j$ is *not diagonal* then it's principal components are used. So instead of v_{ij} the eigenvalues of $\mathbf{V}_j^T \mathbf{V}_j$ and a linear combination of (x_{ij}, y_{ij}), are used in (8.14) and (8.15). This linear combination corresponds to the eigenvectors of $\mathbf{V}_j^T \mathbf{V}_j$. With this modification to the approach, the final flow chart is see in Figure 8.5.

8.6 Euclidean Reconstruction

The objective of Euclidean reconstruction is to estimate the $\mathbf{A}$ in (8.6) such that the estimated $\mathbf{a}_i$,$\mathbf{b}_i$ and $\mathbf{c}_i$ of (8.1) are as orthonormal as possible. In the paraperspective case [156], which is the linearization used by Christy and Horaud [39], the M_i's composing $\mathbf{M}$ are given by:

$$\mathbf{M}_i = \frac{1}{t_i^z} \left[\begin{array}{c} \mathbf{a}_i^T - x_{oi}\mathbf{c}_i \\ \mathbf{b}_i^T - y_{oi}\mathbf{c}_i \end{array} \right] = \left[\begin{array}{c} \mathbf{I}_i^T \\ \mathbf{J}_i^T \end{array} \right] \quad ,$$

where (x_{oi}, y_{oi}) is the object origin projected in frame i.

Since the paraperspective approximation is obtained by linearizing $\frac{1}{t_i^z}\mathbf{c}_i^T \cdot \mathbf{P}_j$ the orthonormal constraints are restricted to $\mathbf{a}_i$ and $\mathbf{b}_i$.With $\mathbf{Q} = \mathbf{A}\mathbf{A}^T$ these constraints can be formulated as [39, 156]:

$$\begin{array}{ll}
\forall i & \mathbf{a}_i^T \mathbf{Q} \mathbf{a}_i = \mathbf{b}_i^T \mathbf{Q} \mathbf{b}_i \Rightarrow \\
\forall i & \dfrac{\mathbf{I}_i^T \mathbf{Q} \mathbf{I}_i}{1 + x_{oi}^2} - \dfrac{\mathbf{J}_i^T \mathbf{Q} \mathbf{J}_i}{1 + y_{oi}^2} = 0 \\
\forall i & \mathbf{a}_i^T \mathbf{Q} \mathbf{b}_i = 0 \Rightarrow \\
\forall i & \mathbf{I}_i^T \mathbf{Q} \mathbf{J}_i - \dfrac{x_{oi} y_{oi} (\mathbf{I}_i^T \mathbf{Q} \mathbf{I}_i)}{2(1 + x_{oi}^2)} - \dfrac{x_{oi} y_{oi} (\mathbf{J}_i^T \mathbf{Q} \mathbf{J}_i)}{2(1 + y_{oi}^2)} = 0
\end{array}$$

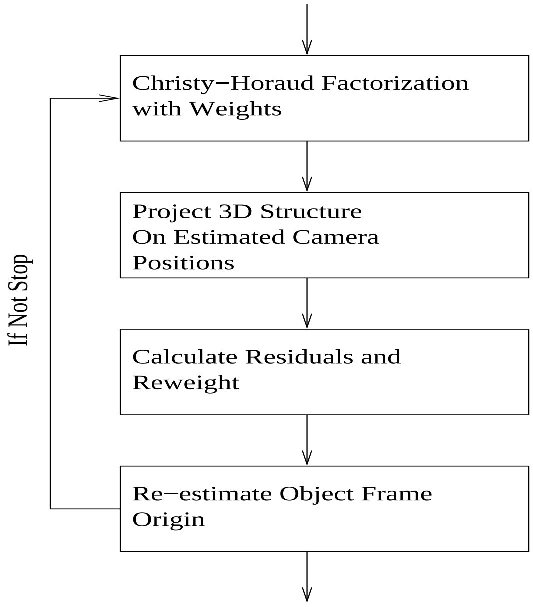

Christy−Horaud Factorization with Weights
Project 3D Structure On Estimated Camera Positions
Calculate Residuals and Reweight
Re−estimate Object Frame Origin
If Not Stop

Figure 8.5: Overview of the proposed approach. This approach also deals with estimating the object frame origin robustly.

With noise, this cannot be achieved for all i and a least squares solution is sought. In order to avoid the trivial null-solution the constraint $\mathbf{a}_1^T\mathbf{Q}\mathbf{a}_1 = \mathbf{b}_1^T\mathbf{Q}\mathbf{b}_1 = 1$ is added [39, 156] and the problem is linear in the elements of $\mathbf{Q}$.

Unfortunately it is impossible to reconstruct $\mathbf{A}$ if $\mathbf{Q}$ has negative eigenvalues. This problem indicates that an unmodeled distortion has overwhelmed the third singular value of $\mathbf{S}$ [156]. This is a fundamental problem when the factorization method is used on erroneous data.

To solve this problem we propose to parameterize, $\mathbf{Q}$ as:

$$\mathbf{Q}(\mathbf{e},\lambda) = \mathbf{R}(\mathbf{e}) \begin{bmatrix} \lambda_1^2 & 0 & 0 \\ 0 & \lambda_2^2 & 0 \\ 0 & 0 & \lambda_3^2 \end{bmatrix} \mathbf{R}(\mathbf{e})^T \quad , \tag{8.17}$$

where $\mathbf{R}(\mathbf{e})$ is a rotation matrix with the three Euler angles denoted by $\mathbf{e}$. The term $\mathbf{a}_1^T\mathbf{Q}\mathbf{a}_1 = \mathbf{b}_1^T\mathbf{Q}\mathbf{b}_1 = 1$ is replaced by $det(\mathbf{A}) = 1$, such that the overall scale of $\mathbf{A}$ is much more robust and less sensitive to the noise in a particular frame.

Hence the estimation of $\mathbf{Q}$ is a nonlinear optimization problem in six variables, with a guaranteed symmetric positive definite $\mathbf{Q}$. Our experience shows that this approach to the problem is well behaved with a quasi–Newton optimization method.

8.7 Experimental Results

In order to illustrate the proposed algorithms ability to deal with the errors identified, it has been applied to a set of real images. To provide a more systematic test, it was then applied to a set of simulated data, where arbitrary errors could be induced.

8.7.1 Real Data

The proposed approach was run on three sequences, demonstrating different properties. The first sequence was the hotel sequence [135] with accompanying features, see Figure 8.6. Some of these features were mismatched. To illustrate the handling of mismatched features, notice the different position of feature 64 in Figures 8.7 and 8.8. The lines in the images denotes the residual between the tracked 2D features and the back projected 3D estimates. The correct position of a feature is determined by the location with the greatest support.

The proposed approach with the truncated quadratic error function was applied to the hotel sequence ($k = 3$, $k' = 2$, $\alpha = 20$). The result is shown in Figures 8.7 and 8.8. Two things should be noted here. First, that the mismatched feature 64 does not effect the estimate of the other features. Second, that the back projections of the 3D estimate of feature 64 are located correctly in all images (including 8.7 where the 2D feature itself is mismatched).

Comparing with the Christy–Horaud approach it is seen that the mismatch error in Figure 8.7 effects the overall estimation of feature 64's 3D position. To give a more quantitative evaluation, the residuals of the non erroneous features were summed up – it is assumed that there were no more than 5% errors. It is seen from Table 8.2 that due to the capability to

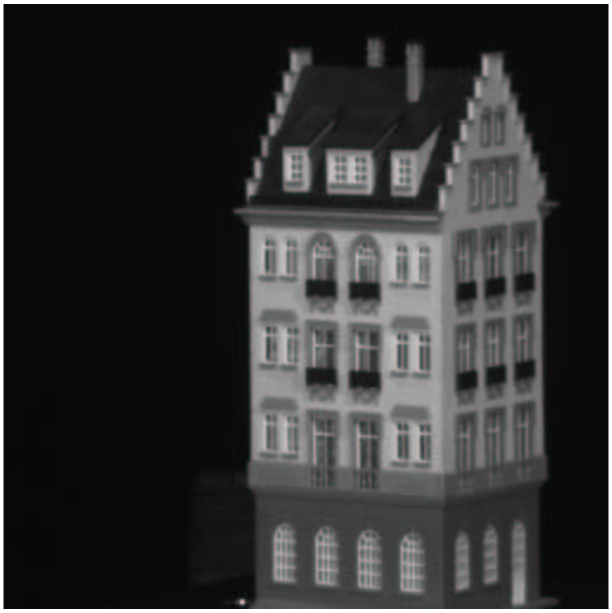


Figure 8.6: A sample frame from the Hotel sequence with 197 tracked features, some of these are mismatched.

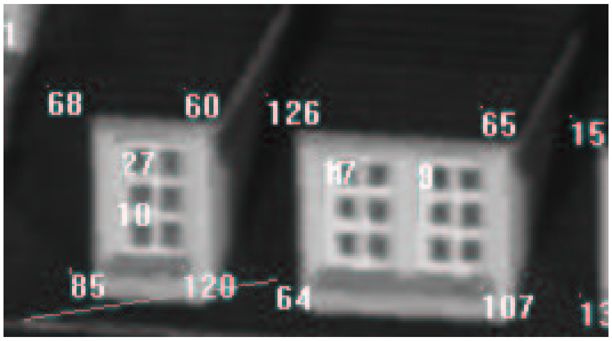


Figure 8.7: A section of the Hotel sequence illustrating where feature 64 is mismatched. The correct position of feature 64 is at the end of the residual vector (bottom left of the image).

Error Function:	$\frac{1}{n}\sum_i \|Res\|_i$
Truncated Qudratic (robust)	1.94 $pixels^2$
2–Norm (non–robust)	3.40 $pixels^2$

Table 8.2: Comparison of the 95% smallest residuals. It is noted, that without sub–pixel feature location this number is highly unlikely to be lower than 1.

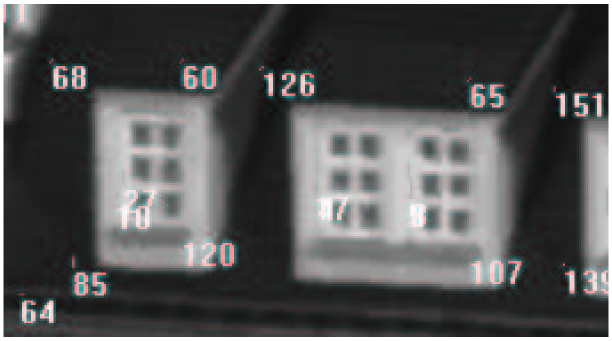


Figure 8.8: A section of the Hotel sequence where feature 64 is not erroneous. Notice how little effect the false match error has, with the proposed method.

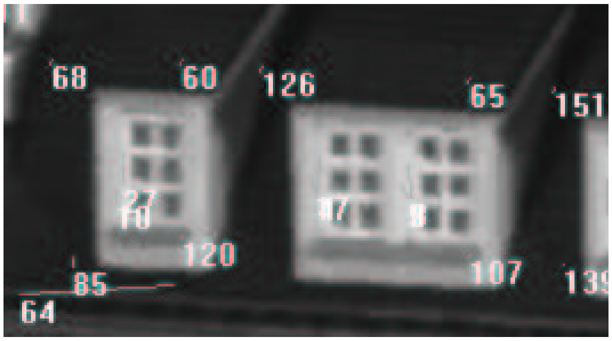


Figure 8.9: Same section and frame as Figure 8.8, but with the Christy–Horaud method. Notice the increased effect of the errors.

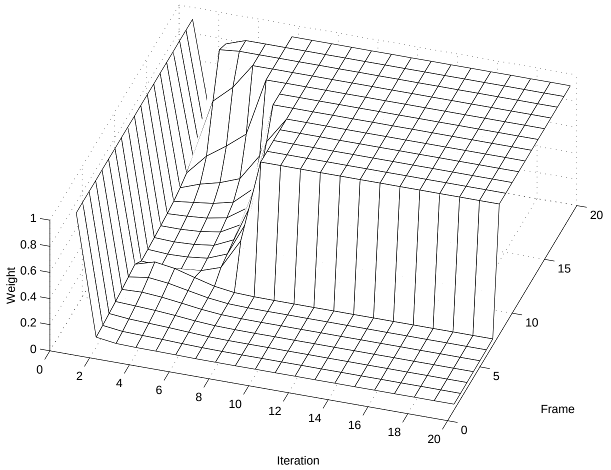


Figure 8.10: Weight evolution for feature 64 plotted as a function of frame and iteration.

implement a robust error function, the fit to the non–erroneous data is improved significantly.

To illustrate the reweighting process, the evolution of the weights off the mismatched feature 64 is depicted in Figure 8.10. After the first iteration all the 2D features are down weighted, since only an erroneous 3D estimate exists due to the mismatch features and the uniform weighting. In the following iterations the correctly located features obtains increasing weights whereas the weights of the mismatch feature decrease toward zero. The plot directly shows how the proposed method can be used to detect mismatched features.

To demonstrate the proposed approach's ability to deal with missing features, it was run on the kitchen sequence [135], see Figure 8.11. Here some of the accompanying features were missing, e.g. feature 5 in figure 8.13. Note that the back projection of feature 5 in figure 8.13 is located correctly, indicating that the 3D estimate is correct. From the depicted residuals, which are hardly visible, it is seen that the missing features does not disrupt the structure and motion estimation. Thus giving the desired result in dealing with missing features.

The proposed modification to the Christy–Horaud algorithm should not considerably decrease it's ability to deal with perspective reconstruction. To validate this, both approaches were applied to a sequence with considerable depth, see Figure 8.14. The 2D features were carefully hand–tracked to ensure that the algorithms were directly comparable. The results were evaluated by comparing the two estimated structures via the standard shape distance measure from statistical shape analysis, the Procrustes distance [57, 81]. The Procrustes distance is obtained by normalizing the two structures and the applying the similarity transform [89] such that the mean squared error is minimized. The remaining mean squared error is then the Procrustes distance. The resulting Procrustes distance was 0.03. This implies that the Christy–Horaud method with the proposed enhancement maintains it's ability to deal

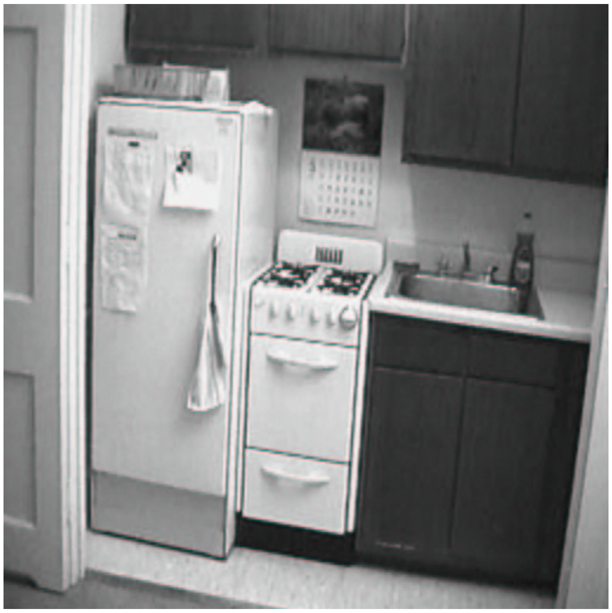


Figure 8.11: A sample image from the kitchen sequence.

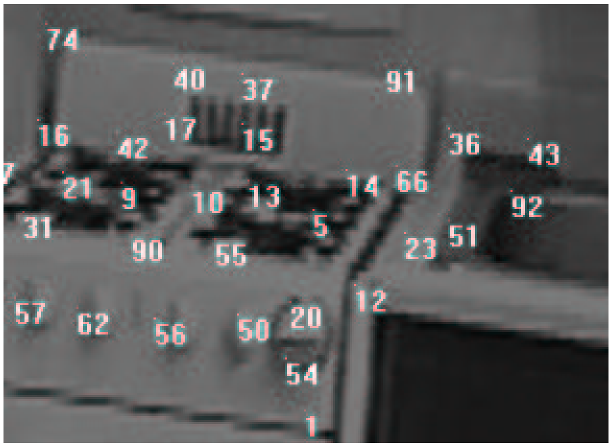


Figure 8.12: Close up on the kitchen sequence. Notice the location of feature 5.

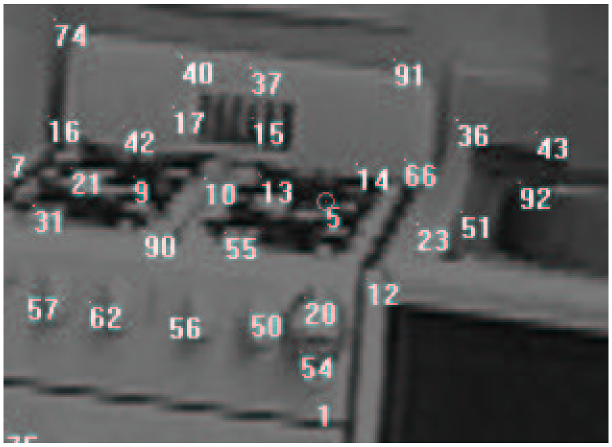


Figure 8.13: Closeup on the kitchen sequence. A circle around a feature denotes it is the estimated location. The reason being that the respective feature is missing in this frame. Not that the estimated position of feature 5 correspond to it's position in figure 8.12.

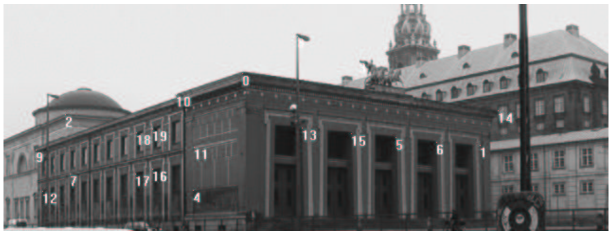


Figure 8.14: A sample frame from an image sequence of Thorvaldsens Museum in Copenhagen with 20 hand tracked features through 8 frames.

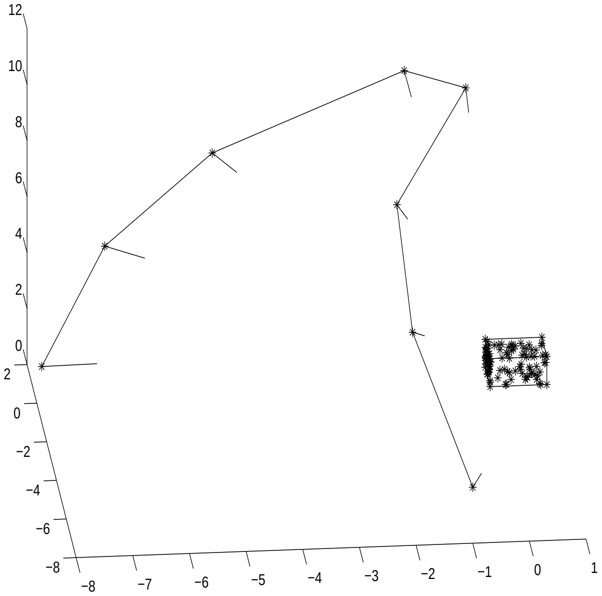


Figure 8.15: The setup for the simulated data. A box with 100 features is 'photographed' from 8 views. These views are marked on the trajectory curve.

with perspective data.

8.7.2 Simulated Data

To perform a more systematic test of the proposed algorithms ability to deal with errors in the tracked data a simulated data set was created, see Figure 8.15. Several kinds of errors were introduced into this data set, hereby testing the approach with respect to the three identified types of errors. The Hubers M–estimator and truncated quadratic error functions were applied with the parameters settings $k = 0.02$ [1] and $k' = 2$ and we set $\alpha = 20$. The results were evaluated by the Procrustes distance [57, 81]. In the extreme cases where an estimation process did not converge, this is denoted by a missing measurement.

The first experiment consists of corrupting the 2D features of the simulated data by a compound distribution of two Gaussians one with a standard deviation of 0.005 and the other 10 times larger. Two different error functions were applied; the Huber M-estimator and the truncated quadratic. For comparison the Christy–Horaud algorithm was also applied. From Figure 8.16 it is seen, that the choice of error functions has a considerable effect on the result, and that the proposed approach is capable of implementing them.

[1] It should be noted, that k depends on the image noise and hence also on the image size. In this simulated data, the image size was unnaturally small (0.25 $\times$ 0.25), and as such $k = 0.02$ should not be seen as a guide line.

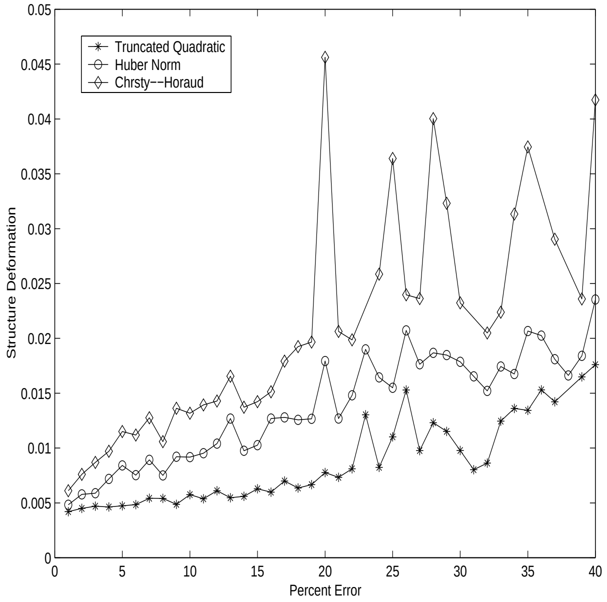


Figure 8.16: Errors for the simulated data corrupted with a compound distribution of two Gaussians. The abscissa denotes the likelihood, $[0 \ldots 1]$, that samples are obtained from the higher varying Gaussian.

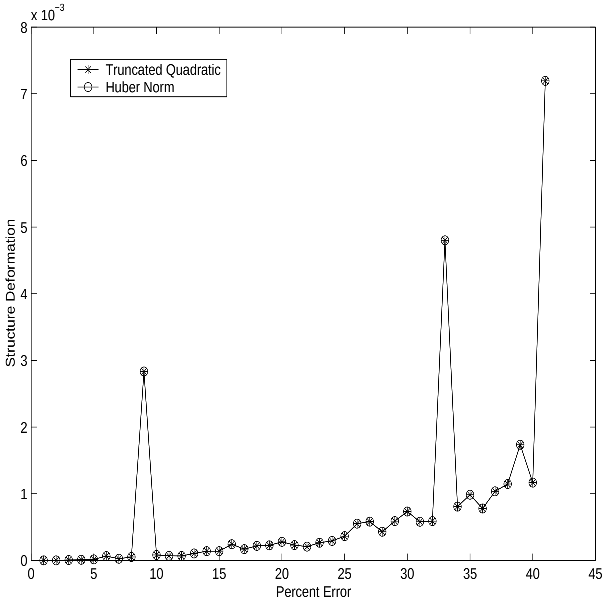


Figure 8.17: Errors for corrupting the simulated data by removing the 2D features at random. The abscissa indicates how many percent of the data has been removed. No difference is seen between the two error functions.

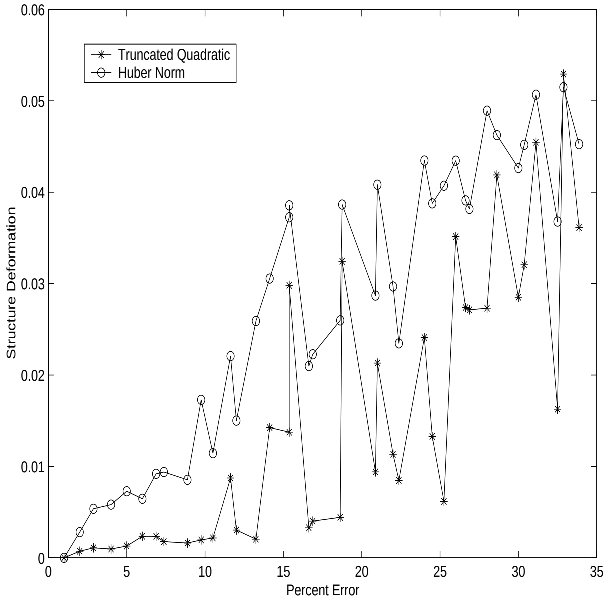


Figure 8.18: Errors for the simulated data with swapped features emulating mismatched features. The data set upon which features were swapped was the original uncorrupted. The abscissa denoted the percentage of altered features.

In the second experiment, an increasing number of the 2D features were removed from the data set. These missing features were emulated as being located in the center of the frame in question with a weight equal to 10^{-6} of the weight of the normal 2D features. From the results, it is seen that the proposed approach is highly robust towards this type of error, since up to 40% missing features corrupts the estimated structure by less than 10^{-2}, see Figure 8.17. The approach of Christy and Horaud could not work on this data and was not included.

To test the tolerance to mismatched features, we emulated these by swapping 2D features of the simulated data. The results in Figure 8.18 illustrate that the proposed approach also deals efficiently with mismatched features and has been shown to be robust toward the identified types of errors.

To challenge the proposed approach all the experiments on the simulated data are performed with up to 35–40% errors. It should be noted, that the algorithm works very well with up to 10–20% errors. This is the amount of errors that the proposed approach is expected to work on.

To evaluate the effect of the proposed approach for estimating the object frame origin noise was added exclusively to the feature that Christy–Horaud uses as object frame origin. The evaluation were performed on the simulated data and the noise was Gaussian with a standard deviation of 0.05. Five experiments were conducted comparing the Christy–Horaud approach to the proposed approach. The results are shown in Table 8.3. It is seen that significant improvement is obtained with respect to error as well as convergence. For validation purposes the same experiment was made, but this time noise was added to a different feature then the one used as object origin by Christy-Horaud. In this case the two approaches gave similar results. Note also, that the original approach for estimation of the object frame origin could be problematic to use with missing features, since the 2D feature chosen as object frame origin is likely to be missing in at least one frame.

Factorization Approach	Percent Convergence	Mean Procrustes Distance
Christy–Horaud	60%	0.0153
Proposed Approach	100%	0.0055

Table 8.3: Results with noise exclusively on the feature Christy–Horaud uses as object frame origin. This amounts to 1% of the data being erroneous.

In order to demonstrate the benefits of the proposed method for Euclidean reconstruction, the experiments with mismatched features was repeated *without* the proposed approach. Instead the original method for Euclidean reconstruction proposed in [39, 156] was applied. The number of runs that did not converge have been summed up with and without the proposed method. The change is dramatic.

To illustrate, that the proposed method is not overly sensitive to the choice of error function and the involved parameters, these were varied. This was done with the truncated quadratic error function in the experiments with mismatched features, see Figure 8.18. The k parameter was altered with $\pm 50\%$. From the results in Figure 8.20 it is seen that the proposed approach is not overly sensitive to the choice of parameters in the error function.

8.8 Conclusion and Discussion

An approach for applying arbitrary error functions in the factorization algorithms for structure and motion has been presented. This ability has been exploited to implement robust statistical techniques, whereby errors in the 2D features have been dealt with effectively. The algorithm has been implemented as an extension to the Christy–Horaud factorization algorithm [39], whereby perspective reconstruction is achieved. The core of the approach is however so general, that it can also be applied to most other factorization schemes like [17, 94, 189, 195, 156, 204]. This approach has been applied to both simulated and real data, where it is demonstrated that the algorithm deals well with all the expected types of errors. Thus making the proposed approach a robust factorization method.

To further investigate the possibilities of the proposed approach, we aim at implementing it with other factorization algorithms, e.g. [94], hence making a robust factorization algorithm with uncalibrated cameras. The proposed method also allows for more elaborate noise structures incorporating covariance within and between images. This covariance structure has been implemented with the algorithm of Section 8.4 and sparsely tested, but should be thoroughly tested on real image data, in order to evaluate it's robustness. This is the goal of ongoing research.

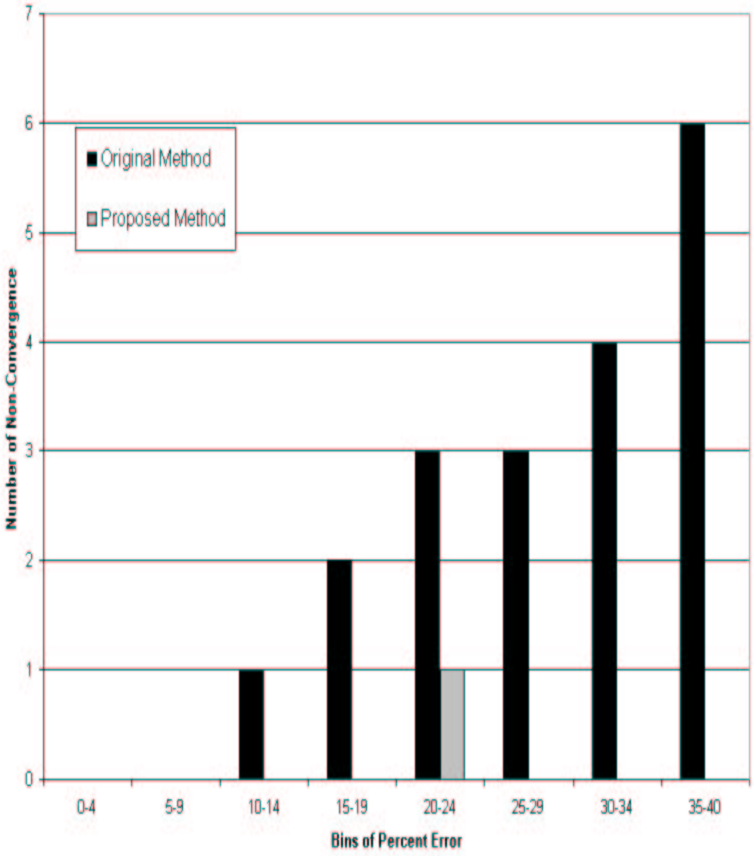


Figure 8.19: The number of non–converging runs on the experiment of Figure 8.18, using the Huber M–estimator as error function. The experiment was made with and without the proposed method for Euclidean reconstruction. The runs are pooled in bins of 5 to give a better overview.

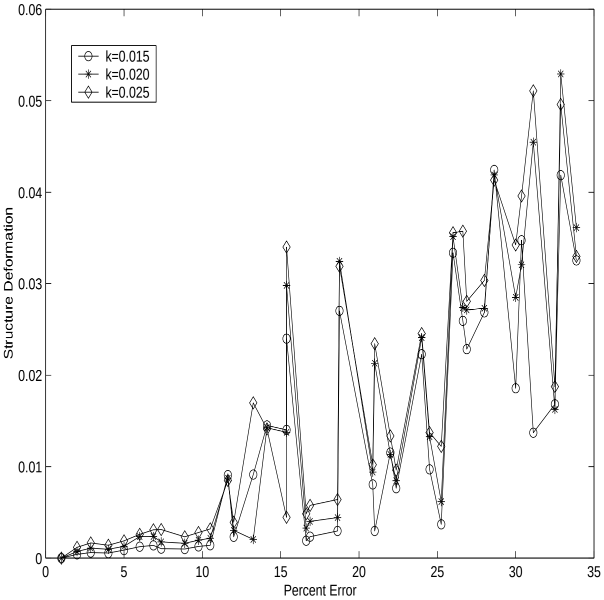


Figure 8.20: The experiment of Figure 8.18 repeated with different parameter settings for k in the truncated quadratic error function.

8.9 Acknowledgements

We would like to thank the Modeling by Videotaping group at the Robotics Institute, Carnegie Mellon University, for supplying the hotel and kitchen data sequences, and the accompanying tracked features [135]. We would also like to thank Radu Horaud and Claus Gramkow, for valuable comments to this work. And lastly Hans Bruun Nielsen for comments on the numerical aspects.

CHAPTER 9

Separation of Structure and Motion by Data Modification

by: Hans Bruun Nielsen and Henrik Aanæs

Preliminary Version

Abstract

Estimation of a subspace that contains the significant part of a given data set is used in many branches of data analysis and machine learning. A notable example is the principal component analysis (PCA), which can be used when all the data points have identical and independent Gaussian noise structure. In many cases, however, extended weighting structures are needed, and PCA is not applicable. We present a new method for the numerical solution of such non–trivially weighted subspace problems. The method is based on successive modification of the data, and is somewhat related to the widely used method of alternating least squares (ALS). Tests on artificial data show that the new method compares very favourably with ALS.

9.1 Introduction

Subspace learning and principal component analysis are different names for the same estimation of a linear subspace of ones data which express the significant part of it. The associated numerical operations are often performed via the singular value decomposition, SVD, which as such is sometimes seen as synonymous with the above mentioned methods. These methods have been used in most branches of data analysis, often with good results. The specific area we are interested in here is structure from motion cf eg [4, 89, 204].

The estimation of this subspace is usually done by minimizing the distance between the subspace – of varying or fixed dimensions – and the data, as measured by the Frobenius norm. However, this usually implies that all the observations in the data are weighted equally, that there are no missing data and that there is no cross correlation. Such implications or assumptions are infeasible in many real applications. Thus, considering the applicability of the method and its popularity, there is a need for more flexible frameworks, where arbitrary weight structures can be applied to the data. In statistical terms this corresponds to imposing an arbitrary correlation structure instead of a homoscedastic univariate.

This subspace fitting with non–trivial correlation structure is the focus of this paper, for which the initial motivation was the structure from motion problem[4]. Here the weighted subspace fitting is needed to impose robust statistics, deal with missing data and the uncertainty of the image feature location. In order to keep the paper concrete and easy to read we have chosen to concentrate on the structure from motion problem. This implies that the subspace to be estimated has dimension 3, but the results can easily be generalized to the wealth of other applications where SVD is used for subspace fitting.

Other work has been made addressing this important problem of weighted subspace estimation, for a good survey see [50]. A theoretical treatment is presented in [167], and a simple algorithm is proposed in [32] – where missing data are set to zero, thus inducing a bias. The problem has also received some attention within the neural network literature [113, 217]. A popular approach to the weighted subspace estimation is the alternating least squares (ALS) [4, 181, 203, 50, 215, 220], which is also in focus here. In [4, 51, 50] subspace estimation is done via robust techniques by use of iterative least squares (IRLS) [16, 96].

We shall give a thorough treatment and further development of the method proposed in [4]. This method allows for an arbitrary correlation structure of the individual observations – arranged as columns of the data matrix – and as such, the method is a versatile tool for data analysis and inference. The method can be considered as a further development of the weighted ALS, by proposing a method for circumventing the solution of the normal equations where the design matrix is the Kronecker product of the weights and the estimated projections of the data onto the subspace. This is a potentially very large matrix, and the proposed method has a potential of a considerable speed–up.

9.1.1 Structure from Motion

As mentioned structure from motion is the motivation for this work, hence a short overview is given her. Note, however, that a deeper insight into this subject is not required here.

Structure from motion is the estimation of the 3D structure of a rigid object and relative camera motion from 2D images of the structure, taken by a moving camera. This subject is popular within the computer vision community, and has among others spawned the so called factorization algorithms for solving the problem cf [4, 89, 204]. These factorization algorithms linearize the camera or observation models whereby the observed image features $\mathbf{S}$ can be expressed as

$$\mathbf{S} = \mathbf{M}\mathbf{P} \ .$$

$\mathbf{S}$ is an $m \times n$ matrix corresponding to n features observed in $\frac{1}{2}m$ frames[1]. $\mathbf{M}$ is a $m \times 3$ matrix representing the stacked camera or observation models, and $\mathbf{P}$ is a $3 \times n$ matrix corresponding to the 3D points to be reconstructed. The dimension 3 correspond to the space we live in. The goal is to estimate $\mathbf{M}$ and $\mathbf{P}$ from a given $\mathbf{S}$. This corresponds to finding the dominant subspace $\mathbf{M}$ of $\mathbf{S}$, see eg [204]. The observations in $\mathbf{S}$ might, however, have a covariance structure very different from identity, hence the motivation for this work.

9.2 Problem Overview

Assume, that m–dimensional observations S_i, $i = 1, \ldots, n$ have been arranged in a $m \times n$ matrix $\mathbf{S}$. In the structure from motion setting this represents the projection of a 3D feature P_i in the n frames[2]. The goal is to determine the 3D subspace behind the observations. This subspace is denoted by the $m \times 3$ matrix $\mathbf{M}$ representing the motion. If univariate homoscedastic Gaussian noise is assumed, this corresponds to minimizing

$$\min_{\mathbf{M},\mathbf{P}} \|\mathbf{S} - \mathbf{MP}\|_F^2 = \min_{\mathbf{M},\mathbf{P}} \sum_{i=1}^{n} \|S_i - \mathbf{M}P_i\|_2^2 \,, \tag{9.1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\mathbf{P}$ is a $3 \times n$ matrix

$$\mathbf{P} = \begin{bmatrix} P_1 & \cdots & P_n \end{bmatrix} \,.$$

Note, that a general issue of subspace estimation is that the solution is ambiguous. For any full rank 3×3 matrix $\mathbf{C}$,

$$\mathbf{MP} = \mathbf{MCC}^{-1}\mathbf{P} = (\mathbf{MC})\left(\mathbf{C}^{-1}\mathbf{P}\right) \,.$$

Hence $\mathbf{MC}$ is just a good a solution to the problem as $\mathbf{M}$. Let the SVD of $\mathbf{S}$ be

$$\mathbf{S} = \mathbf{U\Sigma V}^T \,, \quad \mathbf{\Sigma} = \operatorname{diag}(\sigma_1, \ldots, \sigma_{\min\{m,n\}}) \,,$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{m,n\}} \geq 0$, and let U_j and V_j denote the jth column of the orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. According to the famous theorem of Eckart and Young (see eg [121]) one way of expressing the solution to (9.1) is

$$\mathbf{MP} = \sum_{j=1}^{3} \sigma_j U_j V_j^T \,. \tag{9.2}$$

This corresponds to choosing $\mathbf{M}$ as the submatrix consisting of the first three columns of $\mathbf{U}$.

It is seen that arbitrary per observation weights or covariance structures can be achieved by extending (9.1) to

$$\min_{\mathbf{M},\mathbf{P}} \left\{ \varphi(\mathbf{M}, \mathbf{P}) = \sum_{i=1}^{n} \|\mathbf{W}_i \left(S_i - \mathbf{M}P_i\right)\|_2^2 \right\} \,, \tag{9.3}$$

[1] The reason m is divided by two is that an image observation is a 2-dimensional entity.

[2] The coordinates of P_i are unknown.

where $\mathbf{W}_i$ represents the weights associated with observation S_i. Note that $\left(\mathbf{W}_i^T\mathbf{W}_i\right)^{-1}$ can be interpreted as the covariance of the observations.

It is the solution of the optimization problem (9.3) that is the prime concern here. Unfortunately, SVD is not applicable in this case.

9.3 ALS Overview

A popular method for solving (9.3) is what can be described as ALS: Alternating Least Squares. The first publication on this seems to be [215]. This technique is based on the observation that if either $\mathbf{M}$ or $\mathbf{P}$ is known then (9.3) is just a weighted linear least squares problem in the other. This is then used in an iterative manner to obtain a solution. The algorithm can be outlined as:

1. Obtain an initial guess to the solution of (9.3), eg by means of (9.2).
2. Solve (9.3) for $\mathbf{P}$ given $\mathbf{M}$.
3. Solve (9.3) for $\mathbf{M}$ given $\mathbf{P}$.
4. If not stop goto 2.

Given $\mathbf{M}$ it is seen from (9.3) that the P_i are independent of each other. This implies, that each P_i can be estimated by solving the least squares problem

$$\min_{P_i} \|\mathbf{W}_i S_i - \mathbf{W}_i \mathbf{M} P_i\|_2^2 \ .$$

The estimation of $\mathbf{M}$ given $\mathbf{P}$ is unfortunately not as simple, but can be derived from (9.3) as follows. Let

$$P_i = \begin{bmatrix} p_{1i} \\ p_{2i} \\ p_{3i} \end{bmatrix} , \quad \widetilde{M} = \begin{bmatrix} M_1 \\ M_2 \\ M_3 \end{bmatrix} \ ,$$

where $M_1,\ M_2,\ M_3$ are the three columns in $\mathbf{M}$. For given $\mathbf{P}$ (9.3) is equivalent to

$$\begin{aligned} &\min_{\mathbf{M}} \sum_{i=1}^{n} \left\| \mathbf{W}_i (S_i - \sum_{k=1}^{3} p_{ki} M_k \right\|_2^2 \\ &= \min_{\widetilde{M}} \left(a - 2\widetilde{M}^T b + \widetilde{M}^T \mathbf{A} \, \widetilde{M} \right) \ , \end{aligned} \tag{9.4}$$

where a is the scalar

$$a = \sum_{i=1}^{n} S_i^T \mathbf{\Omega}_i S_i \ , \quad \mathbf{\Omega}_i = \mathbf{W}_i^T \mathbf{W}_i \ ,$$

b is the $3m$–vector

$$b = \sum_{i=1}^{n} \begin{bmatrix} p_{1i}\mathbf{\Omega}_i S_i \\ p_{2i}\mathbf{\Omega}_i S_i \\ p_{3i}\mathbf{\Omega}_i S_i \end{bmatrix} \ ,$$

and $\mathbf{A}$ is the $3m \times 3m$ matrix

$$\mathbf{A} = \sum_{i=1}^{n} \begin{bmatrix} p_{1i}^2 \mathbf{\Omega}_i & p_{1i}p_{2i}\mathbf{\Omega}_i & p_{1i}p_{3i}\mathbf{\Omega}_i \\ p_{2i}p_{1i}\mathbf{\Omega}_i & p_{2i}^2\mathbf{\Omega}_i & p_{2i}p_{3i}\mathbf{\Omega}_i \\ p_{3i}p_{1i}\mathbf{\Omega}_i & p_{3i}p_{2i}\mathbf{\Omega}_i & p_{3i}^2\mathbf{\Omega}_i \end{bmatrix} .$$

This matrix is symmetric and positive (semi)definite, and the minimizer of (9.4) (and thereby the next approximation to $\mathbf{M}$) is the solution to the linear system

$$\mathbf{A}\,\widetilde{M} = b\,.$$

Often, the matrix $\mathbf{A}$ contains special structure that allows for reductions in the computing time for setting up and solving the system. This, however, is not always true, and even for small problems this part of the ALS algorithm may be quite slow. In the next section we propose a new method for obtaining $\mathbf{M}$ given $\mathbf{P}$ that is considerably faster.

9.4 Data Modification Step

As mentioned above, we propose an alternative method for updating $\mathbf{M}$, which avoids the setting up and solution of a system with a $3m \times 3m$ matrix, cf $\mathbf{A}$. The essence of the method is to approximate the weighted problem by an unweighted one, by modifying the data. Specifically, define the residuals N_i, modified residuals $\tilde{N}_i$, and modified data $\tilde{S}_i$,

$$\begin{aligned} N_i &= S_i - \mathbf{M}P_i\,, \\ \tilde{N}_i &= \mathbf{W}_i N_i\,, \\ \tilde{S}_i &= \mathbf{M}P_i + \tilde{N}_i\,. \end{aligned} \tag{9.5}$$

Then, for given $\mathbf{P}$ (9.3) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{M}} \sum_{i=1}^{n} \|\mathbf{W}_i N_i\|_2^2 \\ &= \min_{\mathbf{M}} \sum_{i=1}^{n} \|\mathbf{M}P_i + \tilde{N}_i - \mathbf{M}P_i\|_2^2 \\ &= \min_{\mathbf{M}} \sum_{i=1}^{n} \|\tilde{S}_i - \mathbf{M}P_i\|_2^2\,. \end{aligned} \tag{9.6}$$

The reformulation is illustrated in Figure 9.1.

Now, we change the interpretation slightly: For a given approximation to $\tilde{\mathbf{S}}$ we have a simple problem of the form (9.1) and use (9.2) to get the corresponding approximation to $\mathbf{M}$. The algorithm can be outlined as follows:

1. Obtain an initial guess of $\tilde{\mathbf{S}}$.
2. Use (9.2) with $\mathbf{S}$ replaced by $\tilde{\mathbf{S}}$ to compute $\mathbf{M}$.

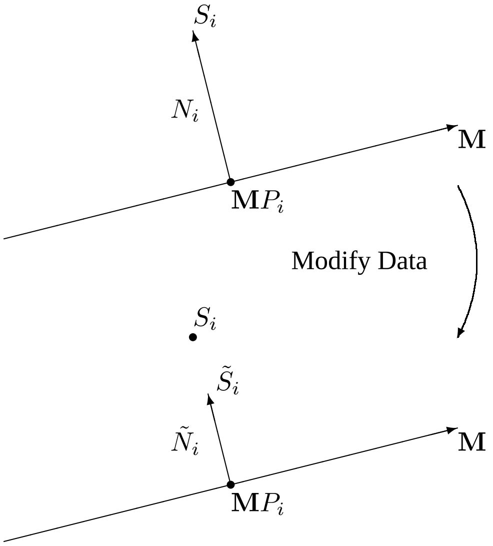


Figure 9.1: A geometric illustration of the data modification.

3. Solve (9.3) for $\mathbf{P}$ given $\mathbf{M}$.
4. Use (9.5) to compute $\tilde{\mathbf{S}}$.
5. If not stop goto 2.

9.5 Experimental Results

9.5.1 Test Problems

We have developed a problem generator that returns weight matrices $\mathbf{W}_i$ and a data matrix $\mathbf{S}$ of desired size and (approximate) noise-to-signal ratio.

The weight matrices have the form

$$\mathbf{W}_i = \mathbf{R}_i^T \mathbf{D}_i \mathbf{R}_i \ ,$$

where the $\mathbf{R}_i$ are rotation matrices (and therefore orthogonal), the $\mathbf{D}_i$ are diagonal with uniform random elements in $]0,1[$. “Wild points” can be introduced by letting some of these elements be especially small.

The data matrix is computed as

$$S_i = \mathbf{M}P_i + \mathbf{W}_i^{-1}\tilde{N}_i \ ,$$

where

$$\begin{aligned}
\mathbf{M} &= \begin{bmatrix} U_1 & U_2 & U_3 \end{bmatrix} \ , \\
\mathbf{P} &= \begin{bmatrix} \sigma_1 V_1 & \sigma_2 V_2 & \sigma_3 V_3 \end{bmatrix}^T \ , \\
\tilde{\mathbf{N}} &= \sum_{j=4}^{r} \sigma_j U_j V_j^T, \quad r = \min\{m, n\} \ .
\end{aligned}$$

Each of the sets of m–vectors U_j and n–vectors V_j are orthonormal. $\sigma_1 = 1$, $\sigma_2 = \sqrt{\sigma_3}$ (with $\sigma_3 < 1$ given as input; we use $\sigma_3 = 0.8$ in the examples given) and the "noise components" decay logarithmic from $\sigma_4 = \gamma\sigma_3$ ($\gamma \in]0,1[$ is input) to $\sigma_r = 10^{-3} \cdot \sigma_4$.

The tests were performed using MATLAB 6.5 on a Dell computer with a 1.99 GHz CPU.

9.5.2 Preliminary Results

First, Figure 9.2 illustrates a typical behaviour of the two methods, ALS and DM (Data Modification) in a case where there is no wild points and a small noise-to-signal ratio. The objective function is given by (9.3), ie

$$\varphi_k = \sum_{i=1}^{n} \|\mathbf{W}_i \left(S_i - \mathbf{M}_k P_i^{(k)} \right) \|_2^2 \ , \tag{9.7}$$

where $\mathbf{M}_k$ and $\mathbf{P}_k$ is the kth approximation to $\mathbf{M}$ and $\mathbf{P}$, respectively. Note that ALS converges in fewer iterations than DM, but the latter is faster.

A closer examination shows that (when they converge) both methods exhibit an asymptotic behaviour of the form

$$\varphi_k \simeq \alpha + \beta\eta^k \ , \tag{9.8}$$

where α, β and η are positive and $\eta < 1$. Thus, α is the limit as $k \rightarrow \infty$, and in both algorithms one of the stopping criteria is

$$\varphi_k - \alpha_k \leq \varepsilon \cdot \varphi_k \ ,$$

where ε is the desired relative accuracy and α_k is the value for a as estimated from $\varphi_{k-4}, \ldots, \varphi_k$. This criterion is applied only, if the model (9.8) matches these 5 values sufficiently well. In the examples we use $\varepsilon = 10^{-4}$, and with the same data as in Figure 9.2 we need 9 iterations with ALS and 21 iterations with DM (but the latter still has a smaller execution time).

9.5.3 Enhancements of the DM Algorithm

The starting guess $\tilde{\mathbf{S}}_0 = \mathbf{S}$ does not exploit the possibilities in the data modification philosophy. Returning to the definition in (9.5), we see that it might be better to use $\tilde{\mathbf{S}}_0$ computed by

$$\tilde{S}_i^{(0)} = \mathbf{W}_i S_i, \quad i = 1, \ldots, n \ . \tag{9.9}$$

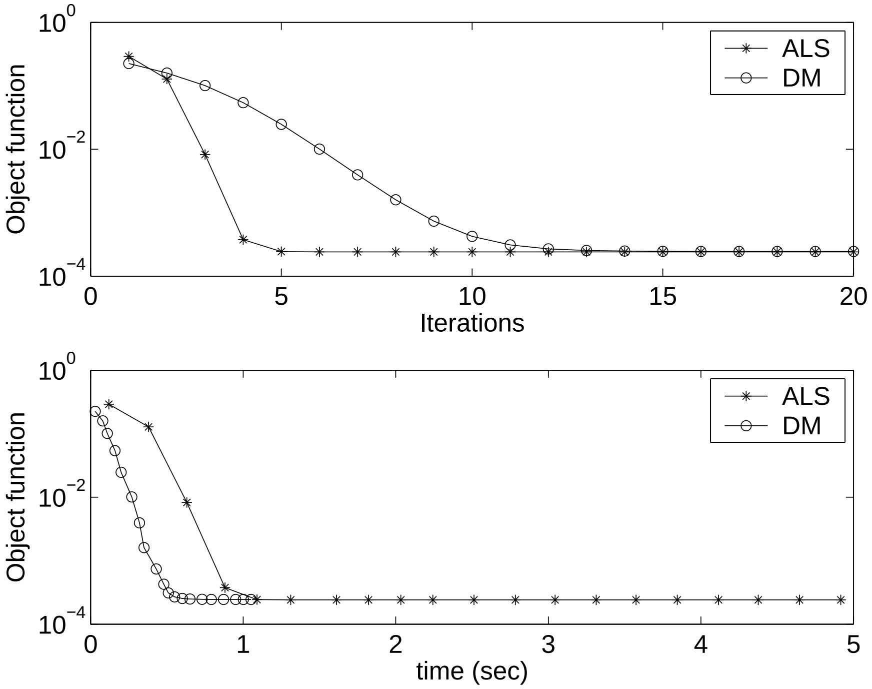


Figure 9.2: $(m, n, \gamma) = (100, 50, 0.01)$, $\tilde{\mathbf{S}}_0 = \mathbf{S}$.

This corresponds to starting by considering all of $\mathbf{S}$ as "noise", and has the effect, eg, that columns with relatively small elements in $\mathbf{W}_i$ have small influence on the first approximation to $\mathbf{M}$. If we use this starting guess on the previous problem, the number of iterations with DM reduces from 21 to 13.

Figure 9.3 illustrates the behaviour for a problem with larger noise-to-signal ratio. The DM method with the starting guess (9.9) is almost unaffected by the larger noise, while the necessary number with ALS is more than doubled.

Finally, Figure 9.4 illustrates the behaviour for a problem with wild points. Again, DM is almost unaffected, while ALS has to show great perseverance in order to solve this problem.

The monotone convergence indicated by (9.8) suggests that it might be worthwhile to use overrelaxation, eg in the updating of $\tilde{\mathbf{S}}$:

$$\tilde{\mathbf{S}}_{k+1} = \tilde{\mathbf{S}}_k + \omega(\mathbf{M}_k\mathbf{P}_k + \tilde{\mathbf{N}}_k - \tilde{\mathbf{S}}_k) \; . \tag{9.10}$$

If we use this with $\omega = 1.5$ on the data in Figure 9.4, the number of iterations with DM is reduced from 15 to 5. We also tried a similar overrelaxation on the sequence of $\mathbf{M}_k$ and/or $\mathbf{P}_k$, but had no success.

The starting point for ALS is given by the SVD of $\mathbf{S}$. If, instead, we use the SVD of $\tilde{\mathbf{S}}_0$ given by (9.9), we get a similar improvement of ALS. For this method, however, the rationale

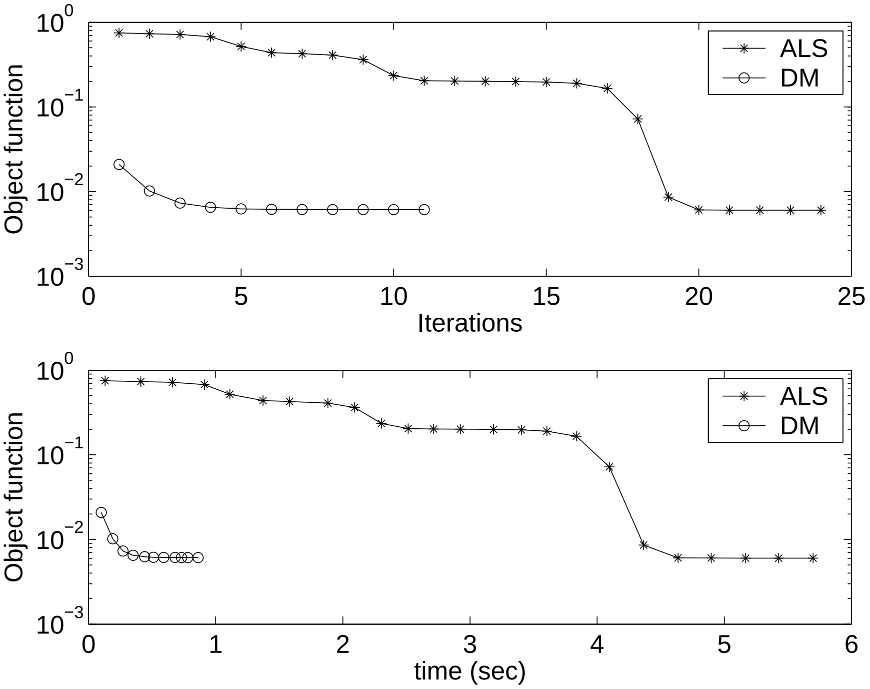


Figure 9.3: $(m, n, \gamma) = (100, 50, 0.05)$. $\tilde{\mathbf{S}}_0$ given by (9.9).

for the improved starting guess is less obvious. Since simple overrelaxation applied to the factors $\mathbf{M}_k$ and $\mathbf{P}_k$ did not work with DM, there is no reason to expect that it would work with ALS.

9.6 Discussion and Conclusion

We have proposed a new method for solving the subspace estimation problem for general weighting. The new method compares very favourably with the widely used method of ALS, alternating least squares.

The new method is presented in connection with the structure and motion problem, ie for fixed dimension $d = 3$. It is, however, straightforward to generalize the method to other values of d. If $d > 3$, then the potential gain as compared with ALS is even larger: the matrix $\mathbf{A}$ in (9.4) will have dimension dm.

The results quoted are for ideal test problems: In every column of the data matrix the weight matrix $\mathbf{W}_i$ and noise N_i are such that $E[(\mathbf{W}_i N_i)(\mathbf{W}_i N_i)^T] = I$. It needs further investigation to see whether the method is also advantageous in less favorable circumstances, eg in connection with iteratively reweighted least squares (IRLS). Other points that are still

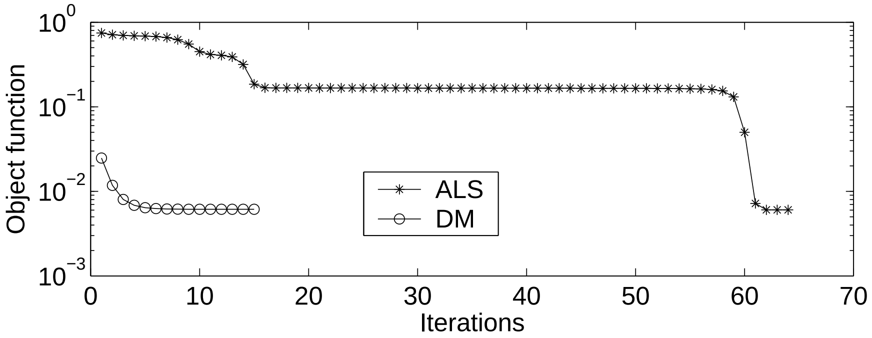


Figure 9.4: $(m, n, \gamma) = (100, 50, 0.05)$. 5% wild points. $\tilde{\mathbf{S}}_0$ given by (9.9).

missing are

- A theoretical analysis of the method.
- An experimental and theoretical investigation of its robustness.
- Development of a public, user friendly implementation and user's guide.
- Writing an internal report that documents some of the more interesting details of the implementation, the testing, and choice of default values for parameters such as ε and ω.

CHAPTER 10

Integrating Prior Knowledge and Structure from Motion

by: Nicolas Guilbert, Henrik Aanæs and Rasmus Larsen

Abstract

A new approach for formulating prior knowledge in structure from motion is presented. The structure is viewed as a 3D stochastic variable, whereby priors are more naturally expressed. It is demonstrated that this formulation is efficient for regularizing structure reconstruction via prior knowledge. Specifically, algorithms for imposing priors in the proposed formulation are presented.

10.1 Introduction

Structure from motion is a field of research which allows the reconstruction of the 3D structure and motion of a rigid body from a sequence of two or more images where 2D features of interest have been detected and correlated through the frames, see e.g. [89]. It is an essential tool in artificial intelligence and image understanding, applicable in robot vehicle navigation, content-based searches in video data or architectural visualization.

However, it is an inherent property of the approach, that the accuracy of the 3D reconstruction is underpinned by the noise in the images, be it due to quantization or shortcomings of the feature extraction algorithm. Hence, enhancing the 3D structure via prior knowledge can improve the results considerably. For instance, knowing that a 3D object primarily consists of planes at right angles to each other, like eg. buildings, can greatly enhance its reconstruction. One might also envisage integrating more sophisticated models along the line of [19, 45], whereby more complex priors can be expressed and applied.

Previously, Baker et al. [168] and Torr et al. [148] have addressed the problem of incorporating prior knowledge into structure from motion. These approaches identify *layers* in the images and constrain the reconstruction accordingly. These methods use backprojection onto the image(s) to determine whether regularization with a given prior is probable. Consequently, the noise is expressed in the 2D image domain, and it is implied that the camera matrices are determined perfectly.

In this paper, we address the problem of combining priors with structure from motion by formulating the estimated reconstruction as a 3D stochastic variable. This approach has the advantage of being intuitively accessible, since the 3D structure is computed explicitly. The explicite reconstruction also allows handling of the uncertainty of the camera motion, since our approach can incorporate that the noise on a given 2D feature potentially affects all of the 3D features. Finally, integration of more complex priors than planes (eg. splines or deformable templates) is natural.

We establish the mean of the 3D structure by standard structure from motion methods, see e.g. [89]. The dispersion of the resulting structure is then obtained by approximating the 2D to 3D transformation by a linear function, whereby the relations between noise on the 2D features and the 3D structure becomes apparent. In order to validate our approach, we then impose the prior of planar surfaces into structure from motion, applying a clustering technique.

The organisation of this paper is as follows: Section 10.2 describes the stochastic structure modeling approach, Section 10.3 describes the algorithm for fitting a plane to a set of points given heteroscedastic and anisotropic noise. Section 10.4 concerns the clustering of smaller planes into the final plane estimates.

10.2 Stochastic Structure Model

As mentioned, we view the estimated 3D structure as a stochastic variable, derived from the the tracked 2D feature points in two or more images. We assume a calibrated pinhole camera, so for a given point $j \in \{1, \ldots, n\}$:

$$\begin{bmatrix} x_{ij} \\ y_{ij} \\ s_{ij} \end{bmatrix} = \mathbf{A}_i \begin{bmatrix} \mathbf{R}_i & t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_j \\ 1 \end{bmatrix} = \mathbf{A}_i \mathbf{P}_i \begin{bmatrix} X_j \\ 1 \end{bmatrix} \tag{10.1}$$

where the 2D feature (x_{ij}, y_{ij}), is the projection of the 3D feature X_j, in image $i \in \{1, \ldots, m\}$. Image i is described by its calibration matrix $\mathbf{A}_i$, its rotation $\mathbf{R}_i$ and translation t_i relative to the world coordinate system.

From the tracked 2D features, and this observation model (10.1), it is well known that the 3D structure, X_j and the camera motion, P_i, can be calculated, see e.g. [39, 89], .

As such, assuming a well posed problem, a given 3D feature X_j can thus be seen as a function, $\mathcal{F}_j$, of the tracked 2D features:

$$\begin{aligned} \mathbf{x} = [&x_{11}, \ldots, x_{1n}, \ldots, x_{m1}, \ldots, x_{mn}, \\ &y_{11}, \ldots, y_{1n}, \ldots, y_{m1}, \ldots, y_{mn}] \end{aligned}$$

such that each component of $\mathbf{x}$ is an image coordinate, i.e.

$$\mathcal{F}_j : \mathbf{x} \rightarrow X_j$$

So, considering X_j and $\mathbf{x}$ as stochastic variables with assumed Gaussian noise such that:

$$X_j \in N(\mu_j, \Sigma_j) \qquad \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} \in N(\mu_{ij}^{2D}, \begin{bmatrix} \sigma_x \; 0 \\ 0 \; \sigma_y \end{bmatrix}_{ij})$$

we would like to estimate X_j given $\mathbf{x}$. Since we do not have $\mathcal{F}_j$ in closed form, we approach this problem by linearizing $\mathcal{F}_j$, by means of numerical gradients. Hence:

$$\begin{aligned} X_j \in \mathcal{F}_j(\mathbf{x}) &\approx \mathcal{F}_j(\mu_{\mathbf{x}}) + \sum_k \frac{\partial \mathcal{F}_j}{\partial z_k} \Delta z_k \Rightarrow && (10.2) \\ Var(X_j) &\approx \sum_k \left(\frac{\partial \mathcal{F}_j}{\partial z_k} \right) \left(\frac{\partial \mathcal{F}_j}{\partial z_k} \right)^T Var(z_k) && (10.3) \\ E(X_j) &= \mathcal{F}_j(\mathbf{x}) && (10.4) \end{aligned}$$

where z_k denotes an elements of $\mathbf{x}$. And it is assumed, that the noise on $\mathbf{x}$ is independent. Hence from (10.3) the Gaussian noise structure of the X_j can be derived.

As noted, this derived noise structure of the X_j, is based on the approximation of (independent) Gaussian noise, and of linearizing $\mathcal{F}_j$. These assumptions seam reasonable, in that the approach is capable of capturing, how variations in the image data, $\mathbf{x}$, affects the estimated structure, X_j. These effects on the estimated structure, X_j, are all what is reasonable to expect, since the noise structure on the 2D features is rarely known and as such all distributions – e.g. Gaussian – are approximations.

Our assumption of a calibrated camera – $\mathbf{A}_i$ known – is by no means necessary, as seen in [89], except if the imposed priors are non invariant to projective transformations, which they will seldom be.

10.3 Estimating the Plan (Prior)

In order to test if a set of given 3D features belongs to the same plane, and if so enforce that plane upon the the structure, we need to estimate the most likely plane. In this case it is non–trivial, since the 3D features have different or heteroscedastic anisotropic noise. This implies that the Mahalanobis distance measure or norm for each 3D feature is different.

Hence given m 3D features $\{X_1, \ldots, X_m\}$, and corresponding Gaussian variance structures $\{\Sigma_1, \ldots, \Sigma_m\}$ we want to find the plane that minimizes the distance between 3D feature X_j and the plane, in the norm induced by the Σ_j^{-1}. Thus, let a plane be denoted by its normal vector, π, and its offset from origo, α_0, then any point, X, on the plane satisfies:

$$\pi^T \cdot X + \alpha_0 = 0$$

The most likely plane is estimated by iteratively approximating the noise structures by heteroscedastic *isotropic* noise, until convergence is achieved. In this isotropic case, the most

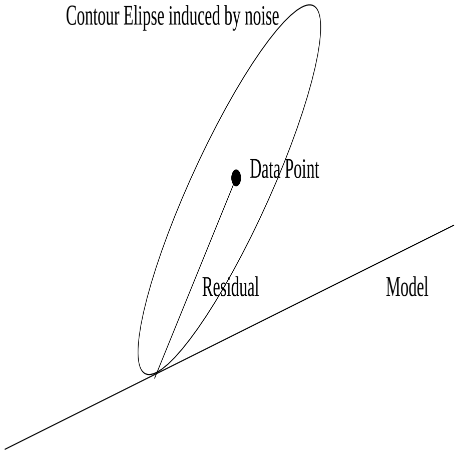


Figure 10.1: It is seen that the induced norm changes the concept of orthogonal, and as such the residual, $\tilde{r}$, is not perpendicular to the plane in the usual sense.

likely plane is given by:

$$\begin{aligned} \alpha_0 &= -\frac{\sum_{j=1}^m \sigma_j X_j}{\sum_{j=1}^m \sigma_j} \\ \pi &= \min_{\pi'} \sum_{j=1}^{m} ||\sigma_j (X_j + \alpha_0)^T \cdot \pi'||_2^2 \end{aligned} \tag{10.5}$$

where σ_j denotes the isotropic variance of Σ_j. These σ_j – or the approximated noise – are naturally derived from the residual between the 3D feature and the fitted plane, but in the induced norm. The latter is quite cumbersome, for as seen in Figure 10.1, this alters the concept of orthogonality. So given a plane, the σ_j are given as the ratio between the length of the minimum distance between the 3D feature and the plane, $\tilde{r}_j$, in the induced norm relative to the minimum distance in the 2-norm, r_j, i.e.

$$\sigma_j = \frac{\tilde{r}_j^T \Sigma_j^{-1} \tilde{r}_j}{r_j^T r_j} \tag{10.6}$$

In short, the algorithm is, where q denotes iterations:

1. **Initialize** $q = 0$, $\forall j \ \sigma_j^0 = \sqrt[3]{det \Sigma_j}$.

2. **Estimate Plane** $\pi^q, \alpha_0^q$with isotropic noise (10.5).

3. **Update Isotropic Noise** σ_j^q via (10.6)

4. **If not Stop**, $q = q + 1$, goto 2. The stop criteria is

$$\max_j(\sigma_j^q - \sigma_j^{q-1}) < tolerance.$$

It is seen, that the σ_j are updated such that the given plane has the right likelihood, with the original heteroscedstic *an*isotropic noise. So if $\forall j\ \sigma_j^q = \sigma_j^{q-1}$ then the optimal plane fitted with isotropic noise is also optimal in the *an*isotropic case.

10.4 Imposing the Prior

As mentioned, we validate our approach by imposing the prior of planar surfaces onto our structure. This is done by first triangulating the estimated 3D structure, whereby a set of three–point planes are constructed. A recomended methods for triangulation is [137] by Morris and Kande.

As such, this triangulation does not reguaralize the 3D structure, but it serves as a initialization for an algorithm where neighboring planes (e.g triangles) which are likely to be the same plane are merged. A statistical test for coplanarity is described below.

10.4.1 Test for Coplanarity

Given a set of 3D features, as described above, located on a plane, this plane can be estimated optimally as described in Section 10.3, then the residuals are realizations of the respective noise, $N(0, \Sigma_j)$, as estimated in Section 10.2. Hence evaluating the residuals, $\tilde{r}_j$ is a good test of the set of 3D features $\mathbf{X}$ being located in the same plane.

More formally, if the set $\mathbf{X}$ is located in the same plane, then the normalized residuals squared are part of a χ^2 distribution. For if this assumption of coplanarity holds, these residuals are elements in a $N(0, 1)$ distribution. And hence:

$$\sum_{j=1}^{m} \tilde{r}_j^T \Sigma_j^{-1} \tilde{r}_j \ \in \ \chi^2(m-3) \tag{10.7}$$

where -3 originates from the three degrees of freedom used for fitting the plane.

10.4.2 Clustering Algorithm

The algorithm for combining neighboring planes is a greedy clustering algorithm in that at any given time, the two neighboring planes with the highest likelihood of being the same plane are combined, if this likelihood is above a given threshold. Hence the triangles achieved by triangulation, are clustered into planes.

When these clusters have been calculated, the derived planes are enforced upon the 3D features, and the 3D features moved onto the plane, whereby the prior of planar structures is imposed. See Figure 10.2

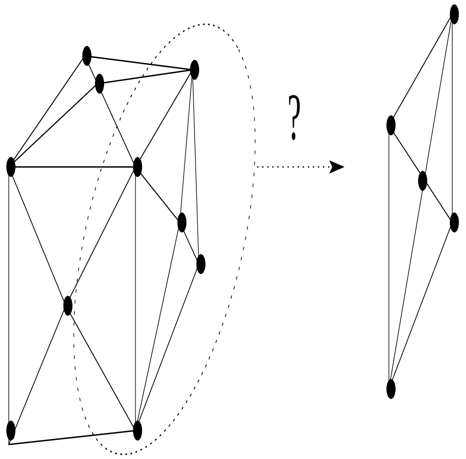


Figure 10.2: We check if a set of 3D features are likely to be coplanar, and if such we inforce the planar prior.

10.5 Results

We tried our approach on 3 simulated images of a box with added Gaussian noise, see Figure 10.3, where each side consisted of four equal triangles. Here we have used the method described in [39] to estimate the structure and motion, and the noise structure as seen in Section 10.2, along with the rest of our approach.

On this case it was noted, that 2D feature k had a considerable effect on 3D feature l for $k \neq l$, hence the estimated motion of the box, was highly susceptible to noise, a property that was captured by our approach. It was also noted, that the clustering achieved the desired result, in that the triangels belonging to the same side were correctly grouped, see Figure 10.4.

Lastly it is noted that, the prior of planar surfaces was enforced correctly, since all the faces of the reconstructed and regularized box are planar, see Figure 10.4, and hence the algorithm worked as expected.

10.6 Conclusion

We have presented a new formulation of applying prior knowledge to 3D structure and motion reconstruction. This approach has the advantage that it is formulated in 3D where it is more natural to work with prior knowledge of 3D objects. We have succesfully applied the approach on simulated data and hereby described an algorithm for imposing planar surface priors on a 3D structure.

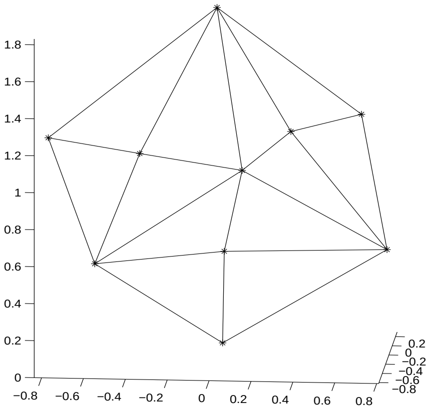


Figure 10.3: Reconstruction from the noisy 2D features, *without* use of the prior – i.e. planar surfaces.

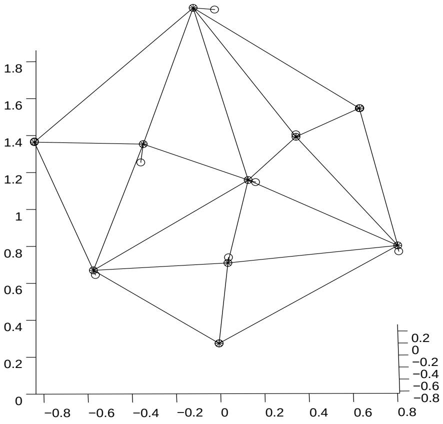


Figure 10.4: Reconstruction *with* use of the prior. The residuals from the structure in Figure 10.3 are indicated by line segments.

10.7 Further Work

We are presently planning to extend the approach to deal with multiple types of structural ’primitives’, e.g. planes and right angles among planes. We would also like to incorporate prior knowledge of the texture and hereby construct a more general framework.

10.8 Acknowledgments

We thank Allan Aasbjerg Nielsen for enlightening discussions on plane fitting with heteroscedastic anisotropic noise.

CHAPTER 11

PDE Based Surface Estimation for Structure from Motion

by: Henrik Aanæs, Rasmus Larsen and J. Andreas Bærentzen

Abstract

The method of Faugeras and Keriven for solving the multiple view stereo problem by partial differential equations (PDE's) in a level set implementation has been well received. The main reasons are that it produces good results and deals effectively with objects which are, topologically, more complex than a ball. This makes it a good choice for the final step in the usual structure from motion approach, namely making a full 3D surface reconstruction from estimates of camera orientations and 3D point structure.

Here an approach is proposed whereby the 3D point structure, estimated using structure from motion, is used to initialize the method of Faugeras and Keriven. The proposed approach has the advantage of a) considerably improving the run time of the approach, and b) making it more resistant towards noisy data and data with patches of low variability. Both advantages make the approach much more effective on real data.

11.1 Introduction

The estimation of surface models from multiple images, the so–called multiple view stereo problem, is one of the classical problems within computer vision (e.g. [66, 119, 145, 225]). Some of the best results are achieved by Faugeras and Keriven by posing the problem as a partial differential equation PDE [64, 65, 66], this has been further developed in [106]. Some of the merits of this approach are, that it in a naturally way employs all the images simultaneously as opposed to the adapted 2 image stereo approaches (e.g. [145, 225]) and that it is capable of dealing with objects of arbitrary topology.

Another major problem of computer vision is the reconstruction of structure and motion of a rigid object from an image sequence, the so–called structure from motion problem. Here the usual approach is to first restrict the 3D structure estimation to a few distinct points whereby the structure is represented as a point cloud (e.g. [89]). As a part of this point based structure from motion the camera calibration is also determined – outer and often also inner orientation. But for many applications of structure from motion a full 3D model is needed, and hence a multiple view stereo approach is natural (e.g. [67, 157]).

We propose using the PDE based surface estimation approach of Faugeras and Keriven [66] to solve the surface estimation problem from the structure and motion solution based on points. The contribution of this work is altering the approach by using the structure estimates, in the form of the 3D points, to much better initialization of the PDE based surface estimation algorithm. This has the effect of reducing the running time of the algorithm considerably. On a standard 1 GHz PC the order of magnitude is from whole and half days to an hour or often less. Secondly, the proposed approach also renders the algorithm much more resistant to noisy and/or erroneous data, as well as to objects with patches of low variance. Both these issues make the approach more effective with real data.

11.2 PDE Based Surface Estimation

As a courtesy to the reader and to introduce notation, a short overview of PDE based surface estimation is presented. For a more through introduction the reader is referred to [64, 65, 66, 106].

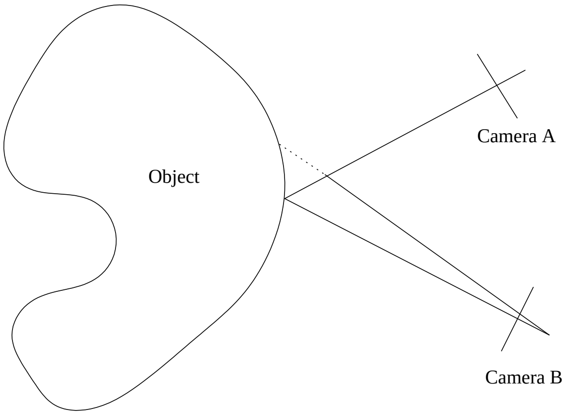


Figure 11.1: Based on the Lambertian assumption, the projection of a given surface point, should look the same in different images. A property that will, in general, not hold for image points being projections of different 3D points.

The main idea behind PDE based surface estimation is illustrated in Figure 11.1, namely that the projection of points on the physical surface should look the same in all images. The implication is that the true physical surface, $\mathcal{S}^*$, is the minimizer of:

$$\mathcal{S}^* = \arg\min_{\mathcal{S}} \int_{\mathcal{S}} \Phi(\mathbf{x})dA \quad , \tag{11.1}$$

where $\Phi(\mathbf{x})$ is a similarity measure of the projection of $\mathbf{x}$ in all the cameras. The *basic* similarity measure used in [64, 65, 66, 106] and here is; given neighborhoods of the projection of $\mathbf{x}$ in camera i, denoted $\mathcal{N}_i$. Then

$$\Phi(\mathbf{x}) = \frac{1}{\sum_{i\neq j} v_{ij}(\mathbf{x})} \sum_{i\neq j} \rho(\mathcal{N}_i,\mathcal{N}_j)\cdot v_{ij}(\mathbf{x}) \quad , \tag{11.2}$$

where $\rho(\cdot,\cdot)$ denotes the correlation and $v_{ij}(\mathbf{x})$ is an indicator function denoting whether $\mathbf{x}$ is visible in both image i and j or not.

The PDE determining the gradient flow of $\mathcal{S}$ is then:

$$\frac{\partial\mathcal{S}}{\partial t} = -\nabla\left(\Phi(\mathbf{x})\right)^T\cdot\mathbf{n}+\alpha\mathcal{C} \quad , \tag{11.3}$$

where $\mathbf{n}$ is the local normal to the surface $\mathcal{S}$, $\mathcal{C}$ is the local curvature, which is basically a second order smoothing term. The constant α determines the amount of smoothing imposed. Since the neighborhood patches, $\mathcal{N}_i$, are warped according to the local orientation of the surface at $\mathbf{x}$, a term dependent on $\mathbf{n}$ should also be present in (11.3). As noted in [106], this missing term is so similar to the local curvature that it can be dropped[1]. This is done here.

It should be noted, that Jin et al. [106] use the median instead of the mean in (11.2). Hereby an approach which is robust towards a break–down of the Lambertian assumption is obtained.

11.2.1 Implementation

The PDE of (11.3) is used to optimize (11.1) in a level set framework [146, 147, 174]. An important issue in this regard, is how to initialize the optimization scheme. The usual initialization scheme [64, 65, 66, 106], is to use a bounding ball or box, which contains the surface with probability 1 (c.f. Figure 11.2).

An issue with this approach is it speed. The evaluation of a given voxel of the level set grid is very costly, in that each patch, $\mathcal{N}_i$, should be warped in all images, and pairwise correlation between these patches should be estimated. Due to the warping, pre–calculation of intermediate results is infeasible. Hence, if a large change of volume is required, as illustrated in Figure 11.2, many costly evaluations are needed, and the approach will be very time consuming. In order to address this problem it is proposed [66] to add an inward force to (11.3). This inward force is set to zero if $\Phi(\mathbf{x}) > \Phi_{\max}$, such that it is ’turned off’ when the fit is ’good’. This has the advantage of speeding up the approach, and as far as we can

[1] This slightly changes the interpretation of α.

see, it also help avoid local minima, in that at such local minima it can be assumed that $\Phi(\mathbf{x}) < \Phi_{\max}$. Hence (11.3) is in reality changed to:

$$\frac{\partial \mathcal{S}}{\partial t} = -\left(\nabla\Phi(\mathbf{x})\right)^T \cdot \mathbf{n} + \alpha\mathcal{C} + \mathcal{G}(\Phi(\mathbf{x})) \ ,$$

where $\mathcal{G}(\Phi(\mathbf{x}))$ is the inward force. Even with this inward force, an optimization is not unlikely to take the better part of a day on a standard PC, due to the large change required in $\mathcal{S}$.

The inward force $\mathcal{G}(\Phi(\mathbf{x}))$ has some unintended side effects, since $\Phi(\mathbf{x})$ can be small on the true optimal surface $\mathcal{S}^*$. This fact is usually caused by image noise and/ or a low variability on part of the object in question. In cases of small $\Phi(\mathbf{x})$ on the true surface, $\mathcal{S}^*$, $\mathcal{G}(\Phi(\mathbf{x}))$ will cause the evolving surface, $\mathcal{S}$, to pass through $\mathcal{S}^*$. Note that

$$\alpha\mathcal{C} + \mathcal{G}(\Phi(\mathbf{x})) \ ,$$

acts as a prior on the surface, and as such determines the behavior of the algorithm when $\nabla\Phi(\mathbf{x})$ is small (c.f. Figure 11.4).

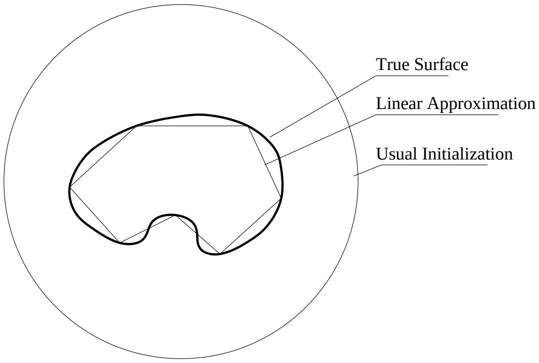


Figure 11.2: The true surface and different initializations of the surface before optimization. It is seen, that the Area – or Volume in 3D – to be traversed for convergence is usually much smaller for the linear approximation.

11.3 Utilizing the 3D Point Structure

In the usual structure from motion setting, the camera orientations and a 3D point structure is estimated first. These estimates are then used as the basis of a full 3D surface model estimation. If the method of [64, 65, 66, 106] was used for this, only the camera orientation would be applied. Hence the information from the already estimated structure is discarded.

We here propose using this 3D point structure by forming a much better initial guess. This is done by applying the method of Morris and Kanade [137], to make an optimal triangulation of the 3D point structure already at hand. This triangulated mesh is then used as an initialization for the level set. Since the elements of the 3D point structure are assumed to lie on the surface, this mesh can be seen as a piecewise linear approximation to the surface (c.f. Figure 11.2). A practical consideration in this regard is how to convert the mesh to a signed distance field, the latter being an initialization for the level set method. This is done with a modified version of [150] described in [26], whereby the signed distance from each voxel to the mesh is calculated. The sign is positive if the voxel is outside the mesh, and negative if it is inside.

The first advantage of this approach is that the initial guess is very likely to be much closer to the true surface, c.f. Figure 11.2. Hence the run time is reduced considerably, in that smaller changes in $\mathcal{S}$ is need . In our experiments the order of magnitude of speed up is from whole to half days to approximately 1 hour.

Secondly, the proposed approach has the advantage of not needing the inward force $\mathcal{G}(\Phi(\mathbf{x}))$. Hence the prior on the surface becomes:

$$\alpha\mathcal{C} \quad . \tag{11.4}$$

The difference is that if the data does impose a force on $\mathcal{S}$, due to the above mentioned reasons, then it should try to smooth out instead of go inward. This makes the approach much more resistant to image noise and surface parts with low variability.

11.4 Results

To validate the proposed approach, we used 5 images of a face as illustrated in Figure 11.3(a). This data set was noisy, in that it is unlikely that the subject was completely still. Secondly, there are many patches with very low or no variance. As such this is a rather challenging data set for a surface reconstruction algorithm, but it is by no means below the standard of what is expected in a structure and motion setting. The quality of the data also makes a solution with an inward force, $\mathcal{G}$, infeasible as seen in Figure 11.4. It is seen how the surface has ’gone through’ the true surface, $\mathcal{S}^*$. In later iterations the holes get bigger, and eventually the smoothness constraint of (11.3) will pull the surface from $\mathcal{S}^*$, whereupon it will collapse under its own curvature.

The proposed approach was also applied. It was initialized with the mesh depicted in Figure 11.3(b). This mesh is optimally triangulated based on a 3D point structure estimated by structure from motion from a series of automatically identified landmarks. The result is seen in Figure 11.5(a), where it is noted that the algorithm converges to an acceptable result, despite the quality of the data.

To improve the results, the use of more advanced regularization was investigated. It turned out that most of the problematic data was at low variance patches. Hence, it was tried to smooth patches, $\mathcal{N}_i$, with low $\Phi(\mathbf{x})$ more. From a histogram of $\Phi(\mathbf{x})$, it was deducted that 0.5 was a good cut off. Hence (11.3) was modified to

$$\frac{\partial\mathcal{S}}{\partial t} = \begin{cases} -\nabla\Phi(\mathbf{x})\cdot\mathbf{n} + \alpha\mathcal{C} & \Phi(\mathbf{x}) \geq 0.5 \\ -0.5\cdot\nabla\Phi(\mathbf{x})\cdot\mathbf{n} + 1.5\cdot\alpha\mathcal{C} & \Phi(\mathbf{x}) < 0.5 \end{cases} \quad . \tag{11.5}$$

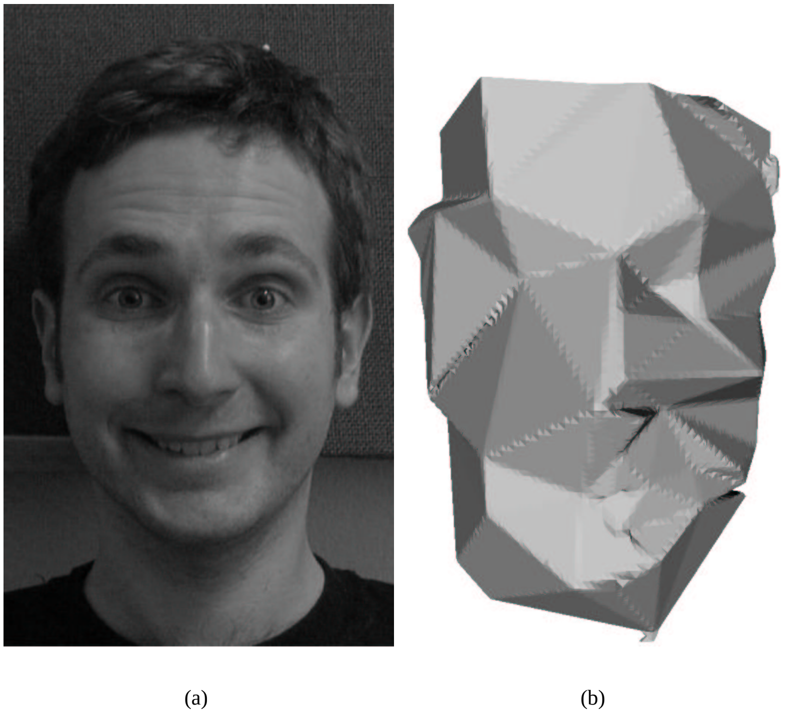


Figure 11.3: (a) A sub image from the face sequence of 5 images. The subject moved slightly during the taking of the images. (b) The initialization used with the face data set, when the proposed method is applied.

The result is seen in Figure 11.5(b). In the results of Figures 11.5(a) and 11.5(b) α was set to 0.25. Here it is seen that this extended regularization improves the result, implying that extended regularization is a fruitful path.

11.5 Discussion

A new approach for PDE based surface estimation has been presented for use in the usual structure from motion framework. This approach uses the estimated 3D point structure to initialize the optimization, whereby a significant speed up and resistance to poor data is achieved. Both these issues are vital when performing structure and motion on real data.

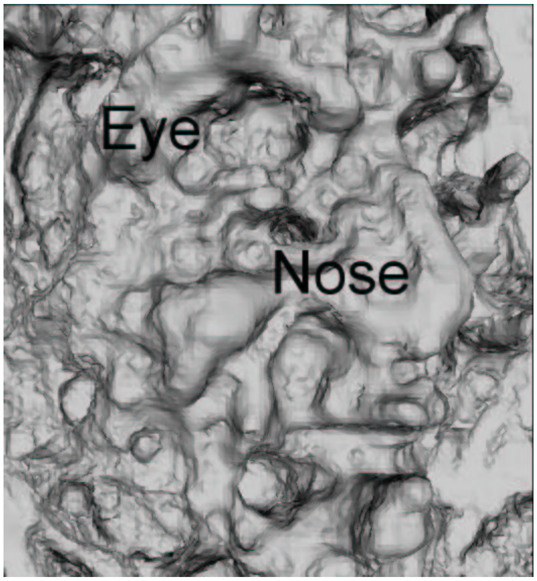
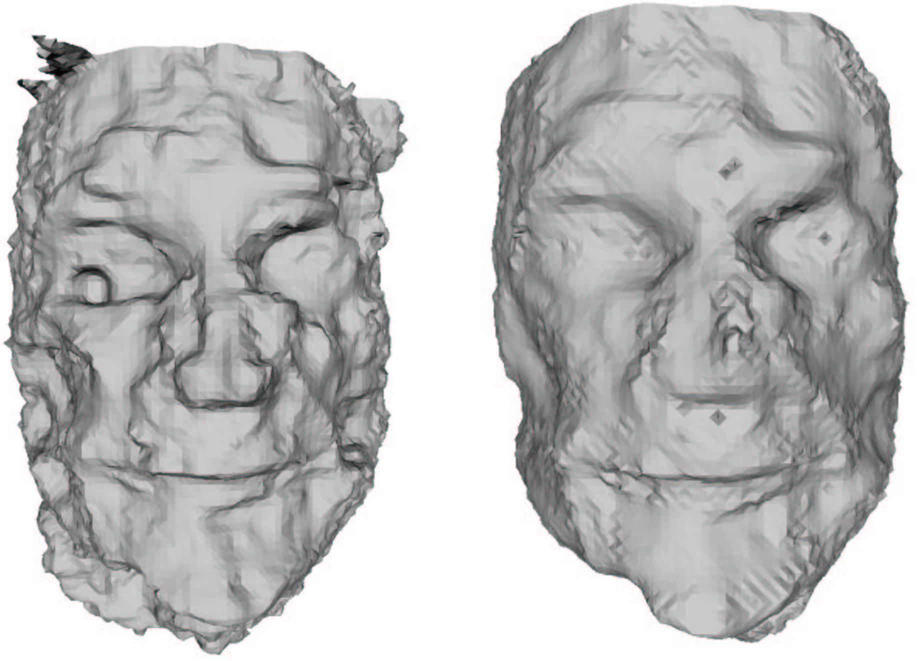


Figure 11.4: An intermediate iteration in face sequence, when using an inward force, $\mathcal{G}$. It is seen how the surface has 'gone through' the true surface, $\mathcal{S}^*$. In later iterations the holes will get bigger, and eventually the smoothness constraint of (11.3) will pull the surface from $\mathcal{S}^*$, whereupon it will collapse under its own curvature.

Acknowledgments

The authors owe great thanks to Henrik Malm, Fredrik Kahl and Björn Johanson delivering the data.



(a) (b)

Figure 11.5: (a) Proposed method on the face data set, see Figure 11.3(a). (b) Proposed method on the face data set, see Figure 11.3(a). Here the extended regularization of (11.5) was used.

CHAPTER 12

Pseudo–Normals for Signed Distance Computation

by:Henrik Aanæs and J. Andreas Bærentzen

Often times reviews can seem harsh to the author. So it was nice to get this comment in regards to this work.

> *This is a difficult problem, one that I personally have looked at before, and abandoned without a suitable solution. This appears to be that solution.*

Anonymous Reviewer 3.

Abstract

The face normals of triangular meshes have long been used to determine whether a point is in- or outside of a given mesh. However, since normals are a differential entity they are not defined at the vertices and edges of a mesh. The latter causes problems in general algorithms for determining the relation of a point to a mesh.

At the vertices and edges of a triangle mesh, the surface is not C^1 continuous. Hence, the normal is undefined at these loci. Thürmer and Wüthrich proposed the angle weighted pseudo–normal *as a way to deal with this problem. In this paper, we undertake showing that the angle weighted pseudo–normal has an important property, namely that it allows us to discriminate between points that are inside and points that are outside the mesh.*

This result is used for proposing a simple and efficient algorithm for computing the signed distance field from a mesh. Moreover, our result is an additional argument for the angle weighted pseudo–normals being the natural extension of the face normals.

12.1 Introduction

When relating 3D triangular mesh structures to other geometric entities, it is often necessary to know how far a given point is from the mesh. Wether the given point is in- or out side of the mesh can also have significant importance, e.g. in collision detection this determines whether the object is hit or not. This entity is efficiently represented as a real number where the absolute value denotes the distance and the sign whether the point is outside or not – negative denoting inside – yielding a signed distance.

Another important use of how a point relates to a mesh is in generating signed distance fields, i.e. a discrete field where each entry contains the signed distance. This implicit representation of a surface is among others used with the level–set method proposed by Osher and Sethian [146]. So if you, e.g. have a surface represented as a mesh and you want to apply a method implemented via the level set framework, such a conversion is needed. It is this signed distance field computation that will be the applicational focus of this paper.

For closed, orientable, and smooth surfaces we can use the normals to find out if a given point $\mathbf{p}$ is inside or outside. Say we find a point on the surface $\mathbf{c}$ so that no other point is closer to $\mathbf{p}$. Then, we know that the normal at $\mathbf{c}$ will point either directly away from or directly towards $\mathbf{p}$ depending on whether we are inside or outside the surface. Hence, the inner product between $\mathbf{r} = \mathbf{p} - \mathbf{c}$ and the normal will tell us whether $\mathbf{p}$ is inside or outside. When the surface is a triangle mesh, the situation is somewhat different because a mesh is not smooth everywhere: At edges and vertices the normal is not defined. To overcome this problem, we need to define a pseudo–normal for vertices and edges with the property that the sign of the inner product determines whether $\mathbf{p}$ is inside or outside. It turns out, that the required pseudo–normal is the *angle weighted pseudo–normal* proposed by Thürmer and Wüthrich [201] and independently by Sequin [190]. This normal is computed for a vertex as the weighted sum of all normals to faces that contain the vertex. The weight is the angle between those two edges of the face that contain the vertex, hence the name.

Our main contribution is proving that the angle weighted normal fullfills our requirement, i.e. that the sign of the inner product between a vector from a closest mesh point $\mathbf{c}$ to the corresponding point $\mathbf{p}$ and the angle weighted normal does indeed discriminate between inside and outside. A number of other pseudo–normals have previously been proposed. For most of these it is easy to show that they do not have this property.

This theoretical result is relevant for any application where it is required to find out whether a point is inside or outside a triangle mesh. The application that motivated our own work was the computation of discrete 3D distance fields. Another obvious application is collision detection or path planning. However, our result also strengthens the argument that the angle weighted pseudo–normal is a natural choice whenever it is necessary to define a normal at the vertex of a triangle mesh and no analytical normal is available. Another argument due to Thürmer et al. [201] is the fact that the angle weighted pseudo-normal depends only on geometry and is invariant to tesselation.

The paper is organized as follows; first the angle weighted pseudo–normal is introduced in Section 12.2 and compared to other pseudo–normals proposed in the literature. The central proof is presented in Section 12.3 followed by a description of how this result can be used for computing signed distance fields in Section 12.4.

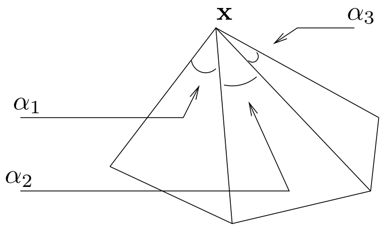


Figure 12.1: The incident angels $\{\alpha_1, \alpha_2, \alpha_3, \dots\}$ of point $\mathbf{x} \in M$.

12.2 Angle Weighted Pseudo–Normal

First some formalism. Let M denote a triangle mesh. Assume that M describes a closed, orientable 2–manifold in 3D Euclidian space, i.e. the problem is well defined. Denote by $\mathcal{M}$ the closure of the interior of M, i.e. $M = \partial\mathcal{M}$. Define the *unsigned* distance from a point $\mathbf{p}$ to M as[1]

$$d(\mathbf{p}, M) = \inf_{\mathbf{x} \in M} \|\mathbf{p} - \mathbf{x}\| \ , \tag{12.1}$$

and let the sign be determined by whether $\mathbf{p}$ is in $\mathcal{M}$ – positive denoting $\mathbf{p} \notin \mathcal{M}$. The optimum or optima in (12.1) are the *closest point(s)* to $\mathbf{p}$. We use $\mathbf{c}$ to denote a closest point.

For a given face, define the normal[2] as $\mathbf{n}$, which is assumed pointing outward. I.e. if the closest point, $\mathbf{c} \in M$, to $\mathbf{p}$ is on a face, then the sign is given by

$$sign\,(\mathbf{r} \cdot \mathbf{n})\,, \quad \mathbf{r} = \mathbf{p} - \mathbf{c} \ . \tag{12.2}$$

The angle weighted pseudo–normal for a given point, $\mathbf{x} \in M$ is then defined as

$$\mathbf{n}_\alpha = \frac{\sum_i \alpha_i \mathbf{n}_i}{||\sum_i \alpha_i \mathbf{n}_i||} \ , \tag{12.3}$$

where i runs over the faces incident with $\mathbf{x}$ and α_i is the incident angle, c.f. Figure 12.1.

Even though Thürmer and Wüthrich [201] only considered this angle weighted pseudo–normal for vertices of the mesh, it generalizes nicely to faces and edges. In the face case there is *one* incident face, namely the face itself, hence

$$\mathbf{n}_j = \mathbf{n}_\alpha = \frac{2\pi\mathbf{n}_j}{||2\pi\mathbf{n}_j||} \ ,$$

since the length of the normal in unit. This illustrates, that the angle weighted pseudo–normal can be seen as a generalization of the standard face normal. In the edge case there are two incident faces, both with weight π, giving the standard average of the two associated normals.

[1] Infimum is the greatest lower bound of a set, denoted inf.

[2] It is assumed, that the normals have length 1.

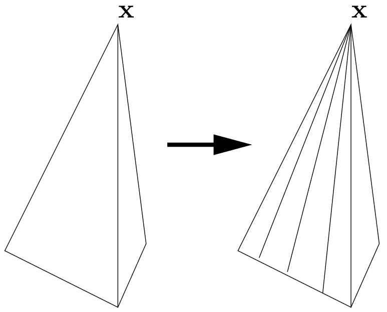


Figure 12.2: By subdividing one of the incident faces of $\mathbf{x}$ enough the unweighted mean normal can come arbitrarily close to the normal of that face. Since just using the normal of an arbitrary incident face clearly does not work, this approach is inapplicable for sign computation in general.

12.2.1 Related Work

Other pseudo–normals have been proposed as extensions to the face normal. However, none of the ones we know of can take the role of the face normal in (12.2) and used for sign computation, as it will be shown below that the angle weighted pseudo–normal can.

The most obvious pseudo–normal, mentioned in [80], is the unweighed mean of normals, i.e.

$$\frac{\sum_i \mathbf{n}_i}{||\sum_i \mathbf{n}_i||} \quad .$$

Glassner proposed a slightly different method [77]. For a given vertex, $\mathbf{v}$, we find all neighbouring vertices and the plane that best fits this set of points using the method of least squares. Finally, the plane normal is used as the normal at $\mathbf{v}$. The method can be modified slightly. Instead of using the neighbouring vertices, we intersect all edges incident on $\mathbf{v}$ with an infinitesimal ball. Now, the points where the edges and the ball intersect are used as the basis for the least squares fit.

It is clear that none of these pseudo–normals are invariant to triangulation. In other words, we can change the normal by changing the mesh but without changing the geometry. This is especially easy to see for the unweighted average normal as illustrated in Figure 12.2. The problem that arises when least squares fitting to neighbouring vertices is illustrated in Figure 12.3.

Another pseudo–normal is proposed by Huang et al. [99], uses the incident face normal with the largest inner product, i.e.

$$\mathbf{n}_m, \quad m = \text{argmax}_j ||\mathbf{r} \cdot \mathbf{n}_j|| \quad .$$

However, this does not always work. A counter example is given in Figure 12.4.

Nelson Max [131] proposes a pseudo normal (for vertices only) consisting of the cross products of all pairs of incident edges (associated with the same face) divided by the squared

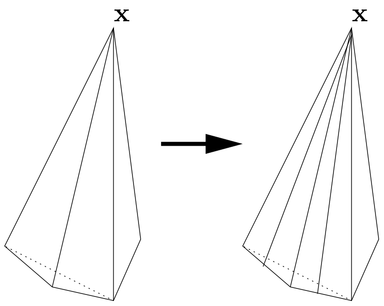


Figure 12.3: A counter example for Glassners proposed pseudo–normals. By shortening or extending edges, we can ensure that the points line in an almost arbitrary plane. Thus we can control the normal leading to the same situation as the one illustrated in Figure 12.2.

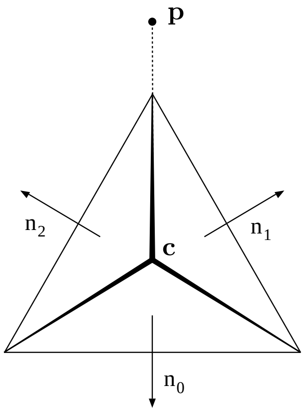


Figure 12.4: A counter example for Huang's normal. In the figure, a pyramid is seen from above. Let $\mathbf{r} = \mathbf{p} - \mathbf{c}$ where $\mathbf{c}$ is the apex and $\mathbf{p}$ is a point above the pyramid. It is clear that the inner product $\mathbf{r} \cdot \mathbf{n}_0$ is greater than both $\mathbf{r} \cdot \mathbf{n}_1$ and $\mathbf{r} \cdot \mathbf{n}_2$. Unfortunately, since $\mathbf{n}_0$ points away from $\mathbf{p}$, the point $\mathbf{p}$ is incorrectly classified as being inside the pyramid.

lengths of these edges, i.e. let $\mathbf{e}_i$ denote an incident edge, then

$$\mathbf{n}_{\times} = \sum_i \frac{\mathbf{e}_i \times \mathbf{e}_{i+1}}{||\mathbf{e}_i||^2 + ||\mathbf{e}_{i+1}||^2} \ .$$

It is demonstrated that this pseudo–normal produces results that are very close to the analytic normals for a certain class of smooth surfaces. However, this normal is not suited for sign computation. To see this, consider what happens when a surface is retesslated to make an edge extremely short. In this case, the normals of the two faces that share the edge will completely dominate the sum. See Figure 12.5.

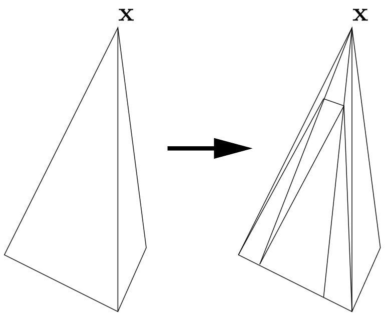


Figure 12.5: By a contrived retriangulation it is possible to change the normal so that it is arbitrarily close to the normal of one of the faces. Thus, we arrive at the same problem as the one illustrated in Figure 12.2.

12.3 Proof

Here it will be proven, that the angle weighted pseudo–normal can take the role of the face normal in (12.2), thus generalizing to all points $\mathbf{x}$ on the mesh, M. Since we are only interested in the sign, we omit the normalization and only consider

$$\mathbf{N}_\alpha = \sum_i \alpha_i \mathbf{n}_i \ , \tag{12.4}$$

easing notation. Also $\mathbf{N}_\alpha$ is more computationally efficient than $\mathbf{n}_\alpha$.

THEOREM 12.1 *Let there be given a point $\mathbf{p}$, and assume that $\mathbf{c}$ is a closest point in M so that*

$$\|\mathbf{c} - \mathbf{p}\| = d \ , \quad d = \inf_{\mathbf{x} \in M} \|\mathbf{p} - \mathbf{x}\| \ .$$

Let $\mathbf{N}_\alpha$ be the sum of normals to faces incident on $\mathbf{c}$, weighted by the angle of the incident face, i.e.

$$\mathbf{N}_\alpha = \Sigma \mathbf{n}_i \alpha_i \ . \tag{12.5}$$

Finally, consider the vector $\mathbf{r} = \mathbf{p} - \mathbf{c}$*. It now holds for*

$$D = \mathbf{N}_\alpha \cdot \mathbf{r} \ , \tag{12.6}$$

that $D > 0$ *if* $\mathbf{p}$ *is outside the mesh.* $D < 0$ *if* $\mathbf{p}$ *is inside.*

To prove this, we first consider the case where $\mathbf{p}$ is outside the mesh.

Define the volume S as the intersection of $\mathcal{M}$ and a ball, B, centered at $\mathbf{c}$. The radius of B is chosen arbitrarily to be 1. However, B may not contain any part of the mesh not incident on $\mathbf{c}$. If that is the case, we can fix the problem by rescaling the mesh. ∂S (the boundary of S) consists of a part coincident with the mesh, ∂S_M, and a part coincident with the ball, $\partial S_B = \partial S - \partial S_M$. Observe that $\partial S = \partial S_M \cup \partial S_B$ and $\partial S_M \cap \partial S_B = \emptyset$.

Introduce a divergence free velocity field, F, where at any point $\mathbf{q}$, $F(\mathbf{q}) = \mathbf{r}$. Then, from the theorem of Gauss we have (F being divergence free)

$$\begin{aligned} \int_{\partial S} F \cdot \mathbf{n}(\tau) d\tau &= 0 \\ = \int_{\partial S_M} \mathbf{r} \cdot \mathbf{n}(\tau) d\tau &+ \int_{\partial S_B} \mathbf{r} \cdot \mathbf{n}(\tau) d\tau \ . \end{aligned} \tag{12.7}$$

LEMMA 12.2 *For any point* $\mathbf{q} \in S$ *the angle* $\angle(\mathbf{cq}, \mathbf{cp})$ *is greater than or equal to* $\pi/2$*, when* $\mathbf{p} \notin \mathcal{M}$*.*

Proof: By construction $\mathbf{c}$ is a star point in S, i.e. the line segment between c and any point in S lies completely in S. Hence, if there is a $\mathbf{q}$ such that $\angle(\mathbf{cq}, \mathbf{cp}) < \pi/2$, there would be a point on the line between $\mathbf{c}$ and $\mathbf{q}$ which is closer to $\mathbf{p}$ then $\mathbf{c}$. This is easily seen, because if

$$\angle(\mathbf{cq}, \mathbf{cp}) < \pi/2 \ ,$$

the line segment from $\mathbf{c}$ to $\mathbf{q}$ must pass through the interior of a closed ball of radius r centered at $\mathbf{p}$, and any point in the interior of this ball will be closer to $\mathbf{p}$ than $\mathbf{c}$. Finally, since $S \subset \mathcal{M}$, this contradicts our requirement that $\mathbf{c}$ is the point in M closest to $\mathbf{p}$. See Figure 12.6. □

For all points $\mathbf{q} \in \partial S_B$ it is seen that the normal, $\mathbf{n}(\mathbf{q})$, is given by $\mathbf{cq}$, since B is the unit sphere centered at $\mathbf{c}$. So, by Lemma 12.2[3], $\mathbf{n}(\mathbf{q}) \cdot \mathbf{r} \leq 0$ for all $\mathbf{n}_B$. Therefore, we have that

$$\int_{\partial S_B} \mathbf{r} \cdot \mathbf{n}(\tau) d\tau < 0 \ . \tag{12.8}$$

The inequality in (12.8) is strict because the left hand side is only zero if the area of ∂S_B is zero, and this, in turn, would require the mesh to collapse breaking our manifold assumption.

From (12.7) and (12.8) it now follows that

$$\int_{\partial S_M} \mathbf{r} \cdot \mathbf{n}(\tau) d\tau > 0 \ . \tag{12.9}$$

[3] Observe that $\partial S_B \subset S$

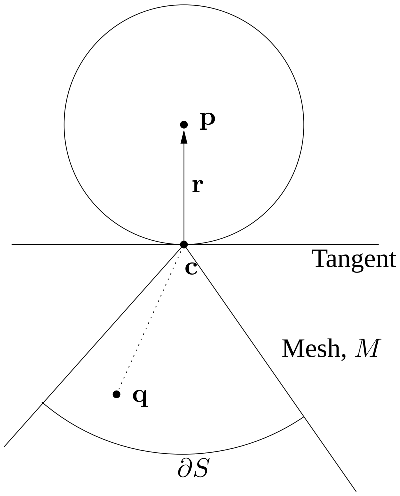


Figure 12.6: Illustration of Lemma 12.2. It is seen that $\angle(\mathbf{cq}, \mathbf{cp}) \geq \pi/2$, since $\mathbf{c}$ is the point in M closest to $\mathbf{p}$. Note that $\mathbf{c}$ is not constrained to be a vertex.

It is seen that the intersection of face i and S has an area[4] equal to α_i, implying that the flux of F through this intersection is given by $\mathbf{r} \cdot \mathbf{n}_i \alpha_i$. So

$$\int_{\partial S_M} \mathbf{r} \cdot \mathbf{n}(\tau) d\tau = \Sigma \mathbf{r} \cdot \mathbf{n}_i \alpha_i = \mathbf{r} \cdot \mathbf{N}_\alpha = D > 0 \ . \tag{12.10}$$

Proving the theorem for $\mathbf{p}$ outside the mesh. If $\mathbf{p}$ is inside the mesh, the situation is essentially the same, except for the fact that the direction of the involved normals point the other way. This means that the integral over ∂S_B changes sign. Thus, D becomes negative which concludes our proof □

Note that we do not assume that the closest point is unique. The proof requires only that $\mathbf{c}$ is *a* closest point. This means that Theorem 12.1 also holds in the case where $\mathbf{p}$ lies on the medial axis.

12.4 Signed Distance Field Algorithm

As mentioned, our original motivation for this work was to simplify the generation of discrete signed distance fields from triangle meshes. Basically, a distance field is a 3D grid of voxels where each voxel contains the signed shortest distance to the mesh. Signed distance fields have a number of applications. For instance, signed distance fields are usually the starting point for the Level-Set Method. Before we discuss our method, we briefly review the literature on distance field generation.

[4] Note that the area of a wedge cut out of the unit disc is equal to the angle of that wedge.

Early work on the distance field computation, signed as well as unsigned, is presented by Payne and Toga in [150]. In order to compute the (unsigned) distance at a voxel, one must compute for each voxel the distance to all triangles and compare to find the shortest. Several optimizations are possible; in particular it is a good idea to use a tree structure for bounding boxes of triangles. this structure can be used to cut down on the required number of distance computations.

Several improvements have been proposed for this work. Jones [107] proposed a method where the search space of each element in the distance field is reduced, especially if only a narrow band distance field is required, c.f. [10]. Yet an optimization aim at hardware implementation is presented in [49], and a hierarchical decomposition of the problem is the aim of [82].

As for computing the sign of the distance field, it has usually been treated as a totally disjoint task from computing the unsigned distance field. Payne and Toga [150] propose converting the mesh to determine what is inside and what is outside.

Mauch [130] computes sign and distance at the same time but his method is a bit more complicated than the classic approach: First one divides 3D space into (truncated) Voronoi regions corresponding to each face, edge, and vertex of the mesh. These Voronoi regions are represented as polyhedra that, in turn, are scan converted. The regions corresponding to edges and faces will be either interior or exterior to the mesh depending on whether the mesh is locally concave or convex.

In the above algorithms for unsigned distance computing, note that for each point in the discrete grid, $\mathbf{p}$, the closest point on the mesh $\mathbf{c}$ and the $\mathbf{r} = \mathbf{p} - \mathbf{c}$ are computed. The latter is needed in order to compute the distance. We propose using this fact with the result of Section 12.3, to form an integrated and simple method for computing the *signed* distance field.

Specifically, it is proposed augmenting the mesh structure with the angle weighted pseudo–normals for each face, edge and vertex, either by precalculation or via a function call. Then it is straight forward to extend the algorithms for computing *unsigned* distance fields to the signed case. Once the $\mathbf{c}$ and $\mathbf{r}$ are found for a given $\mathbf{p}$, the associated distance should be:

$$d = ||r|| sign\left(\mathbf{r} \cdot \mathbf{N}_\alpha\right) \quad ,$$

instead of

$$d = ||r|| \quad .$$

Here $\mathbf{N}_\alpha$ is the angle weighted pseudo–normal associated with $\mathbf{c}$. For further details the interested reader is referred to [26].

Apart from being a simple extension to the unsigned distance field algorithms, it is also seen that the proposed extension does not jeopardize the efficiency, in that all the angle weighted pseudo–normals can be computed in linear time. It can also be mentioned, that we have applied the proposed approach, with success, in a multiple view stereo approach, c.f. [2].

A single example of an application of our method is shown in Figure 12.7. The figure shows a visualization of a distance field created using our method.

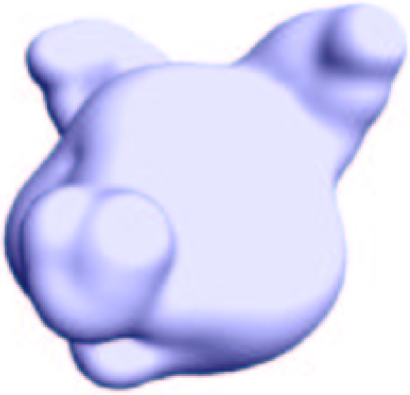


Figure 12.7: This image shows the level 2.2 offset surface of a triangle mesh. The mesh was converted to a distance field using our method, and the offset surface is simply the level 2.2 isosurface of the distance field which was rendered using texture based volume rendering.

12.5 Discussion

We have proven that the angle weighted pseudo–normal proposed by Thürmer and Wüthrich [201] and Sequin [190], have the property that they can be used for determining whether a given point is inside or outside of a given mesh. It is also demonstrated that a wealth of other pseudo–normals do *not* posess this property. This result has general relevance beyond signed distance field computation, in that it strengthens the use of angle weighted pseudo–normal for generalization of face normals.

This result is applied to a simple and integrated algorithm for computing signed distance fields, by a slight extension of the algorithms for computing *unsigned* distance fields.

CHAPTER 13

PDE Based Shape from Specularities

by: Jan Erik Solem, Henrik Aanæs and Anders Heyden

Abstract

When reconstructing surfaces from image data, reflections on specular surfaces are usually viewed as a nuisance that should be avoided. In this paper a different view is taken. Noting that such reflections contain information about the surface, this information could and should be used when estimating the shape of the surface. Specifically, assuming that the position of the light source and the cameras (i.e. the motion) are known, the reflection from a specular surface in a given image constrain the surface normal with respect to the corresponding camera.

Here the constraints on the normals, given by the reflections, are used to formulate a partial differential equation (PDE) for the surface. A smoothness term is added to this PDE and it is solved using a level set framework, thus giving a "shape from specularity" approach. The structure of the PDE also allows other properties to be included, e.g. the constraints from PDE based stereo.

The proposed PDE does not fit naturally into a level set framework. To address this issue it is proposed to couple a force field to the level set grid. To demonstrate the viability of the proposed method it has been applied successfully to synthetic data.

Keywords: Shape, Level Sets, Specularities, BRDF, Surface Estimation, Structure from Motion.

13.1 Introduction

Structure and motion, the reconstruction of an object and the motion of the camera from a sequence of images, is one of the most widely studied fields in computer vision. The final stage of a structure and motion system is estimating the shape of the object based on estimates of cameras and a sparse point cloud representing the structure. One assumption that is frequently made is that the object one tries to model is Lambertian, i.e. that light is reflected equally in all directions and therefore features have constant brightness from all viewpoints. Several methods for surface estimation have been proposed that uses the Lambertian surface assumption [66, 106, 137, 145, 157, 225].

Non-Lambertian features are usually considered as outliers and removed. This means that the parts of the resulting 3D-model corresponding to specular regions will usually be poor. The method proposed in this paper addresses this problem by estimating surfaces using information from the specular reflections.

The information contained in the specular reflections is that for a given camera position and light source the surface normal is constrained. This information can then be used to estimate the shape of the surface in specular regions to obtain a better 3D-model.

Consider the scenario of creating a 3D-model of a car. Cars contain smooth specular surfaces and are very hard to model with standard structure from motion techniques. Typically, the only features that can be extracted are sharp corners of the body or at edges of doors or windows. This is not enough to make a satisfactory model. Using the technique proposed in this paper it could be possible to estimate the surface where specular reflections are observed. In fact it is should even be possible to reconstruct the shape of semi-transparent specular surfaces such as windows, which is not possible with other methods, see Figure 13.1.

The problem setting can be summarized as follows. We wish to estimate a specular surface using the information contained in the specular reflections. This is done in order to complement the models obtained through structure from motion techniques. Our method is not a "stand alone" method since it requires the use of a structure from motion algorithm to determine the camera motion, which is necessary to determine surface normals. We see the shape from specularities scheme proposed in this paper as an integral part of a larger surface estimating scheme.

We use a level set representation of the surface which makes it easy to represent complex surfaces that can change topology as they evolve. We also derive constraints for the surface and in order to evolve the surface according to these constraints we introduce a coupled force field.

The paper is organized as follows: In Section 13.2 the preliminaries are explained and Section 13.3 describes the formalism. Section 13.4 shows how this is implemented and finally Section 13.5 shows experimental results.

13.1.1 Previous Work

The problem of recovering a surface from speculatities is related to the area of shape from shading [28, 221]. Shape from shading deals with reconstructing the shape of smooth Lambertian surfaces from gray level images.

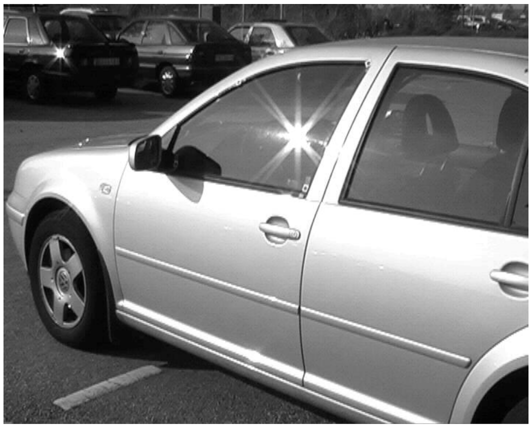


Figure 13.1: Estimating specular surfaces. The specular reflection on the car window gives information on the orientation of the surface normal. An image sequence showing the motion of the specular reflection can be used to estimate the shape of the window. It does not matter that the window is semi-transparent.

Some previous work has been done in the area of reconstructing or estimating surfaces from specularities. An early paper examining the information available from the motion of specularities under known camera motion is [227]. These methods all require some form of laboratory setup. Work has been done on recovering surfaces by illuminating them with circular light sources [224] or extended light sources [142]. Some work has also been done on reconstructing perfect mirror surfaces by studying reflections of a calibration object containing lines passing through certain points [169]. The method proposed in this paper is valid for general camera motion and general smooth specular surfaces as opposed to previous attempts where extended light sources, calibrated scenes, controlled camera motion or other constraints were used.

Our proposed method of adapting the level set framework is similar to the local operators proposed in [139]. But our work differs in that the local operators are different and are adapted to data fitting instead of 3D sculpturing.

Other methods have been proposed for fitting a surface to data using a variational approach, e.g. [35, 40, 116]

13.1.2 Contribution of Paper

The contribution of this paper is to propose a new method that makes it possible to estimate specular surfaces from image sequences taken with ordinary uncalibrated hand-held video cameras. Furthermore, the proposed method is valid for general camera motions. The only assumptions made are that the surface is smooth, i.e. that it has continuous derivatives, and that the light source is distant and point-shaped. The estimation of the surface is done using a level set approach where a force field is coupled to the level set grid.

13.2 Background

As a courtesy to the reader and to introduce notation a brief introduction of background material will be presented.

13.2.1 Camera Model

The following notations will be used: $\mathbf{X}$ denotes object points in homogeneous coordinates and $\mathbf{x}$ denotes image points in homogeneous coordinates. The focal point will be denoted $\mathbf{c}$ and the camera matrix P. We use the standard pin-hole camera model, cf. [89]. The object points are then related to the image points by a matrix multiplication,

$$\mathbf{x} \sim P\mathbf{X} \ , \tag{13.1}$$

where $\sim$ denotes equality up to scale. Given the camera matrix P, the line of sight corresponding to the image point $\mathbf{x}$ is given by

$$\mathbf{r}(\mathbf{x}) = \mathbf{c} + \lambda P^{+}\mathbf{x} \ , \tag{13.2}$$

where $\mathbf{c} = \mathcal{N}(P)$ denote the focal point, λ the depth parameter and P^{+} denote the pseudo-inverse of P. Hence if a specularity is observed at point $\mathbf{x}$ in a given camera, (13.2) denotes the possible locations of the surface reflecting the light.

13.2.2 Structure and Motion Estimation

To determine the shape of a surface with the proposed method it is necessary to know the motion of the camera. This is obtained using structure from motion techniques. Throughout this paper we assume that there are enough features in the scene to determine the motion of the camera and the structure of a limited number of feature points. This is done by extracting and tracking feature points through the image sequence. The fundamental matrix, F, and the trifocal tensor, T are estimated and an initial affine reconstruction is obtained from the cheriality constraints, cf. [144]. Finally an initial Euclidean reconstruction is obtained [95]. The Euclidean structure and camera motion is then refined using bundle adjustment, cf. [89, 185, 211].

13.2.3 Level Set Methods

In this paper we want to evolve a surface to find an estimate for a smooth specular surface using the level set method [146, 147, 174]. The time dependent surface $S(t)$ is implicitly represented as a level set of a scalar-valued function $\phi(\mathbf{x},t)$ in $\mathbb{R}^3$ as

$$S(t) = \{\mathbf{x}(t) \,;\, \phi(\mathbf{x}(t),t) = k\} \ , \tag{13.3}$$

where the value of $k \in \mathbb{R}$ is usually taken to be 0, making the surface a zero set of $\phi(\mathbf{x},t)$. One of the advantages of this representation is that the topology of the surface is allowed to change as the surface evolves, thus making it easy to represent complex surfaces that can merge or split and also surfaces that contain holes.

Differentiating the expression $\phi(\mathbf{x}(t),t) = k$ in (13.3) using the chain-rule gives

$$\phi_t + \nabla\phi(\mathbf{x}(t),t) \cdot \mathbf{x}'(t) = 0 \ . \tag{13.4}$$

This is the fundamental equation of motion for the level set. The normal of the level set surface is given by,

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \ . \tag{13.5}$$

The surface evolves under the influence of a speed function F in the direction of the normal. The function F can be assumed to be normal to the surface such that $F = \mathbf{x}'(t) \cdot \mathbf{n}$, since motion in other directions can be considered as a re-parametrization of the surface. The equation of evolution can then be written as

$$\phi_t + F|\nabla\phi| = 0 \ . \tag{13.6}$$

The surface is evolved by solving this PDE on a discrete grid. For a more thorough treatment of level set surfaces cf. [147, 174].

13.2.4 Specular Reflection

A non-Lambertian, specular, surface reflects light according to some distribution function, called Bi-directional Reflectance Distribution Function (BRDF). For a specular surface the BRDF is not uniform as in the Lambertian case. An example of a BRDF for a specular surface is shown in Figure 13.2a. The specular component can be seen when the surface normal **N** bisects the viewing direction **R** and the light direction **L** as shown in Figure 13.2b. A Lambertian surface would have a symmetric reflectance function without a specular lobe. The condition for specular reflection shown in Figure 13.2b is valid for the limiting case when the specular lobe has very small width and can be considered a delta-function. This would give a hard constraint on the viewing direction. However, if the specular lobe is not a delta-function then there is some uncertainty in the viewing direction. This gives a soft constraint on the direction for observing specularities. If the BRDF of the surface is known this information can be used. This is however, beyond the scope of this paper. To sum up, a specular reflection gives directional information and if the light source direction is known, the surface normal is also known.

13.3 Surface constraints from Specularites

The geometric conditions for specular reflection and the relation between a specularity in an image and the orientation of the surface normals leads us to formulate constraints that a surface has to satisfy in order to be consistent with the observation of specularities.

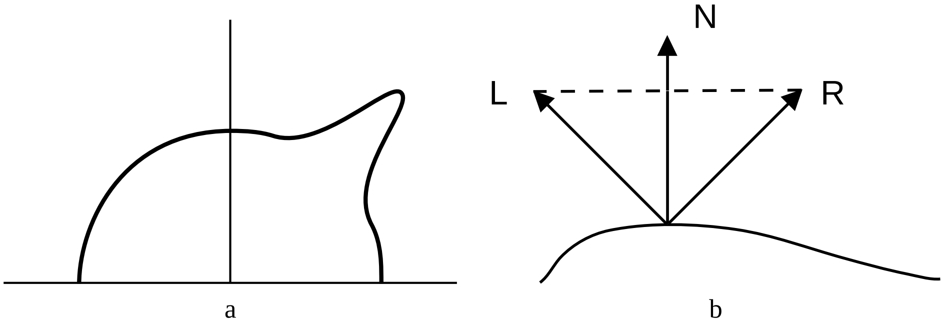


Figure 13.2: Specular reflection. a) An example of a BRDF. The specular lobe shows the increased reflected intensity for certain viewing directions. b) The condition for specular reflection

It is assumed, that there exist enough features in the scene to recover the camera motion and camera parameters, see Section 13.2.2. We also assume that the surface S is a smooth surface with observed specularities and that the light source is distant, point-shaped and its direction known. This means that the light source direction $\mathbf{L}$ is a constant vector. By smooth we mean that all components of S have continuous partial derivatives. These assumptions are reasonable since enough background can be included in the images by the person operating the camera and the distant light source assumption is valid for many scenarios, e.g. outdoor scenes with sunlight.

13.3.1 Specular Constrains

We use the following notation: S is a smooth surface in $\mathbb{R}^3$, $\mathbf{x}_i$ are the image coordinates for specularity i, c_i is the focal point of the corresponding camera and $\mathbf{r}_i$ is the ray from c_i through $\mathbf{x}_i$, see Figure 13.3. It is possible to have more than one specularity in each image so with image i we mean the image corresponding to specularity i. The total number of specularities in the sequence is denoted n.

The condition for observing a specular reflection is that the surface normal bisects the viewing direction and the incident light direction, see Figure 13.2. For a point on a surface S with normal $\mathbf{N}$ and light source direction $\mathbf{L}$, this relation is

$$\mathbf{R}+\mathbf{L} = (2\mathbf{N}\cdot\mathbf{L})\mathbf{N} \ , \tag{13.7}$$

and the specular reflection direction $\mathbf{R}$ can be determined as

$$\mathbf{R} = (2\mathbf{N}\cdot\mathbf{L})\mathbf{N}-\mathbf{L} \ . \tag{13.8}$$

Since we have computed the orientation and position of the camera for the whole sequence and the light source direction can easily be determined (e.g. by having one image where the shadow of the camera is visible) we get a series of constraints on the surface. The surface normal at the specular reflection fulfil the relation in (13.8) above. This means that at

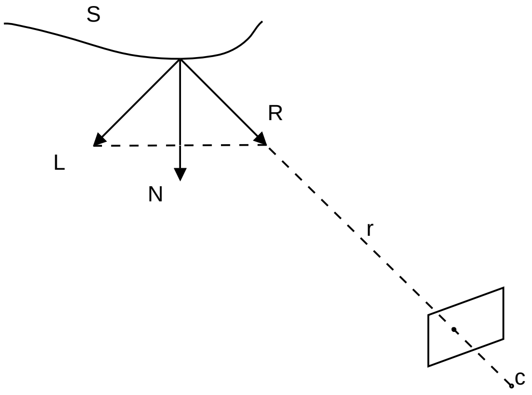


Figure 13.3: The relation between a specularity in an image and the surface normal.

the intersection of the ray $\mathbf{r}_i$, given by (13.2), from the focal point $\mathbf{c}_i$ through the specularity $\mathbf{x}_i$ in image i, and the surface, the normal $\mathbf{N}_i$ is known. This relation is shown in Figure 13.3. Solving for $\mathbf{N}_i$ we get

$$\mathbf{N}_i = \frac{\mathbf{L} - \tilde{\mathbf{r}}_i}{|\mathbf{L} - \tilde{\mathbf{r}}_i|} \quad , \tag{13.9}$$

where $\tilde{\mathbf{r}}_i$ is the directional vector for each ray, normalized so that $|\tilde{\mathbf{r}}_i| = 1$.

13.3.2 Regularization

The problem is that the depths λ_i from (13.2) cannot be determined. A distant light source means that the condition for specular reflection will be fulfilled at all points on the ray $\mathbf{r}_i$. Hence we get a whole family of surfaces that satisfy the normal constraints (13.9). There is then an inherent ambiguity in the solutions since there are many smooth surfaces at different depths λ_i that satisfy the conditions. Note also that the ordering of the rays $\mathbf{r}_i$ is depth-dependent. This is illustrated in Figure 13.4. To solve this ambiguity and to fix the surface in space we require that one or more features corresponding to 3D-points $\mathbf{X}_j \in \mathbb{R}^3$ can be found on the surface or the surface boundary.

Unfortunately, the constraints arising from known depths of a limited number of points on the surface boundary and the constraints on the normal direction to the surface arising from the detected reflections are not sufficient to uniquely determine the shape of the surface. Thus we have to add additional constraints. The most natural constraint to add is some kind of smoothness or regularity constraint on the surface.

To find a surface estimate we then have three different constraints, *point constraints* to position the surface in $\mathbb{R}^3$ and *normal constraints* to find the shape of the surface. These are due to local properties of the surface. A global *smoothness constraint* is also needed in order

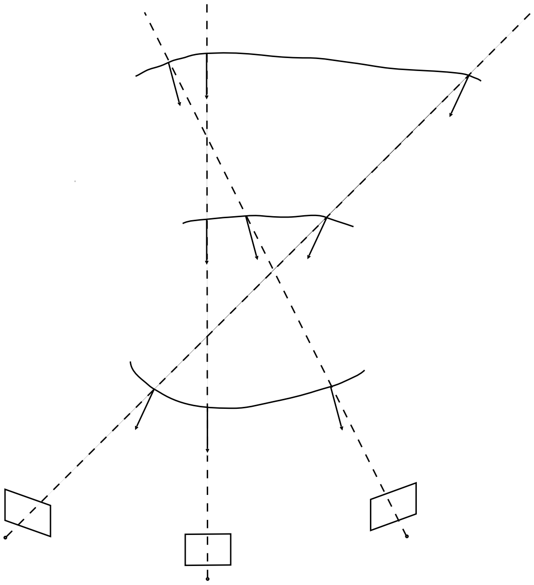


Figure 13.4: The surface ambiguity due to unknown depth (note that the ordering of the rays $\mathbf{r}_i$ is not constant).

to obtain a reasonable surface shape since there are many surfaces that fulfill the specular conditions even after the depth of one or more points are fixed.

The point constraints are obtained from the structure from motion estimation, where the structure is represented as a cloud of points $\mathbf{X}_j$. The constraints are then, that the surface S should pass through these points, or more formally

$$\forall\, \mathbf{X}_j \;\exists\, \mathbf{p}_j \in S \quad s.t.\; |X_j - \mathbf{p}_j| = 0 \;\; . \tag{13.10}$$

13.3.3 Induced Objective Function

We propose to apply the induced constraints, as expressed in (13.9) and (13.10), in the form of soft constraints as explained in Section 13.2.4. That is, instead of requiring that they hold exactly, an objective function will be used were deviations will be punished. As can be seen from the discussion above, such an objective function needs three terms corresponding to the normal constraints, the point constraints and a smoothing term.

We then obtain an expression for the objective function that looks like

$$\sum_i d_n(\mathbf{N}_i, \mathbf{n}_i) + \sum_j d_p(\mathbf{X}_j, S) + area \;\; , \tag{13.11}$$

where d_n and d_p are metrics for the deviation of the surface normal $\mathbf{n}_i$ from the desired normal $\mathbf{N}_i$ and for the point $\mathbf{X}_j$ from the surface. The last term is a mean curvature flow smoothness term [174]. Note that the additive structure of (13.11) makes it straight forward to incorporate other observed properties of the surface, e.g. those presented in [66].

Hence, the proposed scheme can be summarized as follows:

1. Compute camera motion from background features using structure from motion techniques, see Section 13.2.2.
2. Identify specularities in the images and determine the rays $\mathbf{r}_i$.
3. Calculate the light source direction $\mathbf{L}$ (from camera shadow or other method).
4. Determine the constraints for the surface normals $\mathbf{N}_i$ using (13.9) and the constraints for the points on the surface.
5. Find an estimate of the surface using (13.11) with the level set method as described in Section 13.4.

13.4 Level Set Implementation

The solution we are looking for is a smooth surface satisfying the constraints given in Section 13.3. We propose to do this by optimizing (13.11) using level sets. The third term is a standard mean curvature flow. We then propose to use a force field, derived from the normal constraints, when evolving the surface. The speed function F for the evolution equation then becomes

$$F = \alpha \mathcal{C} + \mathcal{F} \quad , \tag{13.12}$$

where α is a real-valued constant that determines the amount of smoothing imposed, $\mathcal{C}$ is the mean curvature flow term and $\mathcal{F}$ is a scalar valued force field incorporating the two first constraints.

13.4.1 Force Field Method

To minimize the objective function (13.11), the normal constraints and the point constraints will change the surface locally. To do this a force field $\mathcal{F}$ is derived as the sum of the local forces contributed by each constraint

$$\mathcal{F} = \sum_i f_n(\mathbf{N}_i) + \sum_j f_p(\mathbf{X}_j) \quad . \tag{13.13}$$

Here $f_n(\mathbf{N}_i)$ and $f_p(\mathbf{X}_j)$ are the force contributions for normal $\mathbf{N}_i$ and point $\mathbf{X}_j$. Since we are only interested in the force in the direction of the surface normal the direction of $\mathcal{F}$ is given. To obtain a scalar valued force field, we then only need to specify the amplitude.

A distance based function $\mathcal{D}$ is introduced to limit the volume affected by each constraint. We propose to use a symmetric three-dimensional gaussian with zero mean and width depending on the error of the corresponding constraint. The function $\mathcal{D}$ around point $\mathbf{p}$ is then

$$\mathcal{D}_p(\mathbf{d}) = \frac{1}{\sqrt{2\pi}\sigma}\, e^{-\mathbf{d}^T\mathbf{d}/2\sigma^2} \quad , \tag{13.14}$$

where $\mathbf{d}$ is the distance from $\mathbf{p}$ and σ is the standard deviation.

This force field method is related to the approach taken in [198] where methods for surface processing are developed by operating on the surface normals in a two step iterative algorithm.

13.4.2 Normal Constraints

Deviation of the surface normal $\mathbf{n}$ from the desired value $\mathbf{N}$ will result in an error ϵ_n according to

$$\epsilon_n = |\mathbf{r}| \quad , \tag{13.15}$$

where

$$\mathbf{r} = \mathbf{N} - \mathbf{n} \quad . \tag{13.16}$$

To change the surface in order to minimize the error we propose to use a local linear force field $\tilde{f}_n$ induced by each constraint as

$$\tilde{f}_n = (\mathbf{p} - \mathbf{X}) \cdot \mathbf{R} \quad , \tag{13.17}$$

Where $\mathbf{p}$ is the point at the intersection of $\mathbf{r}$, given by (13.2), and S, $\mathbf{X}$ is a point on the surface, $\cdot$ denotes the inner product and $\mathbf{R}$ is defined as

$$\mathbf{R} = \mathbf{r} - \frac{\mathbf{r}\cdot\mathbf{n}}{|\mathbf{n}|^2}\mathbf{n} \quad . \tag{13.18}$$

This field is shown in Figure 13.5. This field is then convolved with the volume limiting function $\mathcal{D}$ centered at the point $\mathbf{p}$ where the normal constraint is applied. This gives the final force as

$$f_n(\mathbf{N}_i) = \tilde{f}_n(\mathbf{N}_i) * \mathcal{D} \quad . \tag{13.19}$$

As the surface evolves, the value of λ in (13.2) needs to be updated since the intersection of $\mathbf{r}$ and S changes. This is done by finding the value that minimizes

$$\min_{\lambda} \; |\phi(\mathbf{c} + \lambda P^{+}\mathbf{x})| \quad . \tag{13.20}$$

for each ray $\mathbf{r}$, since the surface is defined as the zero set of ϕ.

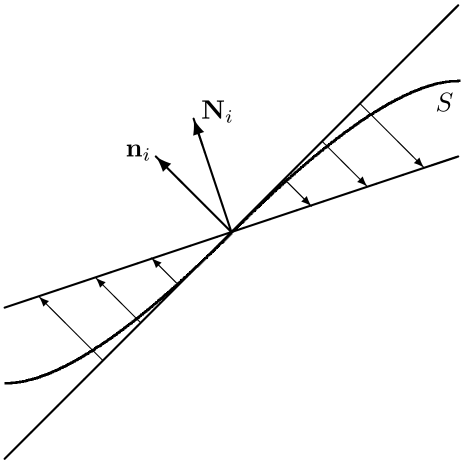


Figure 13.5: The local force field $\tilde{f}_n$ induced from a normal constraint.

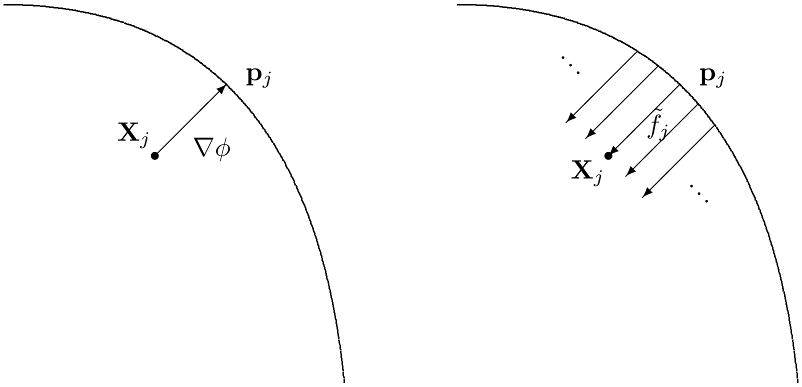


Figure 13.6: The local force field $\tilde{f}_p$ induced from a point constraint.

13.4.3 Point Constraints

The effect of the point constraints is similar to the normal constraints. The constraint of (13.10) in effect imply that the surface S should pass through $\mathbf{X}$. Hence the force needed to achieve this moves S onto $\mathbf{X}$. This field is shown in Figure 13.6. Due to the nature of the level set method, the force needed is:

$$\tilde{f}_p(\mathbf{X}_j) = -\phi(\mathbf{X}_j) \ , \tag{13.21}$$

in that this will make the zero set of ϕ go through $\mathbf{X}_j$. Enforcing locality gives

$$f_p(\mathbf{X}_j) = \tilde{f}_p(\mathbf{X}_j) * \mathcal{D} \ . \tag{13.22}$$

This approach is similar to the point attractors used in [139] to locally move a surface closer to specified points.

13.4.4 Range Adaptation

A given constraint only specifies the properties of a single point on the surface. However, since S is a smooth surface the constraints will influence a region around this point. Hence the force associated with a constraint needs to have local effect. The influence of a constraint is determined by the width of $\mathcal{D}_p(\mathbf{d})$. If the error in a constraint is large, a larger part of the surface should be affected by the resulting force and closer to the desired value the effects should be more local. This means that the standard deviation σ in (13.14) should be different across constraints and iterations.

For the normal constraints we set the width of $\mathcal{D}_p(\mathbf{d})$ so that the value at ϵ_n is 0.99. And for the point constraints we set the value at ϵ_p to be 0.9.

13.5 Experiments

To evaluate the proposed method, it was tested on a synthetic generated data set depicted in Figure 13.7, consisting of $\frac{1}{8}$ of a sphere. There are 3 point constraints, one in each corner and 31 normal constraints, as illustrated in Figure 13.8. The α value of (13.12) is 0.1.

The resulting surface is illustrated in Figure 13.9, where it can be seen, that the surface is reconstructed acceptably near the constraints, but not everywhere. The latter illustrates, that the surface chosen, i.e. the sphere, is not obtained by enforcing the prior of the curvature flow. For a more quantitative evaluation, the mean deviation of the resulting normals from the specified is shown in Table 13.1. The same data are shown for the point constraints. Table 13.1 shows that the proposed method for incorporating the information from specularities preforms well. This should be seen in the light, of numerical noise, and that the area minimizing smoothness constraint induces a force on the surface away from the specified constraints.

As mentioned, the proposed method is not primarily thought of as a "stand alone" method, but is applied as such here to test its possibilities. In this respect it is noted that the smoothness constraint in form of the curvature flow is not a particulary good regularizer for interpolation. So if surface estimation from sparse specular constraints were to be attempted, it

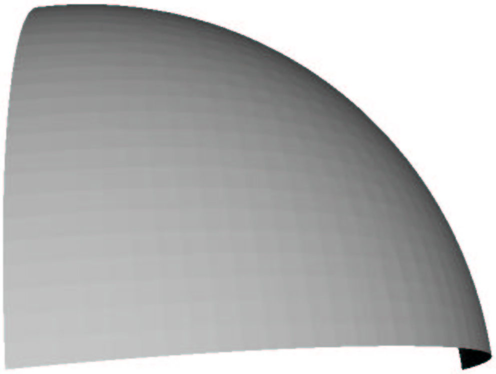


Figure 13.7: The original simulated data set.

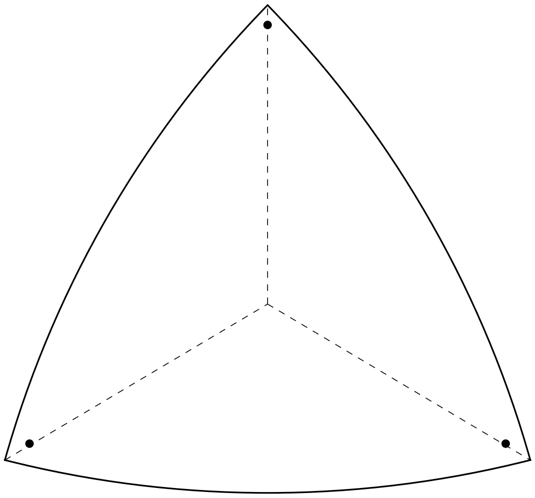


Figure 13.8: A schematic overview of the test set up. The points illustrate the 3 point constraints and the 31 normal constraints are distributed uniformly along the dashed lines.

Mean Deviation from:		
Normal Constraints	6.891	degrees
Point Contstraints	0.276	voxels

Table 13.1: Mean differences between realized and specified properties of the surface.

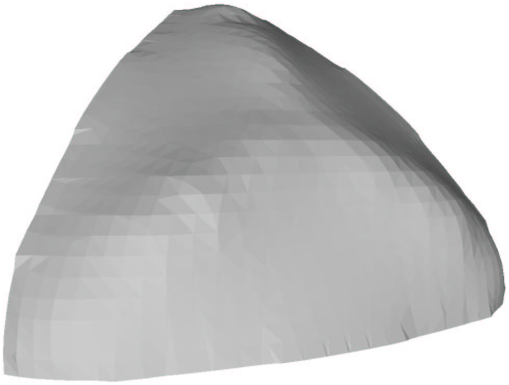


Figure 13.9: The reconstructed surface. Note that the reconstruction is acceptable along the constraints specified in Figure 13.8.

could be preferable to use another smoothness constraint. A likely candidate is a fourth order flow, which unfortunately, is hard to get numerically stable. Recently though, a feasible method has been proposed in [198].

13.6 Summary and Conclusions

We have proposed a method for reconstructing the shape of specular surfaces using a level set implementation to which a force field is coupled. This force field is derived from constraints on the surface that determines the shape of the surface, positions it is space and keeps the surface smooth. Experiments with synthetic data show very promising results.

The proposed method is not primarily intended to be used for surface estimation alone, and in future work it should be integrated with such methods, e.g. [66]. It could also be interesting to try other smoothing schemes, e.g. fourth order flow, if the proposed method should be used alone. Lastly, it could be interesting to formalize the proposed framework more, e.g. by formulating it as minimizing a functional.

Acknowledgement

We would like to thank J. Andreas Bærentzen, for interesting discussions and advice regarding the level set implementation.

CHAPTER 14

Deformable Structure from Motion

by: Henrik Aanæs and Fredrik Kahl

This is a working paper for our efforts on deformable structure from motion. Preliminary versions of this work have been published at [1, 6, 7]. At present we still have some unsolved problems particularly in regards to the nonlinear optimization in conjunction with the bundle adjustment. This induces problems with final validation of the proposed approach. Other approaches to model selection also need to be investigated, but requires the above mentioned issues to be resolved. These issues are discussed more in the following. Main advances have been made since the latest published version, i.e. [6], hence this is the version included here.

Abstract

In this paper the problem of estimating the scene structure and the motion of a non-rigid deformable object is analyzed. The object is supposed to deform according to a linear model while the motion of the camera relative the object can be arbitrary. No domain specific prior of the object is required. A complete algorithm is developed which consists of first creating an initial guess using a factorization algorithm, working on linearized data. Here upon an optimal solution is obtained through a non–linear optimization scheme, i.e. modified bundle adjustment. The complexity of the linear model is determined by model selection. The proposed approach assumes that an appropriate number of features have been tracked on the object.

With non-rigid objects, special issues concerning the well-posedness of the problem arises. There are a number of inherent ambiguities which do not occur in the traditional *setting of the structure and motion problem. These ambiguities have been identified and it is argued that they can be resolved using regularization. Lastly the effectiveness of the proposed method is investigated on both real and simulated data.*

14.1 Introduction

The estimation of structure and motion from image sequences is one of the most studied problems within computer vision. However, until now the literature, and hence the accompanying solutions, have mainly been focused on dealing with rigid objects [39, 89, 94, 204]. There have been some dealings with the estimation of structure and motion of multiple independently moving objects [46, 72, 112, 178, 202].

Yet, smoothly deforming structures are everywhere around us: trees sway, waves flutter and humans walk in a very non-rigid way. In fact, many interesting objects in our environment are non-rigid, e.g. humans, plants and animals. Hence there is a great need to deal with the estimation of their structure as well, since these are common 'objects' in our everyday lives.

In [133, 153] estimation of non-rigid motion is performed using a Kalman filter with physics based priors on the dynamics. However, the deforming objects need to be properly initialized. A lot of work on non-rigid motion estimation use domain specific priors. For example in [182, 186] human motion is estimated using a complete model of the body and its dynamics. Such priors are inconvenient as the problem of object identification is not a solved problem.

In this paper we present an approach for estimating the structure and motion of deforming or non-rigid objects. This is done by employing the Principal Component Analysis (PCA) framework [98, 151], whereby the structure model is a linear deformable model as described in e.g. [19, 45, 212]. These types of models have proven to be highly effective in expressing many types of deforming objects. Thus, the model is fairly general making it applicable in many different scenarios. The motion of the camera relative to the deforming object can be arbitrary. We assume that the cameras are calibrated. However, it is straightforward to generalize the approach to an uncalibrated setting.

The proposed approach assumes that the tracking problem has been solved, and hence assumes a number of tracked features on the object in question. This could e.g. be achieved via [24]. As is common practice with the standard rigid objects, the approach falls in two main steps. First an approximate solution is obtained using a factorization type algorithm, which assumes affine cameras. This approximate solution then forms a much needed initial guess to a non–linear optimization algorithm, which can be seen as a modified bundle adjustment algorithm.

However, the extension of structure from motion from rigid to non-rigid objects is non–trivial. At any given frame only a 2D projection of the 3D structure is given. With rigid objects this problem is circumvented by other projections of the same structure being present in other cameras. In the non-rigid case, this possibility is not present, because the structure is assumed to change between frames. This added complexity is approached here by modelling the variability of the structure in question. The complexity of the model is determined by model selection. The linear model of the object structure also introduces some additional ambiguities into the problem. These ambiguities are resolved by regularization.

The general non–rigid structure from motion problem was initially addressed by Brand et al. in [25] adapting the factorization approach of Tomasi and Kanade [204]. This work was later extended to more sophisticated non–rigid factorization approaches in [22, 209].

The main contribution of this paper is a general non-linear "gold standard"[89] approach to the non–rigid structure from motion problem. It is based on a method for estimating the variance of a 3D point cloud from projections. From this a principal component analysis (PCA) can be made. It is noted, that although the previous work on non–rigid structure from motion also deals with subspace selection, this is done directly on the tracked features instead of on the actual structure. Additionally, we identify the extra ambiguities that the relaxation of the rigidity constraint of the structure induces and propose how these should be dealt with. It is noted that in [22], a regularization prior is a also used to stabilize the solution, but the prior has no direct geometric or statistical meaning.

The organization of this paper is as follows: the problem is identified in Section 14.2 followed by a general scheme for its solution in Section 14.3. The issue of subspace selection is addressed in Sections 14.4, 14.5 and 14.6. Upon this the induced ambiguities are identified and interpreted in Section 14.7 leading to a description of the propped non–linear optimization method in Section 14.8. Lastly the proposed method is validated experimentally in Section 14.9.

14.2 Problem Definition

In this work, it is assumed that a set of tracked features are given, and that they are observed with the perspective camera model:

$$\lambda_{ij}\begin{bmatrix}x_{ij}\\y_{ij}\\1\end{bmatrix}=P_i\begin{bmatrix}S_{ij}\\1\end{bmatrix}=P_i\begin{bmatrix}X_{ij}\\Y_{ij}\\Z_{ij}\\1\end{bmatrix}\quad , \tag{14.1}$$

where x_{ij} and y_{ij} are the 2D projection of feature $j\in\{0\dots m\}$ in frame $i\in\{0\dots n\}$. Hence P_i is a 3×4 projection matrix representing the position and orientation of the camera. It is noted that the 3D features, S_{ij}, are also varying between frames, hence the extra subscript.

The scene structure or combined 3D features are denoted by $\mathbf{S}_i$, i.e.

$$\mathbf{S}_i=\left[\begin{array}{ccc}S_{i1} & \cdots & S_{in}\end{array}\right]\quad .$$

As mentioned, a model is required for the variation of the structure. Here the PCA framework is employed. It is assumed that the variations of the structure can be described by linearly varying modes, i.e.

$$\mathbf{S}_i=\mathbf{M}_\mu+\sum_{k=1}^{r}\beta_{ik}\mathbf{M}_k\quad , \tag{14.2}$$

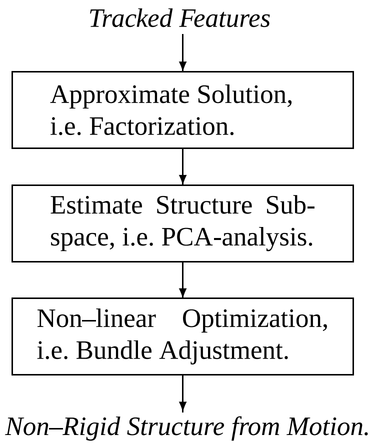

Tracked Features
Approximate Solution, i.e. Factorization.
Estimate Structure Subspace, i.e. PCA-analysis.
Non–linear Optimization, i.e. Bundle Adjustment.
Non–Rigid Structure from Motion.

Figure 14.1: Schematic overview of proposed scheme.

where β_{ik} is a scalar, $\mathbf{M}_k$ is a 3D mode of variation and $\mathbf{M}_\mu$ is the mean shape. This in essence means that the varying structure is handled by restricting it to a subspace of dimension $r+1$. Since it is assumed that $r << 3m$, $\Re^{3m}$ being the space of m 3D points, there should still be plenty of constrains to make an estimate.

The problem is then one of estimating the motion of the camera $\mathbf{P}_i$ and the structure $\mathbf{S}_i$, i.e. both the mean shape and its modes of variation, from the given image data.

14.3 Proposed Scheme

In order to solve the non-rigid structure from motion estimation problem, we propose a scheme heavily inspired by the scheme used for rigid structure from motion as illustrated in Figure 14.1. The main outline of this scheme is applying linearized versions of the camera model to get an approximate solution, which is then used as an initialization of a non–linear optimization or bundle adjustment to achieve a "gold standard solution"[89].

The main difference between the Rigid and Non–rigid approaches is the necessary estimation of the model complexity or order, r, and the accompanying modes, M_k. As mentioned above, this corresponds to restricting the structure to a subspace, which we here propose doing via principal component analysis (PCA).

PCA subspace selection is essentially performing an eigen–value decomposition of the variance of the given data ones data and then selecting the eigen–vectors corresponding to the largest eigen–values as the subspace. This implies the need for an estimate of the variance, which in this case is non–trivial. We propose doing this by taking a temporary structure from motion estimate, e.g. from the factorization algorithm, and disregarding the uncertainty of the camera estimates. Hereby the different images can be seen as given projections of the varying 3D structure, whereby the variance can be calculated, as seen in Sections 14.4 and 14.5.

14.3.1 Factorization

As for the approximate solution or factorization part of the scheme, there exists a number of methods that can be employed [22, 25, 209].

In this work, we used our own rather simple heuristic for obtaining an initial estimate of the structure from motion. According to (14.2), the mean of the 3D structure is M_μ. We use the assumption that the mean is in some sense dominant. More precisely, translated into the language of factorization, *if the factorization method of Tomasi-Kanade [204] is used, then the resulting structure is similar to M_μ* . This yields a simple algorithm for producing an estimate, and it has worked satisfactory for our experiments. However, it is unclear exactly when the assumption is not valid. This approach also has the advantage that the model order does not need to be known in advance. This is in contrast to the above mentioned factorization methods, i.e. [22, 25, 209], which assume that the model order is supplied by the feature tracking algorithm.

14.4 Mean and Variance from Projections

In this section the general problem of estimating mean and variance from projections will be considered. In the next section follows a discussion on how this relates to camera projections.

Assume that we have observed projections of a normal distributed vector, $\mathbf{x}_i$, such that there is a distinct projection to each observation (i.e. we have observed $\mathbf{y}_i$, $i \in \{1 \ldots n\}$). Then

$$\mathbf{y}_i = \mathbf{Q}_i \mathbf{x}_i \quad , \quad \mathbf{x}_i \in N(\mu, \mathbf{\Gamma}) \ , \tag{14.3}$$

where $\mathbf{Q}_i$ is a projection matrix. That is, $\mathbf{Q}_i$ can be written as $\mathbf{R}^T diag(1, \ldots, 1, 0, \ldots, 0)\mathbf{R}$ where $\mathbf{R}$ is a rotation matrix and $diag()$ denotes a diagonal matrix with the indicated elements. Hence it is noted that $\mathbf{Q}_i = \mathbf{Q}_i^T$ and $\mathbf{Q}_i = \mathbf{Q}_i \mathbf{Q}_i$.

Given the observations $\mathbf{y}_i$, we would like to find a suitable subspace for the $\mathbf{x}_i$ by using principal component analysis (PCA). In order to do this, μ and $\mathbf{\Gamma}$ have to be estimated, whereupon a straight forward factorization of $\hat{\mathbf{\Gamma}}$ will give the result. The approach used here for estimating μ and $\mathbf{\Gamma}$ is minimizing the the standard norms for mean and variance of a normal distribution modified for the projections.

At an intuitive level, it is seen that both the mean and variance are estimated as mean sums, and in this regard the $\mathbf{Q}_i$ can be seen as weights. Thus the derived solutions can be seen as weighted means.

14.4.1 Estimating the Mean

In order to estimate the mean or first order moment, $\hat{\mu}$, of $\mathbf{x}_i$ it is seen that it is the solution to the following minimization in the standard 2–norm modified by the $\mathbf{Q}_i$:

$$\min_{\mu} \sum_{i=1}^{n} (\mu - \mathbf{y}_i)^T \mathbf{Q}_i^T \mathbf{Q}_i (\mu - \mathbf{y}_i) = \min_{\mu} \sum_{i=1}^{n} \left(\mu^T \mathbf{Q}_i \mu - 2\mathbf{y}_i^T \mathbf{Q}_i \mu + \mathbf{y}_i^T \mathbf{Q}_i \mathbf{y}_i \right) \quad .$$

Differentiating and setting equal to 0 gives,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mu} \sum_{i=1}^{n} (\mu - \mathbf{y}_i)^T \mathbf{Q}_i (\mu - \mathbf{y}_i) = \sum_{i=1}^{n} (2\mathbf{Q}_i \mu - 2\mathbf{Q}_i \mathbf{y}_i) \Rightarrow \\ \hat{\mu} &= \left(\sum_{i=1}^{n} \mathbf{Q}_i \right)^{-1} \sum_{i=1}^{n} \mathbf{Q}_i \mathbf{y}_i \quad . \end{aligned} \tag{14.4}$$

This is seen to correspond to the weighted mean of the $\mathbf{y}_i$, which is what is expected.

14.4.2 Estimating the Variance

Let $\tilde{\mathbf{y}}_i$ denote the mean corrected observations (i.e. $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \hat{\mu}$). The norm associated with the variance of a multivariate normal distribution is $||\mathbf{A}|| = tr(\mathbf{A}\mathbf{A}^T)$ where $\mathbf{A}$ is a matrix and $tr(\dots)$ denotes the trace, cf. [59]. Hence the norm induced by the projections, and the accompanying minimization problem is:

$$\begin{aligned} &\min_{\mathbf{\Gamma}} \sum_{i=1}^{n} ||\mathbf{Q}_i \left(\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T - \mathbf{\Gamma} \right) \mathbf{Q}_i^T|| = \\ &\min_{\mathbf{\Gamma}} \sum_{i=1}^{n} tr\Big[\left(\mathbf{Q}_i \left(\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T - \mathbf{\Gamma} \right) \mathbf{Q}_i^T \right) \left(\mathbf{Q}_i \left(\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T - \mathbf{\Gamma} \right) \mathbf{Q}_i^T \right)^T \Big] = \\ &\min_{\mathbf{\Gamma}} \sum_{i=1}^{n} tr\Big[\mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i - \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i \mathbf{\Gamma}^T \mathbf{Q}_i - \mathbf{Q}_i \mathbf{\Gamma} \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i + \mathbf{Q}_i \mathbf{\Gamma} \mathbf{Q}_i \mathbf{\Gamma}^T \mathbf{Q}_i \Big] \quad . \end{aligned}$$

Differentiating and setting equal to 0 gives,

$$\begin{aligned} 0 &= \frac{\partial}{\partial \mathbf{\Gamma}} \sum_{i=1}^{n} ||\mathbf{Q}_i \left(\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T - \mathbf{\Gamma} \right) \mathbf{Q}_i^T|| = \sum_{i=1}^{n} -2\mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i + 2\mathbf{Q}_i \mathbf{\Gamma} \mathbf{Q}_i \Rightarrow \\ \sum_{i=1}^{n} \mathbf{Q}_i \mathbf{\Gamma} \mathbf{Q}_i &= \sum_{i=1}^{n} \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i \quad . \end{aligned} \tag{14.5}$$

Denoting the Kronecker product by $\otimes$ and the rearrangement of the elements of a matrix into a vector by $^{-}$, (14.5) can be modified to

$$\sum_{i=1}^{n} \mathbf{Q}_i \otimes \mathbf{Q}_i \bar{\mathbf{\Gamma}} = \left(\sum_{i=1}^{n} \mathbf{Q}_i \otimes \mathbf{Q}_i\right) \bar{\mathbf{\Gamma}} = \overline{\left(\sum_{i=1}^{n} \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i\right)} \Rightarrow \tag{14.6}$$
$$\bar{\hat{\mathbf{\Gamma}}} = \left(\sum_{i=1}^{n} \mathbf{Q}_i \otimes \mathbf{Q}_i\right)^{-1} \overline{\left(\sum_{i=1}^{n} \mathbf{Q}_i \tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_i^T \mathbf{Q}_i\right)} \ .$$

14.5 Camera Projections

In the case of images of 3D points, the imaging process can be seen as a projection of a given point onto the image plane. That is, if a point is observed by a camera – with known calibration – the 3D location of that point can be restricted to a line. This restriction can be described as a projection along this known viewing direction. That is, let Q_{ij}^x, Q_{ij}^y be an othonormal basis spanning the plane perpendicular to the viewing direction of point j in image i. Then the corresponding projection is given by:

$$Q_{ij} = Q_{ij}^x {Q_{ij}^x}^T + Q_{ij}^y {Q_{ij}^y}^T \ . \tag{14.7}$$

For a given frame, i, the 'observed' 3D points can be arranged into a vector $\bar{\mathbf{S}}_i$, i.e.

$$\bar{\mathbf{S}}_i = \begin{bmatrix} Q_{i1}S_{i1} \\ \vdots \\ Q_{im}S_{im} \end{bmatrix} \ , \tag{14.8}$$

Then it is seen that $\mathbf{Q}_i$ is a block diagonal matrix,

$$\mathbf{Q}_i = \begin{bmatrix} Q_{i1} & & \\ & \ddots & \\ & & Q_{im} \end{bmatrix} \ . \tag{14.9}$$

Thus the estimation of mean and variance is achieved by replacing $\mathbf{y}_i$ with $\bar{\mathbf{S}}_i$ in (14.4) and (14.6). Since the S_{ij} are not at hand, the $Q_{ij}S_{ij}$ are achieved by applying Q_{ij} to the 3D image coordinates on the image plane.

14.5.1 Computational Issues

In order to solve this estimation problem computationally efficient, the 3×3 block structure of the $\mathbf{Q}_i$ needs to be exploited. This is especially so when estimating the variance, where (14.6) implies inverting the $(3m)^2 \times (3m)^2$ matrix $\sum_{i=1}^{n} \mathbf{Q}_i \otimes \mathbf{Q}_i$. Denoting the 3×3 sub–matrices of $\mathbf{\Gamma}$ by γ_{gh}, it follows that a corresponding sub–matrix of $\mathbf{Q}_i \mathbf{\Gamma} \mathbf{Q}_i$ (cf. (14.5)) can be written as

$$Q_{ig}\gamma_{gh}Q_{ih} = Q_{ig} \otimes Q_{ih}\bar{\gamma}_{gh} \ . \tag{14.10}$$

Hence it isonly needed to compute and invert the[1] $\frac{m^2-m}{2}$ 9×9 matrices

$$\sum_{i=1}^{n} Q_{ig} \otimes Q_{ih} \quad , \tag{14.11}$$

instead of the $(3m)^2 \times (3m)^2$ matrix $\sum_{i=1}^{n} \mathbf{Q}_i \otimes \mathbf{Q}_i$. The latter would seriously limit the size of computationally feasible problems.

It should be noted, that if an orthographic camera is assumed (as it is done with factorization algorithms), all the Q_{ij} corresponding to a given frame are identical. This again implies that, in order to estimate the variance, cf. (14.6), only one 9×9 matrix needs to be computed and inverted.

14.5.2 Retrieving the Model Parameters

Once the model order, r, has been estimated via model selection, the model parameter of (14.2) has to be retrieved from the variance analysis. $\mathbf{M}_i$ can be achieved by a singular value decomposition (SVD) of $\hat{\mathbf{\Gamma}}$. Here each right singular vector corresponds to a mode rearranged into a vector, previously denoted by $\bar{\ }$. Hence the first r right singular vectors are applied, corresponding to the r most dominant modes.

Retrieving the β_{ik} is a non-linear optimization problem, but since the results are only needed as an initial guess for the bundle adjustment, we propose using a linear approximation. Denote the f^{th} row of the projection matrices P_i by P_{i_f}, then the x–coordinate in image i is given by

$$\frac{P_{i_1}\left(\mathbf{M}_\mu + \sum_{k=1}^{r} \beta_{ik}\mathbf{M}_k\right)}{P_{i_3}\left(\mathbf{M}_\mu + \sum_{k=1}^{r} \beta_{ik}\mathbf{M}_k\right)} \quad ,$$

which can be approximated by

$$\frac{P_{i_1}\left(\mathbf{M}_\mu + \sum_{k=1}^{r} \beta_{ik}\mathbf{M}_k\right)}{P_{i_3}\mathbf{M}_\mu} \quad . \tag{14.12}$$

The same is naturally done for the x–coordinate. Using this approximation the β can be estimated via linear least squares adjustment of (14.12) to the image coordinates – using both x and y coordinates. In some cases, $P_{i_1}\mathbf{M}_k \approx 0$ in which case regularization is needed. It is noted, that the approximation of (14.12) is commonly used in factorization algorithms.

14.6 Model Selection

When using the PCA framework for modelling the variation of structure, a vital point is to chose how many modes or principal components to include in the model. There exist numerous heuristic methods for doing this automatically, cf. [91]. But due to the nature of the problem, separating signal from noise, model selection is to a large extend an art, which should be kept in mind when applying the result. In our case, we want to be parsimonious

[1] $\mathbf{\Gamma}$ is symmetric so only the upper half of it has to be calculated.

with the number of modes, since if the structure is over parameterized, it might start modelling the camera motion by deforming the structure. As such, shrinkage inspired by ridge regression could be employed in the bundle adjustment to address this. Such shrinkage can be interpreted as contentious model selection, cf. [91].

We propose selecting the number of modes based on the variance estimate derived in Sections 14.4 and 14.5. It is seen that there are 3 contributions to the variance namely the image noise the varying structure and the approximation error of assuming an orthographic camera instead of a projective. The latter originates from the factorization approach. The model selection problem is then one of estimating the rank or dimensions of one part of the variance namely the varying structure based on the variance.

In [6] we successfully applied the Bayes information criteria (BIC) [172] as described in the PCA setting by Minka [134]. However an erroneous (14.6) was derived such that

$$\mathbf{Q}_i \otimes \mathbf{Q}_i \quad ,$$

became

$$\mathbf{Q}_i \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \mathbf{Q}_i \quad .$$

Effectively this gave a muffling of the variance estimate drowning contribution of the approximation error, giving good results. However, since this result is based on an error, it is uncertain if it will work in general, and as such we do not recommend using it. Using the correct version of (14.6) and the BIC in our experiments gives a wild over estimate of the number of modes, r, to a point where it becomes useless.

At present we use a generalized version of Horns method [97] on the variance, which gave reasonable results but still has a tendency to over-parameterize. As discussed in Section 14.8, we have not been able to fully validate the approach rigorously, and as such it is unclear if this approach will work satisfactory.

If it turns out that applying Horns method does *not* work satisfactory, we are planing to try methods which do not utilize the estimated variance of the structure directly. The reason being that the approximation error seems to have too much influence in practice. Specifically we are considering cross–validation like methods, cf. [91].

14.7 Ambiguities

The solution in rigid structure from motion is defined only up to an Euclidian similarity transformation. This ambiguity has practical implications in that regularization is needed in order to achieve decent convergence of the numerical optimization. It also has more theoretical implications, in that this reduces the observability, i.e. there are some properties of the system considered that can not be inferred. As such it is important to identify the extra ambiguities of the non–rigid case.

Apart from the ambiguity of an Euclidean similarity transformation, the extension of the modelling framework also induces additional ambiguities. We have been able to account for all degrees of freedom present in our experiments[2]. It is also likely that the extended modelling framework induces additional degenerate configurations, cf. [89, 109], but their identification is beyond the scope of our work.

14.7.1 Reduced Observability

The identified ambiguities, related to further reduction of observability, are found by rewriting the observation model (14.1)

$$\lambda_{ij}\begin{bmatrix}x_{ij}\\y_{ij}\\1\end{bmatrix}=\mathbf{A}_i\mathbf{R}_i\Big[M_{\mu j}+\sum_{k=1}^{r}\beta_{ik}M_{kj}-C_i\Big]=$$

$$\mathbf{A}_i\mathbf{R}_i\Bigg[\left(1-\sum_{k=1}^{r}\beta_{ik}\alpha_k\right)M_{\mu j}+\sum_{k=1}^{r}\beta_{ik}\big(M_{kj}+\alpha_k M_{\mu j}+T_k\big)-\left(C_i+\sum_{k=1}^{r}\beta_{ik}T_k\right)\Bigg] \ ,$$

where $\mathbf{A}_i$, $\mathbf{R}_i$ and C_i are the internal parameters, rotation matrix and camera center corresponding to $\mathbf{P}_i$ respectively, i.e. $\mathbf{P}_i=\mathbf{A}_i\mathbf{R}_i[I-C_i]$. The α_k are mode dependent scalars, and the T_k are 3–vectors. Define the scalar

$$\kappa_i=1-\sum_{k=1}^{r}\beta_{ik}\alpha_k \ ,$$

which is dependent on the β_{ik} and α_k. Then (14.1) is equivalent to:

$$\lambda_{ij}\begin{bmatrix}x_{ij}\\y_{ij}\\1\end{bmatrix} = \kappa_i\mathbf{A}_i\mathbf{R}_i\Big[M_{\mu j}+\sum_{k=1}^{r}\tilde{\beta}_{ik}\tilde{M}_{kj}-\left(\frac{1}{\kappa_i}C_i+\sum_{k=1}^{r}\tilde{\beta}_{ik}T_k\right)\Big] \ , \quad (14.13)$$

where

$$\tilde{\beta}_{ik}=\frac{\beta_{ik}}{1-\sum_{k=1}^{r}\beta_{ik}\alpha_k} \ ,$$

and

$$\tilde{M}_{kj}=M_{kj}+\alpha_k M_{\mu j}+T_k \ .$$

From (14.13) it is seen, that there is a *translation ambiguity* between the modes, $\mathbf{M}_k$, and the camera centers, C_i, accounting for $3r$ degrees of freedom – r denoting the number of modes. The κ_i are seen to induce an over all scaling, which due to the 2D coordinates being represented in homogeneous coordinates is an ambiguity. This *scaling ambiguity* accounts for r degrees of freedom, one for each α_k.

[2] Detected by the singularities of the Jacobian of the un–regularized objective function.

14.7.2 Parametrization Ambiguity

There is also an ambiguity in the parametrization of the deforming structure, but this does not represent a indeterminacy in the structure of the solution. These ambiguities are found by rearranging (14.2)

$$\begin{aligned} S_{ij} &= M_{\mu j} + \beta_{i1} M^{1j} + \cdots \beta_{ir} M_{rj} \\ &= \left[\begin{array}{cccc} M_{\mu j} & M_{1j} & \cdots & M_{rj} \end{array} \right] \left[\begin{array}{c} 1 \\ \beta_{i1} \\ \vdots \\ \beta_{ir} \end{array} \right] \\ &= \left[\begin{array}{cccc} M_{\mu j} & M_{1j} & \cdots & M_{rj} \end{array} \right] \mathbf{B}^{-1}\mathbf{B} \left[\begin{array}{c} 1 \\ \beta_{i1} \\ \vdots \\ \beta_{ir} \end{array} \right] \quad , \end{aligned} \tag{14.14}$$

where $\mathbf{B}$ is an invertible $(r+1) \times (r+1)$ matrix of the form

$$\mathbf{B} = \left[\begin{array}{cccc} 1 & 0 & \cdots & 0 \\ * & * & * & * \\ \vdots & \vdots & \vdots & \vdots \\ * & * & * & * \end{array} \right] \quad .$$

Here $*$ denotes an arbitrary value, and the form of the top row is implied by the coefficient of $M_{\mu j}$ being 1.These parametrization ambiguities account for $r(r+1)$ degrees of freedom.

It is noted, that in PCA analysis, these parametrization ambiguities are addressed by maintaining the canonical form of the eigen–vectors derived from the eigen–value decomposition.

14.7.3 Implications

The ambiguities of (14.13), as mentioned, have the effect of reducing the observability of the system. The translation ambiguity implies that it is impossible to detect whether the camera or the 3D structure – or a mixture of both – moved. The scaling ambiguity indicates that a deformation of the object consisting sole of a scaling cannot be detected from a movement of the camera, i.e. a close small object cannot be distinguished from a far large one. As for the parametrization ambiguities, they are just ambiguities of the notation and as such have no physical interpretation. The total degrees of freedom, are seen to be $r^2 + 5r + 7$, whereof $4r + 7$ are related to observability, as summed up in Table 14.1.

14.8 Bundle Adjustment

Similarly to traditional bundle adjustment [185, 211], we propose minimizing the reprojection errors in the images, given the observation model (14.1). This yields a “gold standard

Ambiguity	**dof**
Euclidian Similarity Transform	7
Translation Ambiguity	$3r$
Scaling Ambiguity	r
Observability	$\mathbf{4r+7}$
Parametrization Ambiguity	r^2+r
Total	$\mathbf{r^2+5r+7}$

Table 14.1: Degrees of freedom (dof) as a function of the number of modes, r.

solution"[89]. Due to the nature of the problem, a non–linear numerical approach is needed to perform the minimization, for which we use the algorithm of Levenberg and Marquardt [123, 129].

It is assumed that the cameras are calibrated, but the same framework and approach would work in the uncalibrated case as well. Each camera is parameterized with a rotation matrix and the coordinates of the camera center. The structure is parameterized as in (14.2).

14.8.1 Regularization

The scaling and translation ambiguities implies that the different time instances of the object cannot be aligned properly, making results hard to interpret. A natural way to restrict these ambiguities is to impose a cost for two consecutive object instances to differ, i.e.

$$\delta\sum_{i=1}^{m}\sum_{j=2}^{n}||S_{ij}-S_{i-1,j}||_2^2 \ , \tag{14.15}$$

where δ is a small constant, e.g. $\delta = 10^{-4}$. This prior states, that if there is an ambiguity of how the 3D structure should move relative to the world coordinate system and scale, then stay stationary. This is the regularization used in [6].

As mentioned in Section 14.6, shrinkage of the structure could also be applied in order to address possible over parametrization. This is done by adding the following cost

$$\rho\sum_{ik}\beta_{ik}^2+\sum_{k}(||M_k||_F-1)^2 \ , \tag{14.16}$$

where $||\cdot||_F$ denotes the Frobenius norm, and ρ is a small constant dependent on the image noise and the size of the β_{ik}. This shrinkage imposes a small penalty on more complex models, whereby the optimization will 'try' to express as much as possible by camera motion.

14.8.2 Convergence

An unsolved issue is the convergence of the non–linear optimization. When using simulated data, we are not always able to achieve the global minimum. Our hypothesis is that this is either due to local minima in the object function, but more likely it is caused by the object function being extremely flat or ill–conditioned near the optimum. This is especially a problem if the model contains more modes than the data, i.e. r is too large. Solving this problem is the primal focus of our further research on this topic, since this is needed to rigorously validate the approach, as e.g. mentioned in Section 14.6.

A likely solution to this convergence problem is a more intelligent fixing of the gauge freedoms as described in [43, 44, 132]. This is likely to have the effect that the objective function will become more well conditioned. Apart from this, using the methods of [43, 44, 132] for gauge fixing should also give a much lower variance on the estimated structure and motion.

It should be noted that the convergence problem has not been greater than good results have been achieved. The numerical optimization is just not good enough for us to make a full rigorous evaluation, since e.g. it is impossible to conclude if a worse fit is due to a change in the data or due to a short coming of the numerical optimization.

14.9 Experimental Results

As a process of validating the proposed framework we ran it on real data, specifically 16 frames of a skeleton doll as illustrated Figures 14.2, 14.3, 14.4 and 14.5. This was experiment was made in conjunction with [6], and as such used the accompanying model selection approach as discussed in Section 14.6. The result of the reconstruction is shown in Figure 14.5, where the estimated structure for all the frames are depicted. It is seen that these resulting structures correspond well with the motion performed by the skeleton doll. To give the reader a feeling for the modes, the three chosen modes as well as the mean $\mathbf{M}_{\mu}$ are illustrated in Figure 14.4. The Root Mean Square (RMS) errors between the measured and reprojected points are 3.0 and 1.9 pixels after the affine factorization and after the bundle adjustment, respectively. Considering that the features were tracked by hand in images of size 960×1280 pixels, the resulting errors are indeed plausible.

As a more rigorous testing, we simulated images of a shaking house, see Figure 14.7. This enabled us to use a known ground truth for evaluation. We – among others – evaluated the approach by increasing the noise, and comparing the estimated fit to the ground truth. The estimated fit was evaluated by calculating the Procrustes distance, cf. [57] [3], between the estimated structure and the ground truth. This was done for each time instance and the resulting distances were averaged as a result.

A result is seen in Figure 14.6, where the number of modes was held fixed, to the correct number of two varying modes. It is noted, that the proposed scheme degrades gracefully as the data gets worse, i.e. the noise level increases. However, it should also be noted, that for no noise the fit is not perfect, i.e. distance 0 corrected for machine precision. This has to do with the convergence problems discussed in Section 14.8. This problem became even worse when an automatic model selection scheme was used, in that wrong choice of modes made the convergence properties worse.

[3] The Procrustes distance between two shapes is the mean squared distance between them after they have aligned with an Euclidean similarity transform and normalized. This is a standard measure with in shape statistics.

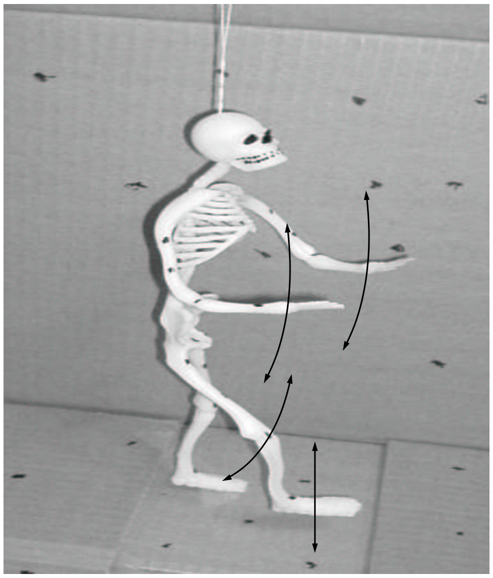


Figure 14.2: A frame of the skeleton sequence, with arrows denoting the way it moves.

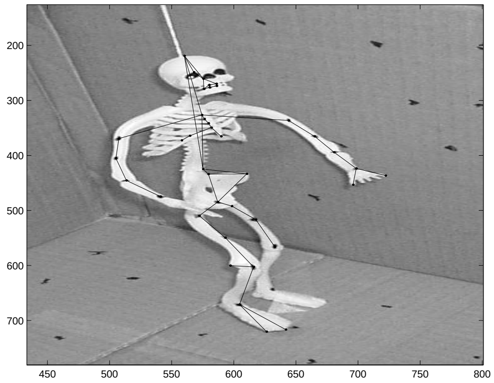


Figure 14.3: The features of the skeleton sequence and the lines connecting them for comprehendable 3D illustrations.

14.10 Conclusion and Discussion

A conclusion of this yet unfinished work is; that a seemingly good theoretical treatment of the problem has been developed. This includes identification of the added ambiguities, and some seemingly fruitful ideas on the model selection problem. However, the transition to practical implementation is not as straight forward as we had expected, specifically in regards to the convergence of the non–linear optimization.

As such, the next step in our investigation of the deformable structure from motion problem is to examine better numerical methods. With out better convergence, it is extremely hard – if not impossible – to fully validate the algorithm, and develop a fully integrated system including model selection. The reason for this is that without a better converging non–linear optimization, it is unknown whether a poor estimate is achieved due to a suboptimal approach or just poor convergence in the given instance.

When this issue has been resolved, we are also planning to further investigate approaches for model selection.

However, the preliminary experimental results are seen to give a reasonable solution to the problem yielding a crude experimental validation of the overall approach.

Acknowledgement

We would like to thank Poul Thyregod for invaluable help and discussion in regard to our variance estimation. We would also like to thank Bjarne Ersbøll for discussions on the model selection issue.

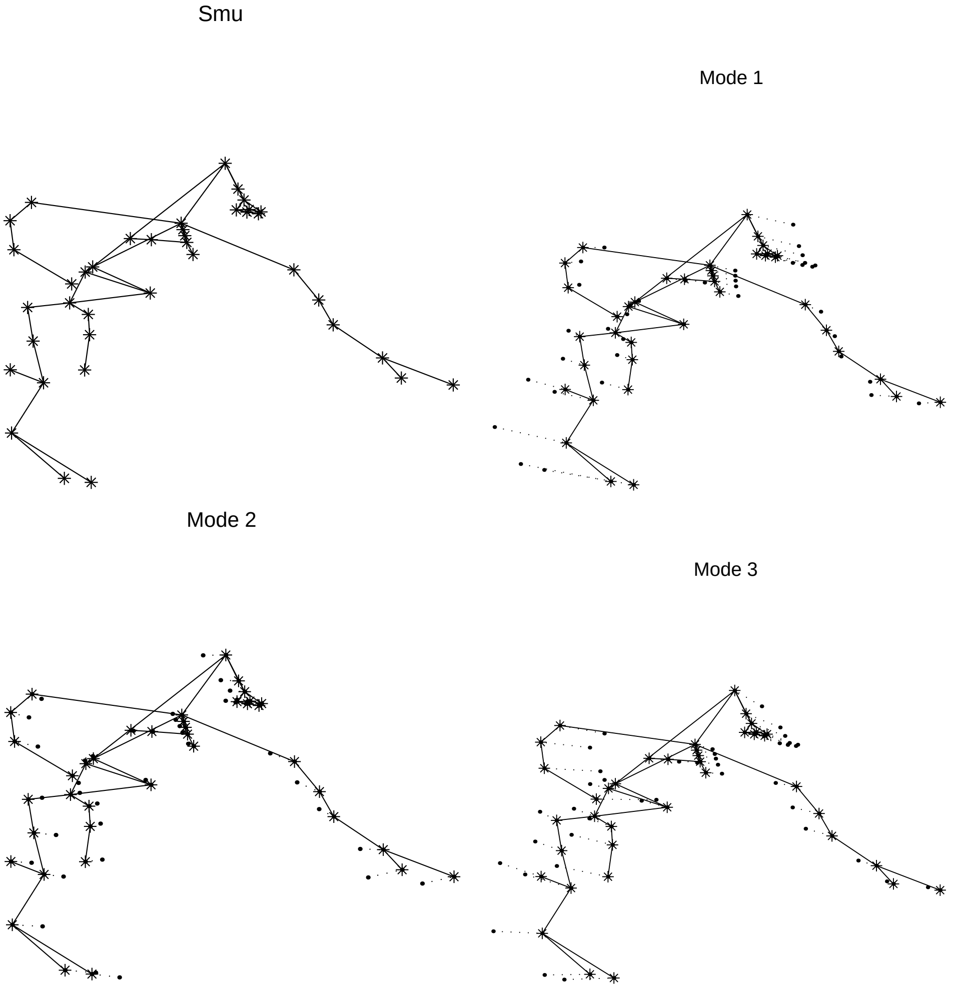


Figure 14.4: The modes of the skeleton sequence. (Upper Left) The mean, $\mathbf{M}_{\mu}$. The dotted lines in the following 3 figures denote the deformation from the mean captured by a given mode. (Upper Right) Mode 1, (Lower Left) Mode 2, (Lower Right) Mode 3

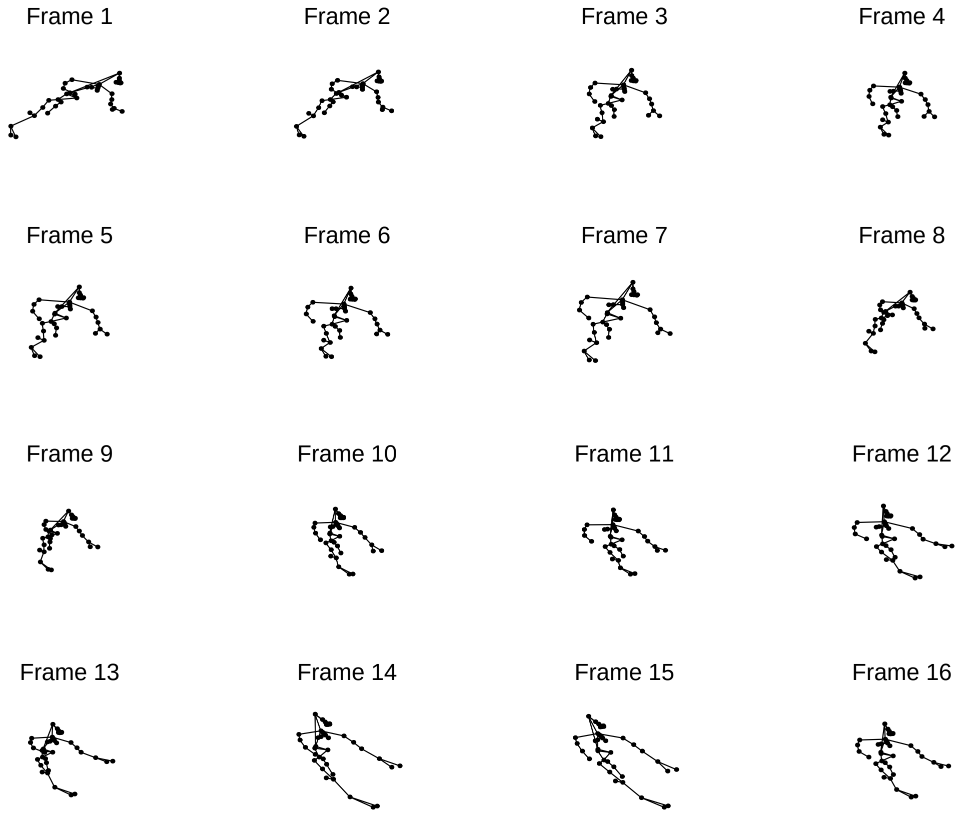


Figure 14.5: The reconstructed 3D structure of the skeleton doll captured in the skeleton sequence. See Figure 14.3 for an interpretation of the dots and lines.

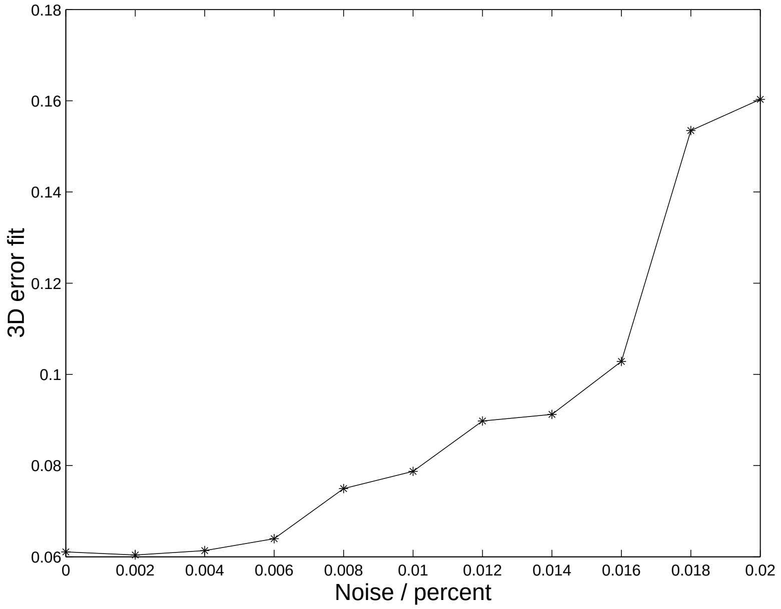


Figure 14.6: Reconstruction results for increasing noise using the house sequence, see Figure 14.7. The added noise is Gaussian where the ratio between the image size and standard deviation is denoted as a percentage along the abscissa.

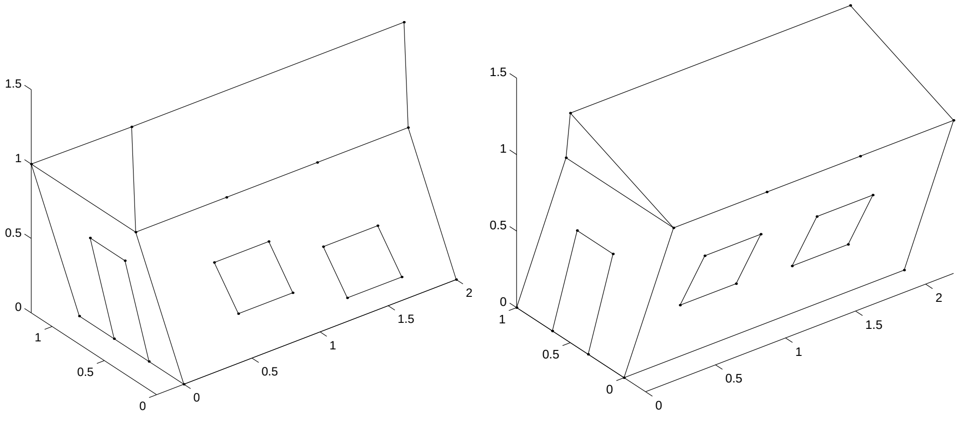


Figure 14.7: Two sample instances of the house structure used for simulation.

CHAPTER 15

Structure Estimation and Surface Triangulation of Deformable Objects

by: Charlotte Svensson, Henrik Aanæs and Fredrik Kahl

Abstract

A system is developed that from an image sequence of a deformable object automatically extracts features and tracks them through the sequence, estimates the non-rigid 3D structure and finally computes a surface triangulation. Also the camera motion is acquired. The object is supposed to deform according to a linear model, while the motion of the camera can be arbitrary. No domain specific prior of the object is required.

For the structure estimation a two-step approach is used, where we first obtain an initial estimate of the structure and motion, and then obtain an optimal solution via a non-linear optimization scheme. The triangulation is optimized to yield a non-rigid faceted surface that well approximates the true 3D surface.

15.1 Introduction

The estimation of structure and motion from image sequences is one of the most studied problems within computer vision. However, almost all the efforts in this area have dealt with rigid objects. Since the world is not a rigid place, it is important to have a system working for deforming objects as well. A common approach to the non-rigid problem is to use a prior model of the object, for example when human body or facial motion is studied [186, 182, 120].

We do not use a prior model, but employ the Principal Component Analysis (PCA) framework, whereby the object is supposed to deform according to a linear model. This type of model is fairly general and have proven to be very effective for expressing many types of deforming objects, e.g. [45]. In the works by [25, 209] such a linear model was used and the structure was estimated via a factorization algorithm. We extend this approach by applying a modified bundle adjustment algorithm to minimize the ML-error.

However, the main novelty compared to previos work is the improved surface modeling. We use the optimized structure to compute a non-rigid surface triangulation, using an approach similar to that of Morris & Kanade [137].

15.2 Tracking

The image sequence is supposed to be taken by a video camera. The feature points are tracked through the sequence using a standard low-level tracking technique, where the correlation of a small window around the feature point between two consecutive frames is used to get the best whole-pixel position. Then we optimize on sub-pixel level, allowing a small affine transformation of the patch.

Without the use of a prior model, tracking can also be facilitated using optical flow, as was introduced by Lucas & Kanade [127]. Applying rank constraints to the flow field helps to overcome the aperture problem, and has been used for both rigid [101] and non-rigid scenes [209].

15.3 Approximate Solution

15.3.1 Model Description

The structure of frame i is denoted by $\mathbf{S}_i = \begin{bmatrix} Q_{i1} & \cdots & Q_{in} \end{bmatrix}$, where Q_{ij} denotes the 3D coordinates of point j in frame i. The Principal Component Analysis (PCA) framework is employed, whereby the object is supposed to deform according to a linear model, i.e.

$$\mathbf{S}_i = \mathbf{S}_\mu + \sum_{k=1}^{r} \beta_{ik} \mathbf{S}_k \quad , \tag{15.1}$$

where β_{ik} is a scalar, $\mathbf{S}_k$ is a 3D mode of variation and $\mathbf{S}_\mu$ is the mean shape. However, we only have the 2D coordinates $w_{ij} = [\, x_{ij} \; y_{ij} \,]^T$, which are the 2D projections of the features Q_{ij}:

$$\lambda_{ij} \begin{bmatrix} x_{ij} \\ y_{ij} \\ 1 \end{bmatrix} = \mathbf{P}_i \begin{bmatrix} Q_{ij} \\ 1 \end{bmatrix} \quad , \tag{15.2}$$

where $\mathbf{P}_i$ is a 3×4 projection matrix. Hence, the problem is to estimate the camera motion $\mathbf{P}_i$ and the structure $\mathbf{S}_i$, i.e. both the mean shape, its modes of variation and the scalars

β_{ik}, from the given image data w_{ij}. Also the number of modes of variation, r, needs to be selected. If too few modes are used, the model cannot fully express the non–rigid structure, whereas excess modes would lead to modeling noise. How this model selection can be done automatically was described in [1].

15.3.2 Motion Estimation

An initial estimate of the camera motion is obtained assuming a rigid structure and solving for structure and motion. This can be done applying the fast factorization technique by Tomasi and Kanade [204], which assumes a linear approximation of the perspective camera model, or with some other standard structure and motion estimation technique, see e.g. [89].

15.3.3 Varying Structure Estimation

With the approximate motion estimate, the remaining task in getting an approximate solution is estimating the structure, i.e. $\mathbf{S}_\mu$, $\mathbf{S_k}$ and β_{ik}.

We note that good estimates of mean and variance of Gaussian distributed variables are obtained by computing the mean of the observations and the squared residuals with regards to this mean. However, full information of the $\mathbf{S}_i$ is unavailable, since the images are only 2D projections hereof. Thus, an image can be viewed as having a 3D observation with high uncertainty along the viewing direction. Hence, we form a weighted mean, where the weights V_i, of size 3×3, capture the direction where there is no information. With $\mathbf{S}_i^{dir}$ denoting the direct estimate of the structure, the weighted mean becomes

$$\mathbf{S}_\mu = \left(\sum_{i=1}^{m} V_i \right)^{-1} \sum_{i=1}^{m} V_i \mathbf{S}_i^{dir} \quad , \tag{15.3}$$

and the variance

$$\mathbf{S_\Sigma} = \mathcal{V}^{-1} \sum_{i=1}^{m} \bar{V}_i \left(\bar{\mathbf{S}}_i^{dir} - \bar{\mathbf{S}}_\mu \right) \left(\bar{\mathbf{S}}_i^{dir} - \bar{\mathbf{S}}_\mu \right)^T \bar{V}_i^T \quad , \tag{15.4}$$

where $\bar{V}_i$ and $\mathcal{V}$ are $3n \times 3n$ matrices given by

$$\bar{V}_i = \begin{bmatrix} V_i & & 0 \\ & \ddots & \\ 0 & & V_i \end{bmatrix} \quad \text{and} \quad \mathcal{V} = \sum_{i=1}^{m} \bar{V}_i \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} \bar{V}_i^T \quad .$$

The formulas for V_i and $\mathbf{S}_i^{dir}$ are given in [1].

After the model selection, whereby $\mathbf{S}_k$, $k = 1, ..., r$ are deducted from $\mathbf{S_\Sigma}$, the β_{ik} can be found by linear least squares minimization between the model and the image data.

15.4 Perspective Solution

15.4.1 Optimal Solution

Similar to traditional bundle adjustment [185], we propose to use a non–linear optimization algorithm on the observation model (15.2) to get a "gold standard solution"[89]. The collection of object points are parameterized by (15.1), and a Levenberg–Marquardt approach is applied in order to minimize the reprojection errors in the images.

We assume that the cameras are calibrated, but the same framework and approach would work in the uncalibrated case as well. Each camera is parameterized with a rotation matrix and the coordinates of the camera centre.

15.4.2 Ambiguities

In the rigid case, there is an ambiguity concerning the world coordinate system and global scale, i.e. the structure and motion can only be determined up to an unknown Euclidean transformation [89]. This ambiguity naturally extends to the non-rigid case. In addition, each mode in the linear model (15.1) introduces four extra degrees of freedom in the reconstruction. In [6] it was shown that this results in an ambiguity concerning relative translation and scale between the camera centres and the deforming modes of the object. This ambiguity is restricted by imposing a cost for two consecutive instances to differ, as a regularizing prior.

Also, there is an ambiguity concerning the parameterization itself, i.e. between (i) the mean $\mathbf{S}_\mu$ and the modes $\mathbf{S}_k$ and (ii) the weights β_{ik}. This introduces $r(r+1)$ extra degrees of freedom for r modes. They will not change the solution, but may slow down the convergence. More details are given in [1].

15.5 Surface Triangulation

15.5.1 Surface Model

It is a standard technique in computer graphics to represent a surface with a triangulation, giving a faceted surface, see e.g. [74]. Our surface model is described by the 3D points, $\mathbf{S}_i$, from Section 15.4 together with a triangulation, T, and a texture map, A. The triangulation specifies a set of edges and faces connecting all the 3D points in such a way that one faceted surface is created. Since we are dealing with deformable objects, a specific triangle, or facet, in the model has different shape and position for each frame. The texture map for the triangle is however constant through the sequence, since we assume constant lighting and a lambertian reflectance model.

For a given set of points on a surface, the triangulation is not unique, and our goal is to find the triangulation for which the faceted surface best matches the true object surface. For this optimal triangulation, a corresponding texture map is computed.

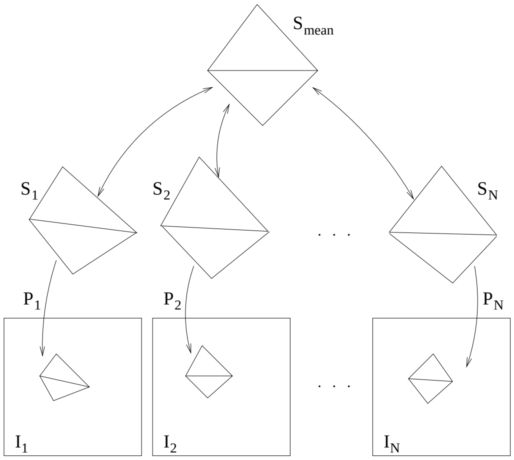


Figure 15.1: The surface model is illustrated for two triangles. Here, S means the 3D points plus the triangulation.

15.5.2 Surface Estimation

For a given triangulation, the texture map is easily found from the image sequence by mapping the images onto the triangulation. One particular triangle corresponds to a 3D facet and, if not occluded, an image triangle for each frame, cf. Figure 15.1. Now consider one such triangle. For each frame, the texture of the image triangle is mapped onto the mean triangle. For a good triangulation, the facets lies close to (the same part of) the true surface for all frames. This means that the mapped textures will be more or less the same. A facet not coinciding with the surface will look very different in different frames due to rotation and deformation of the model. Hence, optimizing the triangulation corresponds to minimizing the variance of the mapped texture triangles,

$$\Sigma = \sum_{i=1}^{N} (A_i - A_\mu)^2 \ , \tag{15.5}$$

where A_i denotes the texture map obtained from all triangles visible in frame i, A_μ is the mean texture across all frames and N is the number of frames. In the optimal case all A_i will be the same, i.e. $A_i = A$.

To optimize the triangulation we use the method described in [137]. A new triangulation is obtained from edge swapping. Two adjacent triangles share an edge and two vertices, and two new triangles are found by deleting this common edge and making a new one between the two vertices of the triangles that were not in common. Which edges to swap is found by a greedy search algorithm, which at each iteration finds the edge swap that will reduce the cost (15.5) the most. Once the optimized triangulation, T, is found, the texture map, A, is given by the mean texture across all the frames, A_μ.

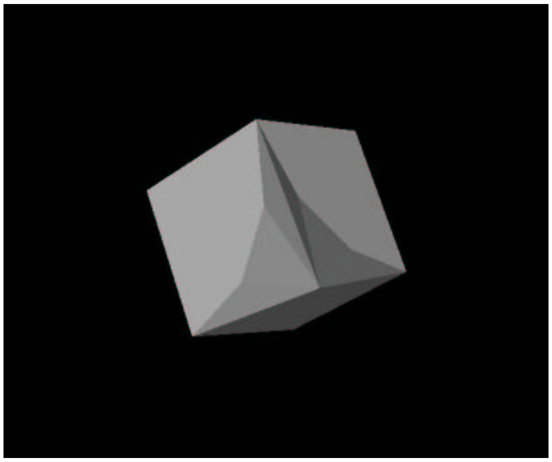


Figure 15.2: The structure subject to some deformation. The box is shown without texture but with lighting for better visualization.

15.6 Experimental Results

15.6.1 Synthetic Data

The triangulation algorithm was first run on a synthetic data set consisting of a box with a checkerboard pattern. Three sides of a box is constructed by 13 nodes, i.e. 7 corner nodes plus two nodes on each side, and a triangulation is made in such a way that three planes are obtained. The box is deformed by moving only the common corner node along a straight line, i.e. we have a one mode deformation where the rectangular box deforms to a structure consisting of several plane surfaces.

The same nodes that were made to build the box are used as nodes in the triangulation algorithm. The initialization of the triangulation gives a mesh not describing a rectangular box, but after optimization the triangulation is the same as the true one, cf. Figure 15.3.

15.6.2 Real Data

The second test sequence is a 135 frames video sequence of a talking person. Corner points from the first frame were extracted as features, and these are mainly located around the eyes, nose and mouth. In the structure estimation, every 5:th frame was used and some outliers had to be removed by hand. However, we are facing problems with the triangulation, possibly because the deformation is rather complex and we have only used two modes of deformation. To obtain a smoother, more appealing, triangulated surface, we also need to have more points at the cheeks and forehead, but such points are very hard to track. This work is still in progress.

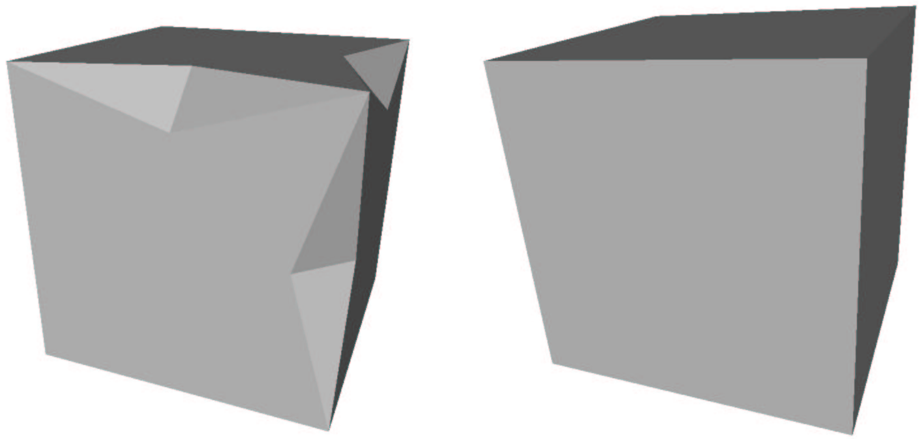


Figure 15.3: The mean shape described by the initial (left) and final (right) triangulation.

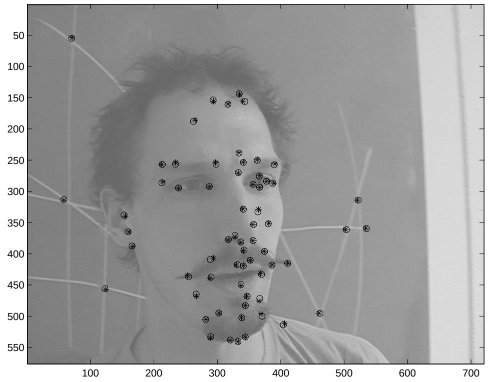


Figure 15.4: Tracked (*) and reprojected (o) points after structure estimation.

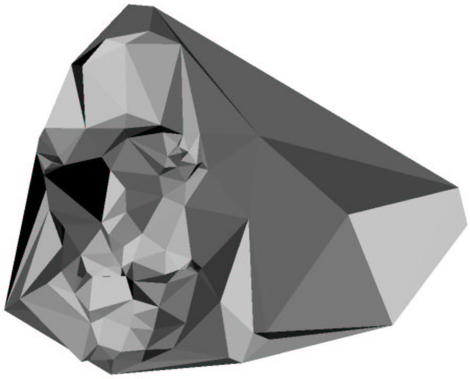


Figure 15.5: A triangulation of the surface. Note that the background points are part of the model.

Bibliography

[1] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. Technical report, Centre for Mathematical Sciences, Lund University, January 2002.

[2] H. Aanæs, R. Larsen, and J.A. Bærentzen. Pde based surface estimation for structure from motion. In *Scandinavian Conference on Image Analysis 2003*, 2003.

[3] H. Aanæs and J. A. Bærentzen. Pseudo–normals for signed distance computation. In *Vision, Modeling, and visualization 2003, Munich, Germany*, 2003.

[4] H. Aanaes, R. Fisker, K. Aström, and J.M. Carstensen. Robust factorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(9):1215–1225, 2002.

[5] H. Aanæs, R. Fisker, and J. M. Carstensen. Robust structure and motion. In *The 9th Danish Conference on Pattern Recognition and Image Analysis, Aalborg*, pages 1–9, 2000.

[6] H. Aanæs and F. Kahl. Estimation of deformable structure and motion. In *Vision and Modelling of Dynamic Scenes*, 2002.

[7] H. Aanæs and F. Kahl. A factorization approach for deformable structure from motion. In *SSAB*, 2002.

[8] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Factorization with contaminated data. In *Presentation and abstract at Eleventh International Workshop on Matrices and Statistics*, 2002.

[9] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Factorization with erroneous data. In *Photogrametric Computer Vision*, 2002.

[10] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–77, 1995.

[11] E.H. Adelson and J.Y.A. Wang. Single lens stereo with a plenoptic camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):99–106, 1992.

[12] H. Akaike. A new look at the statistical model identification. *IEEE Trans. on Automatic Control*, 19(6):716–723, 1974.

[13] K. Åström. *Invariancy Methods for Point, Curves and Surfaces in Computational Vision*. PhD thesis, Lund Institute of Technology, 1996.

[14] W. Baarda. *S–transformations and Criterion Matrices*, volume Band 5 der Reihe 1 of *New Series*. Netherlands Geodetic Commission, 1973.

[15] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 236–242, 1992.

[16] A. E. Beaton and J. W Tukey. The fitting of power series, meaning polynomials, illustrated in band-spectroscopic data. *Technometrics*, 16(2):147–185, 1974.

[17] R. Berthilsson and K. Åström. Extension of affine shape. *Journal of Mathematical Imaging and Vision*, 11(2):119–136, 1999.

[18] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996.

[19] A. Blake and M. Isard. *Active Contours*. Springer–Verlag, London, UK., 1998. 352 pp.

[20] A.J. Booker, J.E. Dennis, P.D. Frank, D.B Serafini, V. Torczon, and M.W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, 1999.

[21] Boujou. by 2D3 Ltd., *http://www.2d3.com/*.

[22] M. Brand. Morphable 3d models from video. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 2:456 –463, 2001.

[23] M. Brand. Incremental singular value decomposition of uncertain data with missing values. *Computer Vision - ECCV 2002. 7th European Conference on Computer Vision. Proceedings, Part I (Lecture Notes in Computer Science Vol.2350)*, pages 707–20, 2002.

[24] M. Brand and R. Bhotika. Flexible flow for 3d nonrigid tracking and shape recovery. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:315–322, 2001.

[25] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*, pages 690–6 vol.2, 2000.

[26] J. A. Bærentzen and H. Aanæs. Computing discrete signed distance fields from triangle meshes. Technical Report 21, IMM, DTU, 2002.

[27] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, second edition, 2000.

[28] M. J. Brooks and B. K. P. Horn. *Shape and Source from Shading*. MIT Press, Cambridge, MA, 1989.

[29] L. Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, December 1992.

[30] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. ii. the new algorithm. *Journal of the Institute of Mathematics and Its Applications*, 6(3):222–31, 1970.

[31] T. Buchanan. The twisted cubic and camera calibration. *Computer Vision, Graphics and Image Processing*, 42:130–132, 1988.

[32] N.A. Campbell. Robust procedures in multivariate analysis. i. robust covariance estimation. *Applied Statistics*, 29(3):231–7, 1980.

[33] S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *Int'l J. Computer Vision*, 27(3), 1998.

[34] J.M. Carstensen(Editor). *Image Analysis, Vision and Computer Graphics*. Tehnical University of Denmark, 2002.

[35] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int'l J. Computer Vision*, 1997.

[36] Yang Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–55, 1992.

[37] Y. Choe and R.L. Kashyap. 3-d shape from a shaded and textural surface image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(9):907–919, 1991.

[38] S. Christy and R. Horaud. Euclidian shape and motion from multiple perspective views by affine iterations. Technical Report 2421, INRIA, December 1994.

[39] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iteration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, November 1996.

[40] L. D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 666–673, 1996.

[41] R.T. Collins. A space-sweep approach to true multi-image matching. *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 358–363, 1996.

[42] A. Colosimo, A. Sarti, and S. Tubaro. Image-based object modeling: a multiresolution level-set approach. *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2:181 –184 vol.2, 2001.

[43] M. A. R. Cooper and P. A. Cross. Statistical concepts and their application in photogrammetry and surveying. *Photogrammetric Record*, 12(71):637–663, 1988.

[44] M. A. R. Cooper and P. A. Cross. Statistical concepts and their application in photogrammetry and surveying (continued). *Photogrammetric Record*, 13(77):645–678, 1991.

[45] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models – their training and application. *Computer Vision, Graphics and Image Processing*, 61(1):38–59, January 1995.

[46] J. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *Int'l J. Computer Vision'98*, 29(3):159–179, 1998.

[47] I.J. Cox, S.L. Higorani, S.B.Rao, and B.M. Maggs. A maximum likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–67, 1996.

[48] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. In *NATO Advanced Research Workshop on Confluence of Computer Vision and Computer Graphics, Ljubljana, Slovenia*, pages 25–47, 2000.

[49] F. Dachilie and A. Kaufman. Incremental triangle voxelization. *Proceedings Graphics Interface 2000*, pages 205–12, 2000.

[50] F. De la Torre and M. J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(2):117–142, 2003.

[51] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 1:362–369 vol.1, 2001.

[52] P. Debevec, Yizhou Yu, and G. Borshukov. Efficient view-dependent image-based rendering with projective texture-mapping. *Rendering Techniques '98. Proceedings of the Eurographics Workshop*, pages 105–16, 329, 1998.

[53] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *SIGGRAPH - Computer Graphics*, pages 11–20, 1996.

[54] D. DeCarlo. Towards real-time cue integration by using partial results. *Computer Vision - ECCV 2002. 7th European Conference on Computer Vision. Proceedings, Part IV (Lecture Notes in Computer Science Vol.2353)*, pages 327–42, 2002.

[55] D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *Int'l J. Computer Vision'94*, 15(1–2):123–141, 1995.

[56] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces by volumetric regularization using radial basis functions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(10):1358–1371, 2002.

[57] I. L. Dryden and K. V. Mardia. *Statistical Shape Analysis*. Wiley, 1998.

[58] P. Dutre, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A K Peters, Ltd., first edition, 2003.

[59] M. L. Eaton. *Multivariate Statistics, A Vector Space Approach*. John Wiley & Sons, Inc., 1983.

[60] P. Eisert, E. Steinbach, and B. Girod. Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views. *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, 6:3509–3512 vol.6, 1999.

[61] L. Falkenhagen. Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces. In *Proc. of European Workshop on combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg*, 1994.

[62] L. Falkenhagen. Hierarchical block-based disparity estimation considering neighbourhood constraints. In *International workshop on SNHC and 3D Imaging, Rhodes, Greece.*, 1997.

[63] O. Faugeras. Stratification of three-dimensional vision: Projective, affine and metric representations. *Journal of the Optical Society of America*, 12:465–484, 1995.

[64] O. Faugeras and R. Keriven. Variatinal principles, surface evaluation, pde's, level set methods and the stereo problem. Technical Report 3021, INRIA, October 1996.

[65] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. *Computer Vision - ECCV'98. 5th European Conference on Computer Vision. Proceedings*, pages 379–93 vol.1, 1998.

[66] O. Faugeras and R. Keriven. Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *Image Processing, IEEE Transactions on*, 7(3):336 –344, 1998.

[67] O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3-d reconstruction of urban scenes from sequences of images. Technical Report 2572, INRIA, June 1995.

[68] O. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In G. Sandini, editor, *European Conf, Computer Vision*, volume 588 of *Lecture notes in Computer Science*, pages 321–334. Springer-Verlag, 1992.

[69] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, Cambridge, Massachussets, USA, 2001.

[70] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[71] A. W. Fitzgibbon and A. Zisserman. Automatic 3D model acquisition and generation of new images from video sequences. In *Proceedings of European Signal Processing Conference (EUSIPCO '98), Rhodes, Greece*, pages 1261–1269, 1998.

[72] A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-d reconstruction of independently moving objects. In *European Conf, Computer Vision'2000*, pages 891–906, 2000.

[73] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13(3):317–22, 1970.

[74] J.D. Foley, A. van Dam, S.K. Feiner, and J.F. Hughes. *Computer Graphics: Principles and Practice in C*. Addison-Wesley, 2 edition, 1995.

[75] D. A. Forsyth and J. Ponce. *Computer Vision – A modern approach*. Prentice Hall, 1 edition, 2002.

[76] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: combining multi-image stereo and shading. *International Journal of Computer Vision*, 16(1):35–56, 1995.

[77] A. S. Glassner. *Computing Surface Normals for 3D Models*, pages 562–566. Academic Press, 1990.

[78] D. Goldfarb. A family of variable metric methods derived by variational means. *Maths. Comp.*, 24:23–26, 1970.

[79] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. Lumigraph. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 43–54, 1996.

[80] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–629, 1971.

[81] J.C. Gower. Generalized Procrustes analysis. *Psychometrika*, 40:33–50, 1975.

[82] A. Guéziec. "meshsweeper": Dynamic point–to–polygonal mesh distance and applications. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):47–60, 2001.

[83] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of vornoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, April 1985.

[84] N. Guilbert, H. Aanæs, and R. Larsen. Integrating prior knowledge and structure from motion. In *Proceedings of the Scandinavian Image Analysis (SCIA'01)*, pages 477–481, Bergen, Norway, 2001.

[85] F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley & Sons, 1986.

[86] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. Alvey Conf.*, pages 189–192, 1988.

[87] R. I. Hartley. A linear method for reconstruction from lines and points. In *Int'l Conf. Computer Vision*, pages 882– 887, 1995.

[88] R. I. Hartley, E. Hayman, L. de Agapito, and I. Reid. Camera calibration and the search for infinity. In *Int'l Conf. Computer Vision*, pages 510–517, 1999.

[89] R. I. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 2000.

[90] R.I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–93, 1997.

[91] T. Hastie, J. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Data Mining, Inference and Prediction*. Springer, 2001.

[92] A. Hertzmann and S.M. Seitz. Shape and materials by example: a photometric stereo approach. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:533–540, 2003.

[93] A. Heyden. *Geometry and Algebra of Multiple Projective Transformations*. PhD thesis, Lund Institute of Technology, 1995.

[94] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13):981–991, 1999.

[95] A. Heyden and K. Åström. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 438–443, 1997.

[96] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least-squares. *Commun. Statist.-Theor. Meth.*, A6(9):813–827, 1977.

[97] J. I. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30:179–185, 1965.

[98] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441 and 498–520, 1933.

[99] J. Huang, Y. Li, R. Crawfis, S.-C. Lu, and S.-Y. Liou. A complete distance field representation. *Visualization, 2001. VIS '01. Proceedings*, pages 247–254, 2001.

[100] T.S. Huang and O.D Faugeras. Some properties of the e matrix in two-view motion estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 11(12):1310–1312, December 1989.

[101] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Int'l Conf. Computer Vision*, 1999.

[102] M. Irani and P. Anandan. Factorization with uncertainty. In *European Conf, Computer Vision'2000*, pages 539–553, 2000.

[103] J. Isidoro and S. Sclaroff. Stochastic mesh-based multiview reconstruction. *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pages 568–577, 2002.

[104] D.W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding*, 82(1):57–81, 2001.

[105] H. Jin, S. Soatto, and A.J. Yezzi. Multi-view stereo beyond lambert. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:171–178, 2003.

[106] H. Jin, A. Yezzi, and S. Soatto. Variational multiframe stereo in the presence of specular reflections. Technical Report TR01-0017, UCLA, 2001.

[107] M.W. Jones. The production of volume data from triangular meshes using voxelisation. *Computer Graphics Forum*, 15(5):311–18, 1996.

[108] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38:325–340, 1987.

[109] F. Kahl. *Geometry and Critical Configurations of Multiple Views*. PhD thesis, Lund Institute of Technology, September 2001.

[110] F. Kahl and K. Astrom. Motion estimation in image sequences using the deformation of apparent contours. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 939–42, 1998.

[111] T. Kanade and D. Morris. Factorization methods for structure from motion. *Phil. Trans. R. Soc. Lond.*, A(356):1153–1173, 1998.

[112] K. Kanatani. Motion segmentation by subspace separation and model selection. In *Int'l Conf. Computer Vision'2001*, pages 586–591, 2001.

[113] J. Karhunen and J. Joutsensalo. Generalizations of principal component analysis, optimization problems, and neural networks. *Neural Networks*, 8(4):549–562, 1995.

[114] J.R. Kender. Shape from texture: An aggregation transform that maps a class of textures into surface orientation. In *International Joint Conference on Artificial Intelligence*, pages 475–480, 1979.

[115] Y. Keselman, A. Shokoufandeh, M.F. Demirci, and S. Dickinson. Many-to-many graph matching via metric embedding. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:850–857, 2003.

[116] R. Kimmel and A. M. Bruckstein. Global shape from shading. In *Computer Vision and Image Understanding*, pages 120–125, 1995.

[117] Ron Kimmel and Irad Yavneh. An algebraic multigrid approach for image analysis. *SIAM Journal on Scientific Computing*, 24(4):1218–1231, 2003.

[118] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[119] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:307–314 vol.1, 1999.

[120] A. Lanitis, C. J. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Trans. of Pattern recognition and Machine Intelligence*, 19(7):743–756, 1997.

[121] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Prentice-Hall, Englewood Cliffs, NJ, 1974.

[122] C.-H. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26(2):125–43, 1985.

[123] K. Levenberg. A method for the solution of certain problems in least-squares. *Quart. J. of Appl. Math.*, 12:164–168, 1944.

[124] M. Levoy, S. Rusinkiewcz, M. Ginzton, J. Ginsberg, K. Pulli, D. Koller, S. Anderson, J. Shade, B. Curless, L. Pereira, J. Davis, and D. Fulk. The digital michelangelo project: 3d scanning of large statues. *Computer Graphics Proceedings. Annual Conference Series 2000. SIGGRAPH 2000. Conference Proceedings*, pages 131–44, 2000.

[125] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 392:133–135, 1981.

[126] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface reconstruction algorithm. *Computer Graphics (Siggraph'87)*, 21(4):163–169, 1987.

[127] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. Int. joint Conf. on Artificial Intelligence*, pages 674–579, 1981.

[128] Y. Ma, J. Kosecka, and S. Sastry. Linear differential algorithm for motion recovery: A geometric approach. *Int'l J. Computer Vision*, 36(1):71–89, 2000.

[129] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[130] S. Mauch. A fast algorithm for computing the closest point and distance transform. Technical report, Applied and Computational Mathematics, California Institute of Technology, 2000.

[131] N. Max. Weights for computing vertex normals from facet normals. *Journal of Graphics Tools*, 4(2):1–6, 1999.

[132] P. Meissl. *Least Squares Adjustment. A Modern Approach*. Geodetic Institute of the Technical University Graz, 1982.

[133] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):580–591, 1993.

[134] T. P. Minka. Automatic choice of dimensionality for PCA. In *NIPS*, pages 598–604, 2000.

[135] The modeling by videotaping group at the robotics institute, carnegie mellon university. http://www.ius.cs.cmu.edu/ IUS/mbvc0/www/modeling.html.

[136] D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Int'l Conf. Computer Vision'98*, pages 696–702, January 1998.

[137] D. Morris and T. Kanade. Image-consistent surface triangulation. In *IEEE Conf. Computer Vision and Pattern Recognition'2000*, pages 332–338, 2000.

[138] M. G. Mostafa, E. E. Hemayed, and A. A. Farag. Target recognition via 3d object reconstruction from image sequence and contour matching. *Pattern Recognition Letters*, 20(11-13):1381–1387, 1999.

[139] K. Museth, D.E. Breen, R.T. Whitaker, and A.H. Barr. Level set surface editing operators. *ACM Transactions on Graphics*, 21(3):330–8, 2002.

[140] P.J. Narayanan and T. Kanade. Virtual worlds using computer vision. *Computer Vision for Virtual Reality Based Human Communications, 1998. Proceedings., 1998 IEEE and ATR Workshop on*, pages 2–13, 1998.

[141] P.J. Narayanan, P.W. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. *Computer Vision, 1998. Sixth International Conference on*, pages 3–10, 1998.

[142] S. K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of lambertian, specular, and hybrid surfaces using extended light sources. In *IEEE Workshop on Industrial Applications of Machine Intelligence and Vision*, 1989.

[143] H. B. Nielsen and H. Aanæs. Separation of structure and motion by data modification. Technical report, IMM, DTU, To appear.

[144] D. Nister. *Automatic Dense Reconstruction from Uncalibrated Video Sequences*. PhD thesis, Royal Institute of Technology, KTH, 2001.

[145] M. Okutomi and T. Kanade. A multiple-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(4):353–363, 1993.

[146] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[147] S. J. Osher and R. P. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 1st edition, November 2002.

[148] A. R. Dick P. H. S. Torr and R. Cipolla. Layer extraction with a bayesian model of shape. In *European Conf, Computer Vision'00*, pages 273–289, June 2000.

[149] S. Pankanti and A.K. Jain. Integrating vision modules: stereo, shading, grouping, and line labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(9):831–842, 1995.

[150] B.A. Payne and A.W. Toga. Distance field manipulation of surface models. *IEEE Computer Graphics and Applications*, 12(1):65 –71, 1992.

[151] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosoph. Mag*, 6(2):559–572, 1901.

[152] M. Pelillo. Matching free trees, maximal cliques, and monotone game dynamics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(11):1535–1541, 2002.

[153] A. Pentland and B. Horowitz. Recovery of nonrigid motion and structure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:730–742, 1991.

[154] A.P. Pentland. Local shading analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(2):170–87, 1984.

[155] PhotoModeler. by Eos Systems Inc., *http://www.photomodeler.com/*.

[156] C. Poelman and T. Kanade. A paraperspective factorizarion method for shape and motion recovery. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997.

[157] M. Pollefeys. Tutorial on 3d modeling from images. *In conjunction with ECCV 2000, Dublin, Ireland*, June 2000.

[158] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1:496–501 vol.1, 1999.

[159] M. Pollefeys, K. Reinhard, and L.V. Gool. Self-calibration and metric reconstruction of varying and unknown intrinsic camera parameters. *Int'l J. Computer Vision*, 32(1):7–25, 1999.

[160] M. Pollefeys and L. Van Gool. Stratified self-calibration with the modulus constraint. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(8):707–724, 1999.

[161] M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *European Conf, Computer Vision*, pages 837–851, 2002.

[162] R. Ramamoorthi. *A Signal-Processing Framework for Forward and Inverse Rendering*. PhD thesis, Stanford University, 2002.

[163] P.W. Rander, P.J. Narayanan, and T. Kanade. Recovery of dynamic scene structure from multiple image sequences. *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on*, pages 305–312, 1996.

[164] A. P. Rockwood and J. Winget. Three-dimensional object reconstruction from two-dimensional images. *Computer-Aided Design*, 29(4):279–285, 1997.

[165] T. Rodriguez, P. Sturm, M. Wilczkowiak, A. Bartoli, M. Personnaz, N. Guilbert, F. Kahl, M. Johansson, A. Heyden, J. M. Menendez, Ronda J. I., and F. Jaureguizar. Visire. photorealistic 3d reconstruction from video sequences. In *IEEE International Conference on Image Processing, Barcelona, Spain*, 2003. To appear.

[166] S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Computer Vision, 1998. Sixth International Conference on*, pages 492 –499, 1998.

[167] F.H. Ruymgaart. A robust principal component analysis. *Journal of Multivariate Analysis*, 11(4):485–97, 1981.

[168] R. Szeliski S. Baker and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Conf. Computer Vision and Pattern Recognition'98*, pages 434–441, 1998.

[169] S. Savarese and P. Perona. Local analysis for 3d reconstruction of specular surfaces. ii. *Computer Vision - ECCV 2002. 7th European Conference on Computer Vision. Proceedings (Lecture Notes in Computer Science Vol.2351)*, pages 759–74, 2002.

[170] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(3):7–42, 2002.

[171] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–72, 2000.

[172] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[173] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[174] J.A. Sethian. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[175] D.F. Shanno. Conditioning of quasi–newton methods for function minimization. *Maths. Comp.*, 24:647–656, 1970.

[176] ShapeSnatcher. by Eyetronics , *http://www.eyetronics.com/*.

[177] A. Shashua. Trilinearity in visual recognition by alignment. In *European Conf, Computer Vision*, pages 479–484, 1994.

[178] A. Shashua and A. Levin. Multi-frame infinitesimal motion model for the reconstruction of (dynamic) scenes with multiple linearly moving objects. *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 592–9 vol.2, 2001.

[179] A. Shokoufandeh, S. Dickinson, C. Jonsson, L. Bretzner, and T. Lindeberg. On the representation and matching of qualitative shape at multiple scales. *Computer Vision - ECCV 2002. 7th European Conference on Computer Vision. Proceedings, Part III (Lecture Notes in Computer Science Vol.2352)*, pages 759–75, 2002.

[180] A. Shokoufandeh, I. Marsic, and S. J. Dickinson. View-based object recognition using saliency maps. *Image and Vision Computing*, 17(5-6):445–460, 1999.

[181] H.-Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):854–67, 1995.

[182] H. Sidenbladh, M.J. Black, and D.J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European Conf, Computer Vision'2000*, pages 702–718, 2000.

[183] G. Slabaugh. *Novel Volumetric Scene Reconstruction Methods for New View Synthesis*. PhD thesis, Georgia Institute of Technology, 2002.

[184] G. Slabaugh, R. W. Schafer, and M. C. Hans. Multi-resolution space carving using level sets methods. In *The International Conference on Image Processing (ICIP)*, 2002.

[185] C.C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, 4:th edition, 1984.

[186] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Conf. Computer Vision and Pattern Recognition'2001*, volume I, pages 447–454, 2001.

[187] J. E. Solem, H. Aanæs, and A. Heyden. PDE based shape from specularities. In *Scale Space, Isle of Skye, UK*, 2003.

[188] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(4):344 –358, 1995.

[189] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Int'l Conf. Pattern Recognition'96*, pages 328 – 333, 1996.

[190] C. H. Séquin. Procedural spline interpolation in unicubix. *Proceedings of the 3rd USENIX Computer Graphics Workshop*, pages 63–83, 1986.

[191] K.A. Stevens. The visual interpretation of surface contours. *Artificial Intelligence*, 17(1-3):47–73, 1981.

[192] M.R. Stevens and J.R. Beveridge. Precise matching of 3-d target models to multisensor data. *Image Processing, IEEE Transactions on*, 6(1):126 –142, 1997.

[193] G. L. Strang van Hees. Variance–covariance transformations of geodetic networks. *Manuscripta Geodaetica*, 7(1):1–20, 1982.

[194] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1100–1105, 1997.

[195] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European Conf, Computer Vision'96*, volume 2, pages 709–720, 1996.

[196] C. Svensson, H. Aanæs, and F. Kahl. Structure estimation and surface triangulation of deformable objects. In *Scandinavian Conference on Image Analysis 2003*, 2003.

[197] R. Szeliski. A multi-view approach to motion and stereo. *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999.

[198] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. Technical Report UUCS-02-02, School of Computing, University of Utha, January 2002.

[199] D. Tell. *Wide baseline matching with applications to visual servoing*. PhD thesis, KTH, Stockholm, 2002.

[200] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–23, 2000.

[201] G. Thürmer and C.A. Wüthrich. Computing vertex normals from polygonal facets. *Journal of Graphics Tools*, 3(1):43–6, 1998.

[202] T.Y. Tian and M. Shah. Recovering 3d motion of multiple objects using adaptive hough transform. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(10):1178–1183, October 1997.

[203] M.E. Tipping and C.M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 61(3):611–622, 1999.

[204] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int'l J. Computer Vision'92*, 9(2):137–154, November 1992.

[205] P. H. S. Torr. Mlesac: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138–156, April 2000.

[206] P. H. S. Torr, A. W. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *Proc. 6th International Conference on Computer Vision, Bombay, India*, pages 485–491, Jan 1998.

[207] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In *Proc. of the SPIE - The International Society for Optical Engineering*, volume 2056, pages 432–443, 1993.

[208] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int'l J. Computer Vision*, 24(3):271–300, 1997.

[209] L. Torresani, D.B. Yang, E.J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, 1:493 –500, 2001.

[210] B. Triggs. The geometry of projective reconstruction i: Matching constraints and the joint image. In *Int'l Conf. Computer Vision*, pages 338–343, 1995.

[211] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Special sessions - bundle adjustment - a modern synthesis. *Lecture Notes in Computer Science*, 1883:298–372, 2000.

[212] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuro Science*, 3(1):71–86, 1991.

[213] R.T. Whitaker. A level-set approach to 3d reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–31, 1998.

[214] A.P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, 1981.

[215] H. Wold. Nonlinear estimation by iterative least squares procedures. *Research Papers in Statistics, Festschrift for J. Neyman*, pages 411–444, 1966.

[216] M. Woo, J. Neider, T. Davis, D. Shreiner, and OpenGL Architecture Review Board. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL, Version 1.2 (3rd Edition)*. Addison-Wesley Pub Co, 2002.

[217] L. Xu and A.L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *Neural Networks, IEEE Transactions on*, 6(1):131 –143, 1995.

[218] A.J. Yezzi and S. Soatto. Structure from motion for scenes without features. *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, 1:525–532, 2003.

[219] Y. Yu, P. Debevec, J. Malik, and T. Hawkins. Inverse global illumination: Recovering reflectance models of real scenes from photographs. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, pages 215–224, 1999.

[220] S. Zamir and K.R. Gabriel. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, 1979.

[221] R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.

[222] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *Int'l J. Computer Vision'98*, 27(2):161–195, 1998.

[223] Z. Zhang, R. Deriche, O. Faugeras, and Q. T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *AI Jnl*, 78(1):87–119, 1995.

[224] J. Y. Zheng and A. Murata. Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):913–920, 2000.

[225] M. Ziegler, L. Falkenhagen, R. Horst, and D. Kalivas. Evolution of stereoscopic and three-dimensional video. *Image Communication*, 14, 1998.

[226] A. Zisserman, A. Fitzgibbon, and G. Cross. Vhs to vrml: 3d graphical models from video sequences. *Multimedia Computing and Systems, 1999. IEEE International Conference on*, 1:51 –57 vol.1, 1999.

[227] A. Zisserman, P. Giblin, and A. Blake. The information available to a moving observer from specularities. *Image and vision computing*, 1989.