Linear and Bilinear Subspace Methods for Structure from Motion

Mei Han

February 2001

Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Thesis committee:
Takeo Kanade, Chair
Martial Hebert
Paul Heckbert
P. Anandan, Microsoft Research

Abstract

Structure from Motion (SFM), which is recovering camera motion and scene structure from image sequences, has various applications, such as scene modeling, robot navigation and object recognition. Most of previous research on SFM requires simplifying assumptions on the camera or the scene. Common assumptions are a) the camera intrinsic parameters, such as focal lengths, are known or unchanged throughout the sequence, and/or b) the scene does not contain moving objects. In practice, these are unrealistic assumptions. In this thesis we present a collection of reconstruction methods for dealing with image sequences taken with uncalibrated cameras and/or of multiple motion scenes.

The methods produce Euclidean reconstruction directly from feature point locations and are based on the bilinear relationship of camera motion and scene structure. For uncalibrated image sequences, we embed the camera intrinsic parameters within the camera motion representation. For image sequences of multiple motion scenes, we incorporate multiple motions into the scene structure representation. In this way, we derive linear and bilinear subspace constraints on the large amount of information integrated over the entire image sequences. By taking advantage of this redundant information we can achieve accurate and reliable reconstruction.

Firstly, we propose a uncalibrated Euclidean reconstruction method from multiple uncalibrated views. This method first performs a projective reconstruction using a bilinear factorization algorithm, and then converts the projective solution to a Euclidean one by enforcing metric constraints. We present three normalization algorithms to generate the Euclidean reconstruction and the intrinsic parameters. The first two algorithms are linear, one for dealing with the case that only the focal lengths are unknown, and another for the case that the focal lengths and the constant principal point are unknown. The third algorithm is bilinear, dealing with the case that the focal lengths, the principal points and the aspect ratios are all unknown.

Secondly, we present a linear method to reconstruct a scene containing multiple moving objects together with the camera motion. The number of the moving objects is automatically detected without prior motion segmentation. Assuming that the objects are moving linearly with constant speeds, we propose a unified geometrical representation of the static scene and the moving objects. This representation enables the embedding of the linear motion constraints into the scene structure, which naturally

leads to a factorization-based method.

Thirdly, we describe a method for multiple motion scene reconstruction from uncalibrated views. The method recovers the scene structure, the trajectories of the moving objects and the camera intrinsic (except skews) and extrinsic parameters simultaneously assuming that the objects are moving with constant velocities. We embed the assumptions within the scene representation and therefore propose a bilinear factorization algorithm to generate a projective reconstruction, and then impose metric constraints to compute the Euclidean reconstruction and the camera intrinsic parameters.

We also discuss other issues related to the accuracy and reliability of these reconstruction methods, such as minimum data requirement and gauge selection. The reconstruction methods have been tested on a series of synthetic sequences to evaluate the quality of the methods, and real image sequences to demonstrate their applicability.

Acknowledgements

I would like to thank my advisor, Takeo Kanade, for his insights, enthusiasm and hard working attitude which will always inspire me. I would also like to thank the other members of my thesis committee Martial Hebert, Paul Heckbert, and P. Anandan for their insightful suggestions and helpful feedback.

Thanks to Simon Baker for his valuable suggestions, comments and encouragement, Jianbo Shi, for his great help on my writing and presentation, Bob Collins and Yanxi Liu for many helpful discussions.

I would like to thank Daniel Morris for the countless fruitful discussions. Thanks to my best-in-the-world officemates David Larose and Teck Khim Ng for their support and friendship. I also want to express my gratitude to Long Quan, Amnon Shashua, Rakesh Kumar for their insightful suggestions.

Thanks to Harry Shum and Richard Szeliski from Microsoft Research for the fruitful internship. Thanks to Bo Zhang and Guangyou Xu from Tsinghua University for originally guiding me into this area of research.

Most of all, I would like to thank my husband, Wei Hua, for his endless love and support.

Contents

1 Introduction **1**

1.1 Problem definition 1

1.2 Related work 2

1.3 Thesis overview 5

2 Euclidean Reconstruction with Uncalibrated Cameras **11**

2.1 Review of Tomasi and Kanade's factorization method 11

2.1.1 Orthographic projection 12

2.1.2 World coordinate system location 14

2.1.3 Decomposition 14

2.1.4 Normalization 15

2.1.5 Motion and shape recovery 16

2.2 Projective reconstruction 16

2.3 Euclidean reconstruction 18

2.3.1 Normalization algorithm outline 18

2.3.2 Case 1: Unknown focal lengths 22

2.3.3 Case 2: Unknown focal lengths and constant principal point 23

2.3.4 Case 3: Unknown focal lengths, principal points and aspect ratios 24

2.4 Experiments 25

2.4.1 Synthetic examples 25

2.4.2 Real example 1: Building sequence 26

2.4.3 Real example 2: Grand Canyon sequence 29

2.4.4 Real example 3: Calibration setup 29

3 Multiple Motion Scene Reconstruction **35**

3.1 Feature points representation 35

3.2 Scene reconstruction 37

3.2.1 Moving world coordinate system location . . . 37
3.2.2 Decomposition . . . 38
3.2.3 Normalization . . . 38
3.2.4 Motion and shape reconstruction . . . 40
3.2.5 Summary of algorithm . . . 40
3.3 Degenerate cases . . . 41
3.3.1 Rank approximation . . . 42
3.3.2 Rank-$\mathbf{4}$ case . . . 42
3.3.3 Rank-$\mathbf{5}$ case . . . 43
3.4 Extensions to weak perspective and perspective projections . . . 46
3.4.1 Scene reconstruction under weak perspective projection . . . 46
3.4.2 Scene reconstruction under perspective projection . . . 49
3.5 Experiments . . . 52
3.5.1 Synthetic examples . . . 52
3.5.2 Real example 1: Toy sequence . . . 54
3.5.3 Real example 2: Smith Hall sequence . . . 55

4 Multiple Motion Scene Reconstruction with Uncalibrated Cameras 61
4.1 Projective reconstruction . . . 61
4.2 Euclidean reconstruction . . . 63
4.2.1 Moving world coordinate system location . . . 65
4.2.2 Normalization . . . 66
4.2.3 Camera calibration and scene reconstruction . . . 68
4.2.4 Algorithm outline . . . 69
4.3 Experiments . . . 69
4.3.1 Synthetic examples . . . 69
4.3.2 Real example . . . 72
4.4 Degenerate cases . . . 72
4.4.1 Rank-$\mathbf{4}$ case . . . 75
4.4.2 Rank-$\mathbf{5}$ case . . . 77

5 Reconstruction Analysis 79
5.1 Minimum data requirement . . . 79
5.1.1 Counting the arguments . . . 80
5.1.2 Analyzing the solution . . . 83
5.1.3 Empirical results . . . 86

5.2 Gauge selection 88
5.2.1 Gauge selection in static scene reconstruction 88
5.2.2 Gauge selection in multiple motion scene reconstruction 91

6 Conclusion **95**
6.1 Contributions 95
6.1.1 Theories 95
6.1.2 Systems 96
6.1.3 Applications 96
6.2 Future work 97
6.2.1 Critical motion sequences 97
6.2.2 Uncertainty modeling 98
6.2.3 Sequences with missing data 101
6.2.4 Dense shape recovery 102

A Homography-Based Scene Analysis from Image Sequences **105**
A.1 Introduction 105
A.2 Robust homography 106
A.2.1 Image intensity adjustment 106
A.2.2 Progressive transformation complexity 107
A.2.3 Robust estimation 108
A.3 Recovery of scene depth 108
A.3.1 Scene depth and homography 108
A.3.2 Camera motion solver 109
A.3.3 Scene depth solver 110
A.4 Temporal integration over image sequences 110
A.4.1 Depth integration 111
A.4.2 Plane integration 111
A.4.3 Focal length recovery 112
A.4.4 Application to motion detection 112
A.5 Discussion 113

List of Figures

1.1 **Image measurements:** the feature points are overlaid on the image. 6

1.2 **Uncalibrated reconstruction process:** the reconstruction process is decoupled into two steps: projective reconstruction and Euclidean reconstruction. 7

2.1 **Orthographic projection:** Projection of feature point $\mathbf{p}_j$ represented in the world coordinate system C_{obj} to the image coordinates $(u_{ij}\ v_{ij})$. C_{cami} denotes the ith camera coordinate system. 13

2.2 **Projective reconstruction process** 17

2.3 **Normalization cases** 19

2.4 **Building sequence:** Focal lengths of the building sequence recovered by the uncalibrated reconstruction method. The recovered values are changing with the camera motion as expected. 26

2.5 **Building sequence input:** (a) 1st image, (b) 4th image, (c) 9th image of the building sequence. (d) 1st image of the building sequence with the feature points overlaid. . . . 27

2.6 **Building sequence results:** (a)Top and side view of the reconstruction, the 3-axis figures denote the recovered cameras. The top view shows that the recovered camera moves toward the building, then away again as expected. The side view shows that the recovered locations of the cameras are at the same height and the orientations are tilted upward. (b)Bottom and side view of the reconstructed building with texture mapping. . 28

2.7 **Grand Canyon sequence input:** (a) 1st image, (b) 46th image, (c) 91st image of the Grand Canyon sequence. (d) 1st image of the Grand Canyon sequence with the feature points overlaid. 30

2.8 **Grand Canyon sequence results:** (a)Top and side view of the reconstruction, the 3-axis figures denote the recovered cameras. (b)Top and side view of the reconstructed Grand Canyon with texture mapping. 31

2.9 **Grand Canyon sequence:** Focal lengths of the Grand Canyon sequence recovered by the uncalibrated reconstruction method. The recovered values are relatively constant as expected. 32

2.10 **Calibration setup results:** Top and side view of the reconstruction of the calibration setup, the points denote the recovered LED positions, the 3-axis figures are the recovered cameras. 32

2.11 **Calibration setup:** Differences of (a)the focal lengths (b) the principal points (u_0, v_0) (c) the aspect ratios of the calibration setup data recovered by the uncalibrated reconstruction method and by Tsai's calibration algorithm. 33

3.1 **Full rank case:** A scene with a three dimensional motion space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. 53

3.2 **Rank-5 case:** A scene with three motion trajectories which lie in a two dimensional space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. 54

3.3 **Toy sequence input:** (a) 1st image, (b) 7th image, (c) 18th image of the indoor sequence, the moving objects are circled in the 1st image. (d) 1st image of the indoor sequence with the feature points overlaid. 55

3.4 **Toy sequence results:** (a) Two views of the reconstruction with texture mapping, the black lines denote the recovered motion trajectories, the arrows show the motion directions. (b) Two views of the reconstruction with wireframe, the black lines denote the recovered motion trajectories. (c) Two views of the reconstruction, the 3-axis figures are the recovered cameras. 57

3.5 **Smith Hall sequence input:** (a) 1st image, (b) 33th image, (c) 80th image from the outdoor sequence, the moving objects are circled in the 1st image. (d) 1st image of the outdoor sequence with the feature points overlaid. 58

3.6 **Smith Hall sequence results:** (a) Two views of the reconstruction with texture mapping, the black lines denote the recovered motion trajectories, the arrows show the motion directions. (b) Two views of the reconstruction with wireframe, the black lines denote the recovered motion trajectories. 59

4.1 **Synthetic sequence:** Reconstruction of a scene with four moving objects by the uncalibrated multiple motion scene reconstruction method. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. 70

4.2 **Synthetic sequence:** Comparison of the focal lengths recovered by the uncalibrated multiple motion scene reconstruction method and their ground truth values for the synthetic sequence. The maximum error is 7.2% of the true value. 71

4.3 **Synthetic sequence :** Reconstruction of a scene with four moving objects by the multiple motion scene weak perspective method. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. The distortions are caused by the approximation of perspective cameras with weak perspective cameras. 71

4.4 **Real sequence input:** (a) 1st image, (b) 5th image, (c) 10th image of the indoor sequence. The white circles in the 1st image show the feature points selected on the moving objects. (d) 1st image of the sequence with the feature points overlaid. 73

4.5 **Real sequence results:** (a) Two views of the scene reconstruction with texture mapping, the black lines denote the recovered motion trajectories. (b) Two views of the scene reconstruction and the camera positions/orientations, the 3-axis figures are the recovered cameras. 74

4.6 **Real sequence:** Focal lengths of the real sequence recovered by the uncalibrated multiple motion scene reconstruction method. The recovered values change every two frames as expected. 75

5.1 **Gauge selection for orthographic projection:** Average shape errors recovered by the two formulations for orthographic reconstruction. It shows that the shape errors increase with the feature noise and the formulation which fixes the gauge at the beginning (Tomasi and Kanade's method) is more reliable. 90

5.2 **Gauge selection for uncalibrated perspective projection:** Average shape errors recovered by the two formulations for uncalibrated Euclidean reconstruction. It shows that the shape errors increase with the feature noise and the results from the two formulations are very close. 91

5.3 **Gauge selection for multiple motion scenes under orthographic projection:** Average shape errors recovered by the two formulations for multiple motion scenes orthographic reconstruction. It shows that the shape errors increase with the feature noise and the formulation which fixes the gauge at the beginning (the multiple motion scene orthographic reconstruction method presented in Chapter 3) is more reliable. 93

5.4 **Gauge selection for multiple motion scenes under uncalibrated perspective projection:** Average shape errors recovered by the two formulations for multiple motion scenes uncalibrated reconstruction. It shows that the shape errors increase with the feature noise and the results from the two formulations are very close. 93

A.1 **Robust homography and scene depth.** (a) 1st image, (b) 2nd image of the building sequence. (c) White dots denote the outliers of the robust estimation including the tops of the tall buildings and part of the ground. (d) Recovered depth map (darker denotes farther from the camera). 115
A.2 **Application to motion detection:** (a) 1st, 7th and 11th images of the bridge sequence. (b) 1st and 7th difference images between the registered images. White dots show the differences which are actually the outliers of the homographies. (c) 1st and 7th depth images, darker denotes farther. The depth image is improved through integration. (d) 1st and 7th difference images after the depth compensation. White dots show the differences which correspond to the moving objects while the differences due to the depth are cleaned up. 116
A.3 **3D mosaic for the building sequence.** This mosaic is built from 21 images. 117
A.4 **3D mosaic for the bridge sequence.** This mosaic is built from 14 images. 117

List of Tables

1.1 **Related work:** Reconstruction with uncalibrated cameras 4
1.2 **Related work:** Factorization methods 4
1.3 **Related work:** Multiple motion scene reconstruction 5

5.1 **Minimum data requirement:** Counting the arguments. n denotes the number of views and m denotes the number of feature points. 82
5.2 **Minimum data requirement:** Analyzing the solution. n denotes the number of views and m denotes the number of feature points. 84
5.3 **Minimum data requirement:** Empirical results. n denotes the number of views and m denotes the number of feature points. 87
5.4 **Gauge selection:** Comparison of two orthographic reconstruction processes for multiple motion scenes with and without gauge fixing at the first step of reconstruction. . . . 92

Chapter 1

Introduction

1.1 Problem definition

When a camera moves around in a scene, the images taken contain information about the scene structure, the camera motion and the camera intrinsic parameters. Structure from Motion (SFM), which is recovering camera motion and scene structure from image sequences, has various applications, such as scene modeling, robot navigation, object recognition and virtual reality. Most of previous research on SFM requires simplifying assumptions on the camera or the scene. Common assumptions are a) the camera intrinsic parameters, such as focal lengths, are known or unchanged throughout the sequence, and/or b) the scene does not contain moving objects. In practice, these are unrealistic assumptions. In this thesis we present a collection of reconstruction methods for dealing with image sequences taken with uncalibrated cameras and/or of scenes rich with independently moving objects. We refer to such scenes as multiple motion scenes.

The methods produce Euclidean reconstruction directly from feature point locations and are based on the bilinear relationship of camera motion and scene structure. For uncalibrated image sequences, we embed the camera intrinsic parameters within the camera motion representation. For image sequences of multiple motion scenes, we incorporate multiple motions into the scene structure representation. In this way, we derive linear and bilinear subspace constraints on the large amount of information integrated over the entire image sequences. By taking advantage of this redundant information we can achieve accurate and reliable reconstruction.

Firstly, we are interested in image sequences taken with uncalibrated cameras. Given tracked feature points under perspective projections, we simultaneously reconstruct the Euclidean shape, the camera motion and the camera intrinsic parameters assuming zero skews. The reconstruction process is decoupled into two steps: projective reconstruction and Euclidean reconstruction. The reconstruction steps are linear or bilinear depending on the number of unknown intrinsic parameters.

Secondly, we work on image sequences of multiple motion scenes taken from a moving airborne platform. In aerial video sequences the moving objects are often far from the camera. It is therefore difficult to get multiple feature points from every moving object. It is a good approximation to abstract the moving objects as points. As pointed out in [Avidan and Shashua, 2000], recovering the locations of the moving point from a monocular image sequence is impossible without assumptions about its trajectory. We assume that the objects are moving linearly with constant speeds. This assumption is reasonable for most moving objects, such as cars, planes and people, especially for short time intervals. Our goal is to recover the scene structure, the trajectories of the moving objects and the camera motion. The number of the moving objects is automatically detected without prior motion segmentation. The reconstruction method is built on linear subspace constraints.

Thirdly, we discuss the problem of multiple motion scene reconstruction taken with uncalibrated cameras. We assume that the cameras have zero skews and the objects are moving with constant velocities, therefore, we can combine the basic ideas behind the first two cases to get the linear and bilinear reconstruction methods which recover the scene structure, the motion trajectories of the objects, the camera motion together with the camera intrinsic parameters simultaneously.

1.2 Related work

Whether cameras are intrinsically pre-calibrated or **uncalibrated** differentiates various Structure from Motion methods. When nothing is known about the camera intrinsic parameters, the extrinsic parameters or the scene, it is only possible to compute a reconstruction up to an unknown projective transformation [Faugeras, 1992]. There has been considerable progress on projective reconstruction ([Faugeras, 1992, Mohr *et al.*, 1995, Triggs, 1995, Quan, 1995, Quan, 1996, Beardsley *et al.*, 1996, Carlsson and Weinshall, 1998]). Some methods use only two, three or four images to obtain a projective reconstruction by a linear least squares method [Hartley, 1997, Hartley, 1998]. On the other hand, some projective reconstruction methods take advantage of the large amount of information from image sequences [Shashua and Avidan, 1996, Sturm and Triggs, 1996, Triggs, 1996, Heyden, 1998]. Triggs proposed a projective factorization method in [Triggs, 1996] which recovered projective depths by estimating a set of fundamental matrices to chain all the images together. Sturm and Triggs [Sturm and Triggs, 1996] used epipoles and fundamental matrices estimated from the image points to get the scaled image measurements based on which a projective factorization is performed. Heyden [Heyden, 1997, Heyden, 1998] presented methods of using multilinear subspace constraints to perform projective structure from motion. Mahamud and Hebert [Mahamud and Hebert, 2000] proposed an iterative method which simultaneously recovered both the projective depths as well as the structure and motion. They determined the projective depths by solving a generalized eigenvalue problem and proved the monotonic convergence of the iterative scheme to a local maximum.

In order to obtain a Euclidean reconstruction from the projective reconstruction, some additional information about either the camera or the scene is needed . Hartley recovered the Euclidean shape using a global optimization technique assuming that the intrinsic parameters were constant [Hartley, 1994]. Heyden and Åström used a bundle adjustment algorithm to estimate the focal lengths, the principal points, the camera motion and the object shape together [Heyden and Astrom, 1997]. Triggs calibrated the cameras by recovering the absolute quadric which was computed by translating the constraints on the camera intrinsic parameters to the constraints on the absolute quadric [Triggs, 1997]. Pollefeys et al. assumed that the focal length was the only varying intrinsic parameter and presented a linear algorithm which was based on recovering the absolute conic [Pollefeys *et al.*, 1999]. Agapito et al. proposed a linear self-calibration algorithm for rotating and zooming cameras [Agapito *et al.*, 1999].

Assuming zero skews, we decouple the uncalibrated reconstruction process into two steps: projective reconstruction and Euclidean reconstruction. We present a projective factorization algorithm to compute the projective motion and shape based on the bilinear relationship of projective depths and affine reconstruction. This algorithm uniformly considers all the data in all the images. We then impose metric constraints on the projective reconstruction to recovery the Euclidean motion and shape as well as the camera intrinsic parameters based on linear and bilinear subspace constraints. Table 1.1 summarizes some of the related work in this area.

The linear and bilinear subspace reconstruction methods presented in this thesis use the **factorization** technique as the basis of solution. The factorization method, first developed by Tomasi and Kanade [Tomasi and Kanade, 1992] for orthographic views and extended by Poelman and Kanade [Poelman and Kanade, 1997] to weak and para perspective views, achieved its robustness and accuracy by applying the singular value decomposition (SVD) to a large number of images and feature points. Yu [Yu *et al.*, 1996] presented a new approach based on a higher-order approximation of perspective projection by using Taylor expansion of depth. The accuracy of the approximation depended on the order of Taylor expansion and the computation increased exponentially as the order increased. Christy and Horaud [Christy and Horaud, 1996a, Christy and Horaud, 1996b] described a method for perspective camera model by incrementally performing reconstructions with either a weak or a para perspective camera model. Recently, some work has been done to extend the factorization methods from feature-based methods to plane-based methods. Ma and Ahuja [Ma and Ahuja, 1998] recovered a dense shape, which is composed of the recovered plane positions and normals, from region correspondences by factorization. Sturm [Sturm, 2000] presented a factorization-based method to estimate poses of multiple planes. Table 1.2 lists some of the factorization methods. One major limitation with most factorization methods, however, is that they require the use of intrinsically calibrated cameras. In this thesis, we present uncalibrated reconstruction methods for both static scenes, which are the scenes without moving objects, and multiple motion scenes, which are the scenes containing multiple moving objects.

Projective Reconstruction					
Mohr et al. 1995	Triggs 1996	Hartley 1997 1998	Heyden 1997 1998	Mahamud and Hebert 2000	Han and Kanade 2000
nonlinear least squares solution	estimation of a set of fundamental matrices	linear least squares method on 2, 3 or 4 views	multilinear method based on constraints in shape space	iterative method recovering the projective depths and structure simultaneously	bilinear factorization algorithm for static and multiple motion scenes

Euclidean Reconstruction				
Hartley 1994	Heyden and Åström 1997	Pollefeys et al. 1998	Agapito et al. 1999	Han and Kanade 2000
global optimization (constant intrinsic parameters)	bundle adjustment (focal lengths and principal points)	linear algorithm (focal lengths only)	linear algorithm (rotating and zooming cameras)	linear and bilinear algorithm (all intrinsic parameters except skews)

Table 1.1: **Related work:** Reconstruction with uncalibrated cameras

Calibrated Cameras				*Uncalibrated Cameras*
Tomasi and Kanade 1991	Poelman and Kanade 1995	Yu et al. 1996	Christy and Horaud 1996	Han and Kanade 2000
orthographic cameras	weak and para perspective cameras	perspective cameras by Taylor expansion	perspective cameras by affine iteration	perspective cameras based on linear and bilinear constraints

Table 1.2: **Related work:** Factorization methods

Many interesting problems have been discussed on image sequences of **multiple motion scenes** including: scene reconstruction [Kumar *et al.*, 1994, Anandan *et al.*, 1994, Poelman and Kanade, 1997, Han and Kanade, 1998, Irani *et al.*, 1999], motion segmentation [Irani *et al.*, 1992, Torr and Murray, 1993, Sawhney *et al.*, 1999], reconstruction of motion trajectories [Avidan and Shashua, 2000], camera motion recovery [Irani *et al.*, 1997, Costeira and Kanade, 1998] and scene synthesis [Wexler and Shashua, 2000]. Most of these methods deal with the above problems separately. However, the temporal integration of information over sequences provides constraints on the scene reconstruction. We are therefore motivated to seek a one step reconstruction algorithm.

Zelnik-Manor and Irani [Zelnik-Manor and Irani, 1999, Irani, 1999] proposed using subspace constraints on multi-frame information to compute homography and optical flows. Their work demonstrated that the use of geometric constraints provided a good way to integrate information over sequences. The multibody factorization method proposed by Costeira and Kanade [Costeira and Kanade, 1998] reconstructed the motions and shapes of independently moving objects, but required that each object had multiple feature points. Avidan and Shashua [Avidan and Shashua, 2000] recovered the linear trajectory of a 3D point by line fitting. They assumed that the object was moving along a line, but they did not require that the object was moving with constant speed. They assumed the camera motion was given as well as the prior motion segmentation, and did not recover the scene structure. They ex-

Multiple Motion Scene Reconstruction					
Anandan et al. 1994 Kumar et al. 1994 Han and Kanade 1998	Irani et al. 1992 Torr and Murray 1993 Sawhney et al. 1999	Irani et al. 1997	Avidan and Shashua 1999 Shashua et al. 1999	Wexler and Shashua 2000	Han and Kanade 2000
Output: static scene structure	Output: motion segmentation	Output: camera motion recovery	Output: trajectories of moving objects	Output: scene synthesis	Output: scene structure, camera motion, trajectories of moving objects

Multiple Motion Scene Reconstruction based on Subspace Constraints				
Costeira and Kanade 1995	Zelnik-Manor and Irani 1999	Irani 1999	Bregler et al. 2000	Han and Kanade 2000
Output: scene structure and camera motion Requirement: multiple feature points on each object	Output: multiple homographies	Output: multiple frame optical flow	Output: non-rigid shape Requirement: 3D object represented by a basis of shapes	Output: scene structure, camera motion, trajectories of moving objects Requirement: tracked feature points

Table 1.3: **Related work:** Multiple motion scene reconstruction

tended this work to conic shape trajectories in [Shashua *et al.*, 1999]. Shashua and Wolf proposed the concept of *Homography Tensor* to represent three views of static and moving planar points in [Shashua and Wolf, 2000]. Bregler et al. [Bregler *et al.*, 2000] described a technique to recover non-rigid 3D model based on the representation of 3D shape as a linear combination of a set of basis shapes. The complexity of their solution increased with the number of basis shapes. Table 1.3 lists some of the related work to multiple motion scene reconstruction methods.

1.3 Thesis overview

In this thesis we present a collection of reconstruction methods dealing with image sequences taken with uncalibrated cameras and/or of multiple motion scenes. The input to the reconstruction methods are the tracked image measurements as shown in Figure 1.1. Each feature point is represented by $(u_{ij} \quad v_{ij})$ which is generated by the product of the camera projection P_i and the 3D feature point position $\mathbf{x}_{ij}$,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i \mathbf{x}_{ij} \quad \text{or} \quad \lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i \mathbf{x}_{ij} \tag{1.1}$$

Figure 1.1: **Image measurements:** the feature points are overlaid on the image.

where $i = 1 \cdots n$, n is the number of views and $j = 1 \cdots m$, m is the number of feature points, λ_{ij} is a non-zero scale factor, commonly called projective depth. The 3×4 projection matrix P_i is,

$$P_i \sim K_i \left[\begin{array}{cc} R_i & \mathbf{t}_i \end{array} \right] \tag{1.2}$$

The 3×3 matrix K_i encodes the intrinsic parameters of the ith camera. R_i is the ith rotation matrix and $\mathbf{t}_i$ is the ith translation vector. Therefore, P_i is a combination of the camera calibration K_i and the camera motion $[R_i \;\; \mathbf{t}_i]$. Since the 4×1 vector $\mathbf{x}_{ij}$ is the homogeneous representation of the feature position, we have,

$$\mathbf{x}_{ij} \sim \left[\begin{array}{c} \mathbf{p}_{ij} \\ 1 \end{array} \right] \tag{1.3}$$

When the scenes do not contain moving objects, $\mathbf{p}_{ij} = \mathbf{s}_j$ and $\mathbf{s}_j = [x_j \;\; y_j \;\; z_j]^{\mathrm{T}}$, that is, the feature positions are not dependent on when the images are taken. On the other hand, the feature positions $\mathbf{p}_{ij}$ are related to both of the feature number j and the image number i for multiple motion scenes.

Most research on SFM deals with the situations when the cameras are intrinsically calibrated, that is, all of K_i's are known, and/or the situations without moving objects, that is, $\mathbf{p}_{ij} = \mathbf{s}_j$. The uncalibrated reconstruction methods presented in the thesis work on image sequences with **unknown** matrices K_i, $i = 1 \cdots n$. The methods decouple the reconstruction process into two steps: projective reconstruction and Euclidean reconstruction. First, the projective reconstruction is performed to get the projective depths λ_{ij} from which the scaled image measurements are computed. According to Equation (1.1), factorization of the scaled measurements generates the motion and shape. However, the

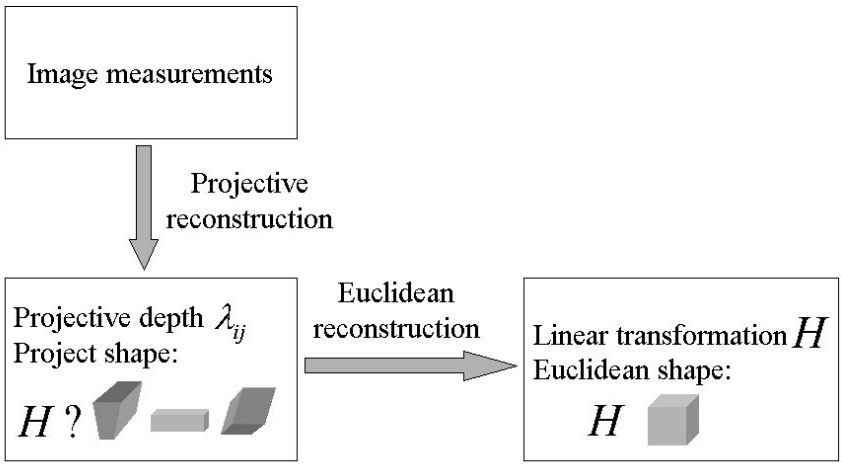


Figure 1.2: **Uncalibrated reconstruction process:** the reconstruction process is decoupled into two steps: projective reconstruction and Euclidean reconstruction.

factorization is not unique. It is up to a linear transformation H. The Euclidean reconstruction is then performed on the projection reconstruction to calculate the transformation H from which the Euclidean motion and shape as well as the camera intrinsic parameters are generated. The reconstruction process is summarized in Figure 1.2.

The multiple motion scene reconstruction methods presented in the thesis are based on a unified representation of the static scene and the moving objects. Assuming that the feature points are moving linearly with constant speeds, we regard every feature point as a moving point with constant velocity: the static points simply have zero velocity. Any point $\mathbf{p}_{ij}$ is represented by,

$$\mathbf{p}_{ij} = \mathbf{s}_j + i\mathbf{v}_j \tag{1.4}$$

in a world coordinate system, where $\mathbf{s}_j$ is the point position at frame 0 (i.e., when the 0th frame is taken) and $\mathbf{v}_j$ is its motion velocity. Based on this representation, we present the factorization-based reconstruction methods for multiple motion scenes.

We start Chapter 2 with a review of Tomasi and Kanade's factorization method [Tomasi and Kanade, 1992], then we describe the uncalibrated Euclidean reconstruction method which recovers motion and shape from multiple uncalibrated views. Given tracked feature points, this method recovers the camera motion, the scene structure and the camera intrinsic parameters (assuming zero skews).

We first present a bilinear factorization algorithm to get a projective reconstruction, then propose three normalization algorithms which impose metric constraints on the projective reconstruction with different assumptions about the intrinsic parameters. The normalization algorithms recover the unknown intrinsic parameters and convert the projective solution to a Euclidean one simultaneously. The first algorithm deals with the case that the focal lengths are the only unknown parameters. The second one deals with the case that the focal lengths and the principal point are unknown, while the principal point is fixed. These two algorithms are linear. The third algorithm, which is bilinear, works in the case that the focal lengths, the principal points and the aspect ratios are all unknown. We also describe the experimental results on real image sequences including building reconstruction, terrain recovery and multi-camera calibration.

Chapter 3 introduces the multiple motion scene reconstruction method with calibrated cameras. Assuming that the objects are moving linearly with constant speeds, we propose a unified representation of the static scene and the moving objects in which each point has an initial position and a constant velocity. Points on the static scene are defined to have zero velocity. This representation embeds the linear motion constraints within the scene structure, which naturally leads to a factorization-based method. The method reconstructs the scene structure, the trajectories of the moving objects and the camera motion simultaneously. The number of the moving objects is automatically detected without prior motion segmentation. We also discuss solutions to degenerate cases and extensions of the multiple motion scene reconstruction method to weak perspective projection and perspective projection. We apply this method to indoor and outdoor image sequences. The results are presented and discussed in this chapter.

Chapter 4 presents a factorization-based method for multiple motion scene reconstruction from uncalibrated views. The method decouples the reconstruction process into projective reconstruction and Euclidean reconstruction assuming that the objects are moving with constant velocities. Given tracked feature points without prior motion segmentation, the method recovers the scene structure, the trajectories of the moving objects, the camera motion together with the camera intrinsic parameters (assuming zero skews). The number of the moving objects is automatically detected. Experiments on synthetic and real images are described.

In Chapter 5 we discuss two important issues about reconstruction methods: minimum data requirement and gauge selection. We first describe the theoretical analysis about minimum number of views and features required by the subspace reconstruction methods presented in the thesis, then we give our empirical results. Gauge selection is the process of specifying the coordinate frame and representing the recovered geometry in the chosen frame. We analyze the gauge selection techniques used in the reconstruction methods described in this thesis and show that the techniques make the reconstruction methods reliable.

Chapter 6 summarizes the contributions of this thesis. We also identify the directions of extending

this research.

Appendix A proposes a method to recover scene depth and camera motion based on image homographies. It also discusses the applications of the method to motion detection and 3D mosaicking. This method takes advantage of the large amount of redundant information stored as the temporal consistency in video sequences to refine the reconstruction results. Different from the linear and bilinear subspace methods which are feature-based batch methods, the homography-based method directly recovers the dense scene structure in a sequential way. We include this work here to demonstrate that information integration over image sequences provides a reliable way for scene reconstruction.

Chapter 2

Euclidean Reconstruction with Uncalibrated Cameras

We first give a review of Tomasi and Kanade's factorization method which provides a technique basis for the reconstruction methods proposed in this thesis. Then we present a uncalibrated Euclidean reconstruction method [Han and Kanade, 2000a]. Unlike Tomasi and Kanade's method, it can work on image sequences taken with uncalibrated cameras. Given tracked feature points, the method recovers the scene structure, the camera extrinsic parameters and the intrinsic parameters simultaneously. Three normalization algorithms for Euclidean reconstruction are described, each of which handles different assumptions about the camera intrinsic parameters. The first algorithm deals with the case that the focal lengths are the only unknown parameters. The second one deals with the case that the focal lengths and the principal point are unknown, while the principal point is fixed. These two algorithms are linear. The third algorithm, which is bilinear, works in the case that the focal lengths, the principal points and the aspect ratios are all unknown. Synthetic experiments are conducted to evaluate the quality of the reconstruction method. Experimental results on real image sequences show the applications of the method to building reconstruction, terrain recovery and multi-camera calibration.

2.1 Review of Tomasi and Kanade's factorization method

The factorization method was first developed by Tomasi and Kanade [Tomasi and Kanade, 1992] for orthographic projections. The cameras are intrinsically calibrated. The core of the method is a process based on Singular Value Decomposition (SVD) which factors a matrix of image measurements into the product of the camera motion matrix and the scene structure matrix. The method does not need any prior assumptions about either the camera motion or the scene structure. In this section we give a brief review of the factorization method and introduce the geometry and notation used in this thesis.

2.1.1 Orthographic projection

Assuming a camera moves around in a scene, C_{obj} represents the world coordinate system attached to the scene and C_{cami} represents the camera coordinate system at different locations, where $i = 1 \cdots n$ and n is the number of frames. Suppose there are m feature points $\mathbf{p}_j$, $j = 1 \cdots m$, in the scene whose 3D locations are,

$$\mathbf{p}_j = \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^{\mathrm{T}} \tag{2.1}$$

which we want to recover by observing how they move in the projected image sequences. The position of $\mathbf{p}_j$ represented in the ith camera coordinate system is given by the transformation,

$$\mathbf{p}'_{ij} = R_i \mathbf{p}_j + \mathbf{t}_i. \tag{2.2}$$

where

$$R_i = \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \\ \mathbf{k}_i^{\mathrm{T}} \end{bmatrix} \qquad \mathbf{t}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix} \tag{2.3}$$

R_i is the ith rotation matrix whose rows $\mathbf{i}_i = [i_{xi} \;\; i_{yi} \;\; i_{zi}]^{\mathrm{T}}$, $\mathbf{j}_i = [j_{xi} \;\; j_{yi} \;\; j_{zi}]^{\mathrm{T}}$ and $\mathbf{k}_i = [k_{xi} \;\; k_{yi} \;\; k_{zi}]^{\mathrm{T}}$ are the axes of the camera coordinate system C_{cami} expressed in the world coordinate system. The vector $\mathbf{t}_i$ represents the position of the world coordinate system at the ith camera coordinate system. The representation (2.2) can be simplified using homogeneous coordinates,

$$\mathbf{x}'_{ij} = \begin{bmatrix} \mathbf{p}'_{ij} \\ 1 \end{bmatrix} = \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_j \\ 1 \end{bmatrix} \tag{2.4}$$

$$= \begin{bmatrix} R_i & \mathbf{t}_i \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \mathbf{x}_j \tag{2.5}$$

where

$$\mathbf{x}_j = \begin{bmatrix} \mathbf{p}_j \\ 1 \end{bmatrix} = \begin{bmatrix} x_j \\ y_j \\ z_j \\ 1 \end{bmatrix} \tag{2.6}$$

Under orthographic projection, the image coordinates of point $\mathbf{p}_j$ at the ith frame, denoted by $(u_{ij} \;\; v_{ij})$, are given by the first two elements of $\mathbf{x}'_{ij}$,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} = \begin{bmatrix} i_{xi} & i_{yi} & i_{zi} & t_{xi} \\ j_{xi} & j_{yi} & j_{zi} & t_{yi} \end{bmatrix} \mathbf{x}_j \tag{2.7}$$

The projection process is shown in Figure 2.1.

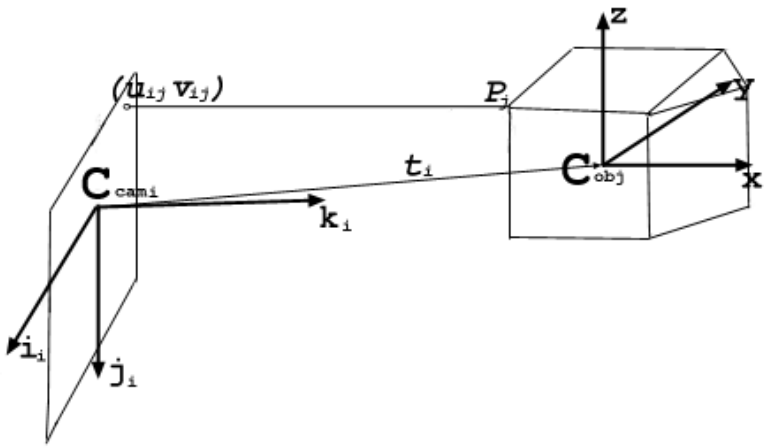


Figure 2.1: **Orthographic projection:** Projection of feature point $\mathbf{p}_j$ represented in the world coordinate system C_{obj} to the image coordinates $(u_{ij}\ v_{ij})$. C_{cami} denotes the ith camera coordinate system.

Imagine m feature points are tracked over n frames and all the image coordinates are put into a single $2n \times m$ matrix,

$$W = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1m} \\ v_{11} & v_{12} & \dots & v_{1m} \\ & \vdots & \vdots & \\ u_{n1} & u_{n2} & \dots & u_{nm} \\ v_{n1} & v_{n2} & \dots & v_{mn} \end{bmatrix} \tag{2.8}$$

Each row of W lists the image coordinates u or v of all the feature points in each frame, and each column represents the image trajectory of one feature point over the entire image sequence. The matrix W is called the *measurement matrix*. According to Equation (2.7),

$$W = MS + T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2.9}$$

with the rotation matrix M composed of the rotation axes,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \end{bmatrix}^{\mathrm{T}} \tag{2.10}$$

where

$$\mathbf{m}_{xi} = \mathbf{i}_i \quad \mathbf{m}_{yi} = \mathbf{j}_i \tag{2.11}$$

and the translation vector T,

$$T = \begin{bmatrix} t_{x1} & t_{y1} & t_{x2} & t_{y2} & \cdots & t_{xn} & t_{yn} \end{bmatrix}^{\mathrm{T}} \tag{2.12}$$

S is the shape matrix containing the feature points positions,

$$S = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_m \end{bmatrix} \tag{2.13}$$

2.1.2 World coordinate system location

Without loss of generality, we place the origin of the world coordinate system at the center of gravity of all the feature points, so that,

$$\sum_{j=1}^{m} \mathbf{p}_j = 0 \tag{2.14}$$

From Equation (2.7), we get,

$$\begin{aligned} \sum_{j=1}^{m} u_{ij} &= \sum_{j=1}^{m} (\mathbf{i}_i \cdot \mathbf{p}_j + t_{xi}) = \mathbf{i}_i \sum_{j=1}^{m} \mathbf{p}_j + m t_{xi} = m t_{xi} \\ \sum_{j=1}^{m} v_{ij} &= \sum_{j=1}^{m} (\mathbf{j}_i \cdot \mathbf{p}_j + t_{yi}) = \mathbf{j}_i \sum_{j=1}^{m} \mathbf{p}_j + m t_{yi} = m t_{yi} \end{aligned} \tag{2.15}$$

Therefore, the camera translation vector can be directly computed from Equation (2.15),

$$t_{xi} = \frac{1}{m} \sum_{j=1}^{m} u_{ij} \qquad t_{yi} = \frac{1}{m} \sum_{j=1}^{m} v_{ij} \tag{2.16}$$

2.1.3 Decomposition

The translation vector T is subtracted from W, leaving a "registered" measurement matrix $\hat{W}$,

$$\hat{W} = W - T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{2.17}$$

We have derived the relationship of W and pair of M and S by modeling the imaging process in Equation (2.9). The reconstruction problem is simplified by starting with the "registered" matrix $\hat{W}$ and obtaining a factorization into the motion matrix M and the shape matrix S. Since M and S can be at most rank 3, $\hat{W}$ will be at most rank 3. In real life situations the rank of W can be higher due to image noise. Singular Value Decomposition (SVD) is performed on $\hat{W}$ to get the best rank 3 approximation,

$$\hat{W} = U \Sigma V^{\mathrm{T}} \tag{2.18}$$

where matrix Σ is a diagonal matrix composed of the three biggest singular values which reveal the most important components in the data, $U_{2n\times 3}$ and $V_{m\times 3}$ are the left and right singular matrices, respectively.

Defining,

$$\begin{aligned} \hat{M} &= U\Sigma^{\frac{1}{2}} \\ \hat{S} &= \Sigma^{\frac{1}{2}}V^{\mathrm{T}} \end{aligned} \tag{2.19}$$

we have the two matrices whose product represents the "registered" measurement matrix $\hat{W} = \hat{M}\hat{S}$. However, this decomposition is not unique since for any non-singular 3×3 matrix A, $M = \hat{M}A$ and $S = A^{-1}\hat{S}$ are also a possible solution,

$$MS = (\hat{M}A)(A^{-1}\hat{S}) = \hat{M}\hat{S} = \hat{W} \tag{2.20}$$

In other words, the singular value decomposition (Equation (2.18)) provides a solution of motion and shape up to an affine transformation.

2.1.4 Normalization

The Euclidean solution can be obtained by finding the appropriate 3×3 matrix A. The correct A is determined using the fact that the rows of the motion matrix M represent the camera rotation axes. This process is called *normalization*.

Matrix A is constrained by orthogonality of the matrix M. Each row of $M = \hat{M}A$ is a unit norm vector and the rows are pairwisely perpendicular. This yields a set of constraints,

$$\begin{aligned} \hat{\mathbf{m}}_{xi}AA^{\mathrm{T}}\hat{\mathbf{m}}_{xi}^{\mathrm{T}} &= 1 \\ \hat{\mathbf{m}}_{yi}AA^{\mathrm{T}}\hat{\mathbf{m}}_{yi}^{\mathrm{T}} &= 1 \\ \hat{\mathbf{m}}_{xi}AA^{\mathrm{T}}\hat{\mathbf{m}}_{yi}^{\mathrm{T}} &= 0 \end{aligned} \tag{2.21}$$

where $i = 1 \cdots n$, $\hat{\mathbf{m}}_{xi}$ and $\hat{\mathbf{m}}_{yi}$ are the corresponding rows of the matrix $\hat{M}$. This is an over constrained system for the 6 elements of the symmetric matrix $Q = AA^{\mathrm{T}}$, which can be solved by linear least squares techniques. The transformation A is then computed from the matrix Q by rank 3 matrix decomposition. This decomposition is up to a three dimensional rotation because the matrix Q is symmetric. We can fix the rotation by aligning the world coordinate system with any orientation, such as the first camera orientation.

2.1.5 Motion and shape recovery

Once the matrix A is computed, the camera motion is recovered as,

$$M = \hat{M}A \tag{2.22}$$

and the scene structure as,

$$S = A^{-1}\hat{S} \tag{2.23}$$

2.2 Projective reconstruction

Tomasi and Kanade's factorization method requires the intrinsically calibrated cameras. Given feature correspondences from uncalibrated views, we cannot perform SVD directly on the measurement matrix W as in Tomasi and Kanade's method because the perspective projection is involved. We decouple the uncalibrated reconstruction process into projective reconstruction and Euclidean reconstruction. In this section we describe the bilinear projective reconstruction algorithm.

Suppose there are n perspective cameras: P_i, $i = 1 \cdots n$ and m feature points $\mathbf{x}_j$, $j = 1 \cdots m$ represented by homogeneous coordinates. The image coordinates are represented by $(u_{ij} \;\; v_{ij})$. Using the symbol $\sim$ to denote equality up to a scale, the following hold,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i\mathbf{x}_j \quad \text{or} \quad \lambda_{ij}\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i\mathbf{x}_j \tag{2.24}$$

where λ_{ij} is a non-zero scale factor, commonly called projective depth. Each P_i is a 3×4 matrix and each feature point $\mathbf{x}_j$ is a 4×1 vector. The equivalent matrix form is,

$$W_s = \begin{bmatrix} \lambda_{11}\begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1m}\begin{bmatrix} u_{1m} \\ v_{1m} \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_{n1}\begin{bmatrix} u_{n1} \\ v_{n1} \\ 1 \end{bmatrix} & \cdots & \lambda_{nm}\begin{bmatrix} u_{nm} \\ v_{nm} \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} P_1 \\ \vdots \\ P_n \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_m \end{bmatrix} = PX \tag{2.25}$$

where W_s is a $3n \times m$ matrix, called *scaled measurement matrix*. Compared with the measurement matrix W which is $2n \times m$ in Tomasi and Kanade's method, the scaled measurement matrix W_s encodes the projected image information in W and the projective depths. Since each P_i is a 3×4 matrix, W_s is

at most rank 4. We therefore apply the following projective factorization algorithm which is similar to Triggs's bilinear approach [Triggs, 1995]. The algorithm iteratively applies rank 4 factorization to the current scaled measurement matrix.

Bilinear Projective Factorization Algorithm

1. Set $\lambda_{ij} = 1$, for $i = 1 \cdots n$ and $j = 1 \cdots m$;
2. Compute the current scaled measurement matrix W_s by Equation (2.25);
3. Perform rank 4 factorization on W_s, generate the projective motion and shape;
4. Reset $\lambda_{ij} = P_i^{(3)}\mathbf{x}_j$, where $P_i^{(3)}$ denotes the third row of the projection matrix P_i;
5. If λ_{ij}'s are the same as the previous iteration, stop; else go to step 2.

The goal of the projective reconstruction process is to estimate the values of the projective depths (λ_{ij}'s) which make Equation (2.25) consistent. Figure 2.2 shows the reconstruction process. The reconstruction results are iteratively improved by back projecting the current projective reconstruction to refine the depth estimates.

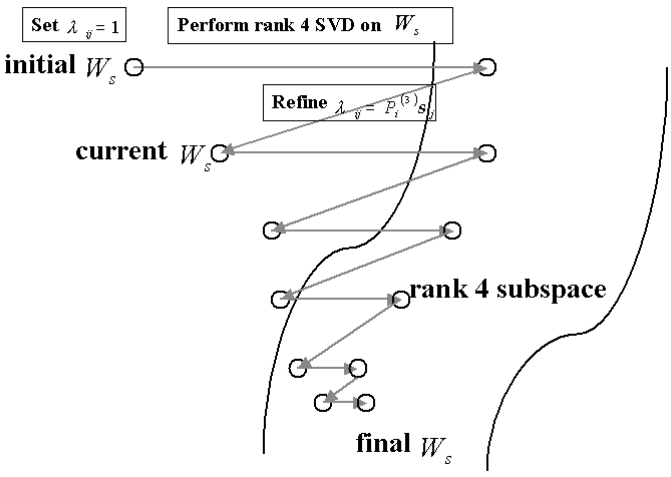


Figure 2.2: **Projective reconstruction process**

2.3 Euclidean reconstruction

The factorization of Equation (2.25) recovers the motion and shape up to a 4×4 linear projective transformation H,

$$W_s = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = PX \tag{2.26}$$

where $P = \hat{P}H$ and $X = H^{-1}\hat{X}$. $\hat{P}$ and $\hat{X}$ are referred to as the projective motion and the projective shape. Any non-singular 4×4 matrix could be inserted between $\hat{P}$ and $\hat{X}$ to get another motion and shape pair.

Let us assume zero skews. We impose metric constraints to the projective motion and shape in order to simultaneously reconstruct the intrinsic parameters (i.e., the focal lengths, the principal points and the aspect ratios) and the linear transformation H, from which we can get the Euclidean motion and shape. We call this process *normalization*. We classify the situations into three cases as shown in Figure 2.3:

Case 1: Only the focal lengths are unknown.

This case includes the situations that the camera undergoes zooming in/out during the sequence. The focal lengths are therefore the main concerns of the reconstruction process.

Case 2: The focal lengths and the principal point are unknown, and the principal point is fixed.

In this case we are interested in the situations in which the camera focal length changes only a little, so that there is no obvious zooming effect and the principal point is very close to being constant. Aerial image sequences taken by a flying platform are examples of this case.

Case 3: The focal lengths, the principal points and the aspect ratios are all unknown and varying.

This case covers the situations that multiple cameras are included. The focal lengths, the principal points and the aspect ratios all vary from image to image.

We present three factorization-based normalization algorithms to deal with these three cases respectively. The algorithms are linear for the first two cases and bilinear for the third case.

2.3.1 Normalization algorithm outline

The projection matrix P_i is,

$$P_i \sim K_i \begin{bmatrix} R_i & \mathbf{t}_i \end{bmatrix} \tag{2.27}$$

where

$$K_i = \begin{bmatrix} f_i & 0 & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{bmatrix} \quad R_i = \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \\ \mathbf{k}_i^{\mathrm{T}} \end{bmatrix} \quad \mathbf{t}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}$$

Assumption / Case	Focal lengths	Principal points	Aspect ratios
Case 1	unknown varying	known	known
Case 2	unknown varying	unknown fixed	known
Case 3	unknown varying	unknown varying	unknown varying

Figure 2.3: **Normalization cases**

The upper triangular calibration matrix K_i encodes the intrinsic parameters of the ith camera: f_i represents the focal length, $(u_{0i} \;\; v_{0i})$ is the principal point and α_i is the aspect ratio. R_i is the ith rotation matrix with $\mathbf{i}_i, \mathbf{j}_i$ and $\mathbf{k}_i$ denoting the rotation axes. $\mathbf{t}_i$ is the ith translation vector. Combining Equation (2.27) for $i = 1 \cdots n$ into one matrix equation, we get,

$$P = \begin{bmatrix} M & T \end{bmatrix} \tag{2.28}$$

where

$$\begin{aligned} M &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \end{bmatrix}^{\mathrm{T}} \\ T &= \begin{bmatrix} T_{x1} & T_{y1} & T_{z1} & \cdots & T_{xn} & T_{yn} & T_{zn} \end{bmatrix}^{\mathrm{T}} \end{aligned}$$

and

$$\begin{aligned} \mathbf{m}_{xi} &= \mu_i f_i \mathbf{i}_i + \mu_i u_{0i} \mathbf{k}_i \quad & \mathbf{m}_{yi} &= \mu_i \alpha_i f_i \mathbf{j}_i + \mu_i v_{0i} \mathbf{k}_i \quad & \mathbf{m}_{zi} &= \mu_i \mathbf{k}_i \\ T_{xi} &= \mu_i f_i t_{xi} + \mu_i u_{0i} t_{zi} \quad & T_{yi} &= \mu_i \alpha_i f_i t_{yi} + \mu_i v_{0i} t_{zi} \quad & T_{zi} &= \mu_i t_{zi} \end{aligned} \tag{2.29}$$

The shape matrix is represented by,

$$X \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{2.30}$$

where

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \end{bmatrix}$$

and

$$\begin{aligned} \mathbf{s}_j &= \begin{bmatrix} x_j & y_j & z_j \end{bmatrix}^{\mathrm{T}} \\ \mathbf{x}_j &= \begin{bmatrix} \nu_j \mathbf{s}_j^{\mathrm{T}} & \nu_j \end{bmatrix}^{\mathrm{T}} \end{aligned}$$

μ_i and ν_j are the scale factors of the homogeneous representations in Equations (2.27) and (2.30).

World coordinate system location

We place the origin of the world coordinate system at the center of gravity of all the scaled feature points to enforce,

$$\sum_{j=1}^{m} \nu_j \mathbf{s}_j = 0 \tag{2.31}$$

we get,

$$\sum_{j=1}^{m} \lambda_{ij} u_{ij} = \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + \nu_j T_{xi}) = \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j \mathbf{s}_j + T_{xi} \sum_{j=1}^{m} \nu_j = T_{xi} \sum_{j=1}^{m} \nu_j \tag{2.32}$$

Similarly,

$$\sum_{j=1}^{m} \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^{m} \nu_j \qquad \sum_{j=1}^{m} \lambda_{ij} = T_{zi} \sum_{j=1}^{m} \nu_j \tag{2.33}$$

Define the 4×4 projective transformation H as,

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{2.34}$$

where A is 4×3 and B is 4×1.
Since $P = \hat{P} H$,

$$\begin{bmatrix} M & T \end{bmatrix} = \hat{P} \begin{bmatrix} A & B \end{bmatrix} \tag{2.35}$$

we have,

$$T_{xi} = \hat{P}_{xi} B \qquad T_{yi} = \hat{P}_{yi} B \qquad T_{zi} = \hat{P}_{zi} B \tag{2.36}$$

From Equations (2.32) and (2.33), we know,

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} u_{ij}}{\sum_{j=1}^{m} \lambda_j} \qquad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} v_{ij}}{\sum_{j=1}^{m} \lambda_j} \tag{2.37}$$

We set up $2n$ linear equations of the 4 unknown elements of the matrix B. Linear least squares solutions are then computed.

Normalization

As $\mathbf{m}_{xi}$, $\mathbf{m}_{yi}$ and $\mathbf{m}_{zi}$ are the sum of the scaled rotation axes, we get the following constraints from Equation (2.29),

$$\begin{aligned} |\mathbf{m}_{xi}|^2 &= \mu_i^2 f_i^2 + \mu_i^2 u_{0i}^2 \\ |\mathbf{m}_{yi}|^2 &= \mu_i^2 \alpha_i^2 f_i^2 + \mu_i^2 v_{0i}^2 \\ |\mathbf{m}_{zi}|^2 &= \mu_i^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} &= \mu_i^2 u_{0i} v_{0i} \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} &= \mu_i^2 u_{0i} \\ \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} &= \mu_i^2 v_{0i} \end{aligned} \tag{2.38}$$

Based on the three different assumptions of the intrinsic parameters (three cases), we translate the above constraints to linear or bilinear constraints on MM^{T} (see Section 2.3.2, 2.3.3 and 2.3.4 for details). According to Equation (2.35),

$$M = \hat{P}A \tag{2.39}$$

therefore,

$$MM^{\mathrm{T}} = \hat{P}AA^{\mathrm{T}}\hat{P}^{\mathrm{T}} \tag{2.40}$$

Define

$$Q = AA^{\mathrm{T}} \tag{2.41}$$

we can translate the constraints on MM^{T} to the constraints on the 10 unknown elements of the symmetric 4×4 matrix Q. Least squares solutions are computed. Then we get the matrix A from Q by rank 3 matrix decomposition.

Motion and shape recovery

Once the matrices A and B have been found, the projective transformation is $H = [A \ B]$. The shape is computed as $X = H^{-1}\hat{X}$ and the motion matrix as $P = \hat{P}H$. We first compute the scales μ_i,

$$\mu_i = |\mathbf{m}_{zi}| \tag{2.42}$$

We then compute the principal points (if applied),

$$u_{0i} = \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{zi}}{\mu_i^2} \qquad v_{0i} = \frac{\mathbf{m}_{yi} \cdot \mathbf{m}_{zi}}{\mu_i^2} \tag{2.43}$$

and the focal lengths as,

$$f_i = \frac{\sqrt{|\mathbf{m}_{xi}|^2 - \mu_i^2 u_{0i}^2}}{\mu_i} \tag{2.44}$$

The aspect ratios (if applied) are,

$$\alpha_i = \frac{\sqrt{|\mathbf{m}_{yi}|^2 - \mu_i^2 v_{0i}^2}}{\mu_i f_i} \tag{2.45}$$

Therefore, the motion parameters are,

$$\begin{aligned} \mathbf{k}_i &= \frac{\mathbf{m}_{zi}}{\mu_i} \quad & \mathbf{i}_i &= \frac{\mathbf{m}_{xi} - \mu_i u_{0i}\mathbf{k}_i}{\mu_i f_i} \quad & \mathbf{j}_i &= \frac{\mathbf{m}_{yi} - \mu_i v_{0i}\mathbf{k}_i}{\mu_i \alpha_i f_i} \\ t_{zi} &= \frac{T_{zi}}{\mu_i} \quad & t_{xi} &= \frac{T_{xi} - \mu_i u_{0i} t_{zi}}{\mu_i f_i} \quad & t_{yi} &= \frac{T_{yi} - \mu_i v_{0i} t_{zi}}{\mu_i \alpha_i f_i} \end{aligned} \tag{2.46}$$

Algorithm outline

The normalization process is summarized by the following algorithm.

Normalization Algorithm

1. Perform SVD on W_s and get the projective motion $\hat{P}$ and the projective shape $\hat{X}$;
2. Sum up each row of W_s and compute the ratios between them as in Equation (2.37);
3. Set up $2n$ linear equations of the 4 unknown elements of the matrix B based on the ratios and compute B;
4. Set up linear equations of the 10 unknown elements of the symmetric matrix Q and get Q;
5. Perform rank 3 matrix decomposition on Q to get A from $Q = AA^{\mathrm{T}}$;
6. Put matrices A and B together and get the projective transformation $H = [A \;\; B]$;
7. Recover the shape using $X = H^{-1}\hat{X}$ and the motion matrix using $P = \hat{P}H$;
8. Recover the intrinsic parameters, the rotation axes and the translation vector according to Equations (2.43)–(2.46).

2.3.2 Case 1: Unknown focal lengths

Assume that the focal lengths are the only unknown intrinsic parameters. Then we have,

$$u_{0i} = 0 \quad v_{0i} = 0 \quad \alpha_i = 1 \tag{2.47}$$

We combine the constraints in Equation (2.38) to impose the following linear constraints on the unknown elements of the matrix Q [Han and Kanade, 1999b],

$$\begin{aligned} |\mathbf{m}_{xi}|^2 &= |\mathbf{m}_{yi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} &= 0 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} &= 0 \\ \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} &= 0 \end{aligned}$$

We can add one more equation assuming $\mu_1 = 1$,

$$|\mathbf{m}_{z1}|^2 = 1 \tag{2.48}$$

Totally we have $4n + 1$ linear equations of the 10 unknown elements of Q.

The only intrinsic parameters to be recovered in this case are the focal lengths. As the aspect ratios are 1, the focal lengths are computed by the average of Equations (2.44) and (2.45),

$$f_i = \frac{|\mathbf{m}_{xi}| + |\mathbf{m}_{yi}|}{2\mu_i} \tag{2.49}$$

2.3.3 Case 2: Unknown focal lengths and constant principal point

In case 2, we assume that the focal lengths are unknown and the principal point is constant. Then,

$$u_{0i} = u_0 \quad v_{0i} = v_0 \quad \alpha_i = 1 \tag{2.50}$$

We translate the constraints in Equation (2.38) to the following constraints [Han and Kanade, 2000c],

$$\begin{aligned} \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{yi}}{\mathbf{m}_{xi} \cdot \mathbf{m}_{zi}} &= \frac{\mathbf{m}_{yi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{zi} \cdot \mathbf{m}_{zi}} \\ (|\mathbf{m}_{xi}|^2 - |\mathbf{m}_{yi}|^2)(\mathbf{m}_{zi} \cdot \mathbf{m}_{zi}) &= (\mathbf{m}_{xi} \cdot \mathbf{m}_{zi})^2 - (\mathbf{m}_{yi} \cdot \mathbf{m}_{zi})^2 \end{aligned} \tag{2.51}$$

and

$$\begin{aligned} \frac{\mathbf{m}_{zi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{zj} \cdot \mathbf{m}_{zj}} &= \frac{|\mathbf{m}_{xi}|^2 - |\mathbf{m}_{yi}|^2}{|\mathbf{m}_{xj}|^2 - |\mathbf{m}_{yj}|^2} \\ \frac{|\mathbf{m}_{xi}|^2 - |\mathbf{m}_{yi}|^2}{|\mathbf{m}_{xj}|^2 - |\mathbf{m}_{yj}|^2} &= \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{yi}}{\mathbf{m}_{xj} \cdot \mathbf{m}_{yj}} \\ \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{yi}}{\mathbf{m}_{xj} \cdot \mathbf{m}_{yj}} &= \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{xj} \cdot \mathbf{m}_{zj}} \\ \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{xj} \cdot \mathbf{m}_{zj}} &= \frac{\mathbf{m}_{yi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{yj} \cdot \mathbf{m}_{zj}} \end{aligned}$$

$$\frac{\mathbf{m}_{yi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{yj} \cdot \mathbf{m}_{zj}} = \frac{\mathbf{m}_{zi} \cdot \mathbf{m}_{zi}}{\mathbf{m}_{zj} \cdot \mathbf{m}_{zj}} \tag{2.52}$$

where $j = i + 1$, if $i \neq n$; $j = 1$, if $i = n$. We also have the following constraint assuming $\mu_1 = 1$,

$$|\mathbf{m}_{z1}|^4 = 1 \tag{2.53}$$

The above constraints can be represented as linear equations of the unknown elements of the symmetric matrix $Q' = \mathbf{q}\mathbf{q}^{\mathrm{T}}$, where $\mathbf{q}$ is a 10×1 vector composed of the 10 unknown elements of the matrix Q. In total, we can have $13n + 1$ linear equations of the 55 unknown elements of the matrix Q'.

Once Q' has been computed, $\mathbf{q}$ is generated by rank 1 matrix decomposition of Q'. We then put the 10 elements of $\mathbf{q}$ into a symmetric 4×4 matrix Q which is factored as AA^{T}.

We compute the principal point as the average of Equation (2.43),

$$\begin{aligned} u_0 &= \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbf{m}_{xi} \cdot \mathbf{m}_{zi}}{\mu_i^2} \\ v_0 &= \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbf{m}_{yi} \cdot \mathbf{m}_{zi}}{\mu_i^2} \end{aligned} \tag{2.54}$$

and the focal lengths as the average of Equations (2.44) and (2.45),

$$f_i = \frac{\sqrt{|\mathbf{m}_{xi}|^2 - \mu_i^2 u_0^2} + \sqrt{|\mathbf{m}_{yi}|^2 - \mu_i^2 v_0^2}}{2\mu_i} \tag{2.55}$$

2.3.4 Case 3: Unknown focal lengths, principal points and aspect ratios

Case 3 includes the situations that the focal lengths, the principal points and the aspect ratios are all unknown and varying. We then represent the constraints in Equation (2.38) as bilinear equations on the focal lengths and the principal points plus the aspect ratios. Starting with the rough values of the principal points and the aspect ratio of the first camera (α_1), we impose linear constraints on the unknown elements of the matrix Q [Han and Kanade, 2000c],

$$\begin{aligned} \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} &= u_{0i} v_{0i}\, \mathbf{m}_{zi} \cdot \mathbf{m}_{zi} \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} &= u_{0i}\, \mathbf{m}_{zi} \cdot \mathbf{m}_{zi} \\ \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} &= v_{0i}\, \mathbf{m}_{zi} \cdot \mathbf{m}_{zi} \end{aligned} \tag{2.56}$$

We add two more equations assuming $\mu_1 = 1$,

$$\alpha_1^2 (|\mathbf{m}_{x1}|^2 - u_{01}^2) = |\mathbf{m}_{y1}|^2 - v_{01}^2$$

$$|\mathbf{m}_{z1}|^2 = 1 \tag{2.57}$$

Once the matrix H has been found, the current shape is $X = H^{-1}\hat{X}$ and the current motion matrix is $P = \hat{P}H$. We compute the refined principal points, the currently recovered focal lengths and the refined aspect ratios according to Equations (2.43), (2.44) and (2.45) respectively. The current motion parameters are then computed as in Equation (2.46).

Taking the refined principal points and the first aspect ratio, the normalization steps are performed again to generate the matrix H, then the focal lengths, the current shape and motion, the refined principal points and aspect ratios. The above steps are repeated until the principal points and the first aspect ratio do not change.

The normalization process is computationally equivalent to recovering the absolute quadric which is computed by translating the constraints on the intrinsic camera parameters to the constraints on the absolute quadric [Triggs, 1997, Pollefeys *et al.*, 1999]. Our representation is explicit in the motion parameters (rotation axes and translation vectors) and enables the geometric constraints to be naturally enforced. The representation also deals with the similarity ambiguity problem directly by putting the world coordinate system at the center of gravity of the object and aligning its orientation with the first camera. Compared with the method presented by Pollefeys et al. in [Pollefeys *et al.*, 1999], the normalization algorithm described in Section 2.3.2 is based on the same constraints as their method, but our framework enables natural extensions to the reconstruction of other intrinsic parameters (normalization algorithms of Section 2.3.3 and 2.3.4) while they used nonlinear bundle adjustment.

2.4 Experiments

In this section we demonstrate experimental results of the uncalibrated Euclidean reconstruction method. Given tracked feature points, we first generate the projective reconstruction as described in Section 2.2, then recover the Euclidean reconstruction and the camera intrinsic parameters using one of the three normalization algorithms described in Section 2.3 . First, synthetic experiments are conducted to evaluate the quality of the reconstruction method. Then, results for real image sequences corresponding to each of the three cases are shown as well. Experimental results on synthetic and real data show that this method is reliable under noise.

2.4.1 Synthetic examples

We synthesize 50 sequences of 20 frames with 8 feature points representing a cube in the scene. The camera undergoes non-critical random motions. The distance between the camera and the cube is between 4 to 15 times the cube size. The camera rotation is through 30 to 65 degrees around the cube.

We add 2 pixels standard noise to the feature locations. The image size is 640×480. The experimental results show that the method converges reliably. The errors of the recovered feature points positions are less than 0.8% of the object size. The recovered focal lengths are always within $1 \pm 1.8\%$ of the true values. The errors of the principal points are less than 0.25% of the image size and the errors of the aspect ratios are less than 0.5% of the true values. The maximum distance between the recovered camera locations and the corresponding ground truth values is 2.4% of the object size and the maximum difference between the recovered camera orientations and the true values is 0.33°.

2.4.2 Real example 1: Building sequence

The building sequence was taken by a hand-held camera in front of a building. The camera was very far from the building at first, then moved toward the building, and away again. The camera was zoomed in when it was far from the building and zoomed out when it was close so that the building appeared to be almost the same size in every image of the sequence. The longest focal length was about 3 times the shortest one according to the rough readings on the camera. The sequence includes 14 frames, of which three are shown in Figure 2.5(a)-(c). 50 feature points were manually selected along the building windows and the corners as shown in Figure 2.5(d). In this example we assume the focal lengths are unknown while the principal points are given (the middle of the images) and the aspect ratios are 1. We apply the projective algorithm described in Section 2.2 and the normalization algorithm described in Section 2.3.2 to this example.

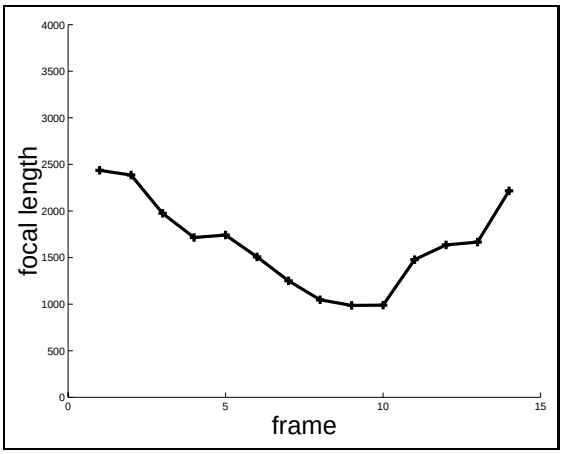
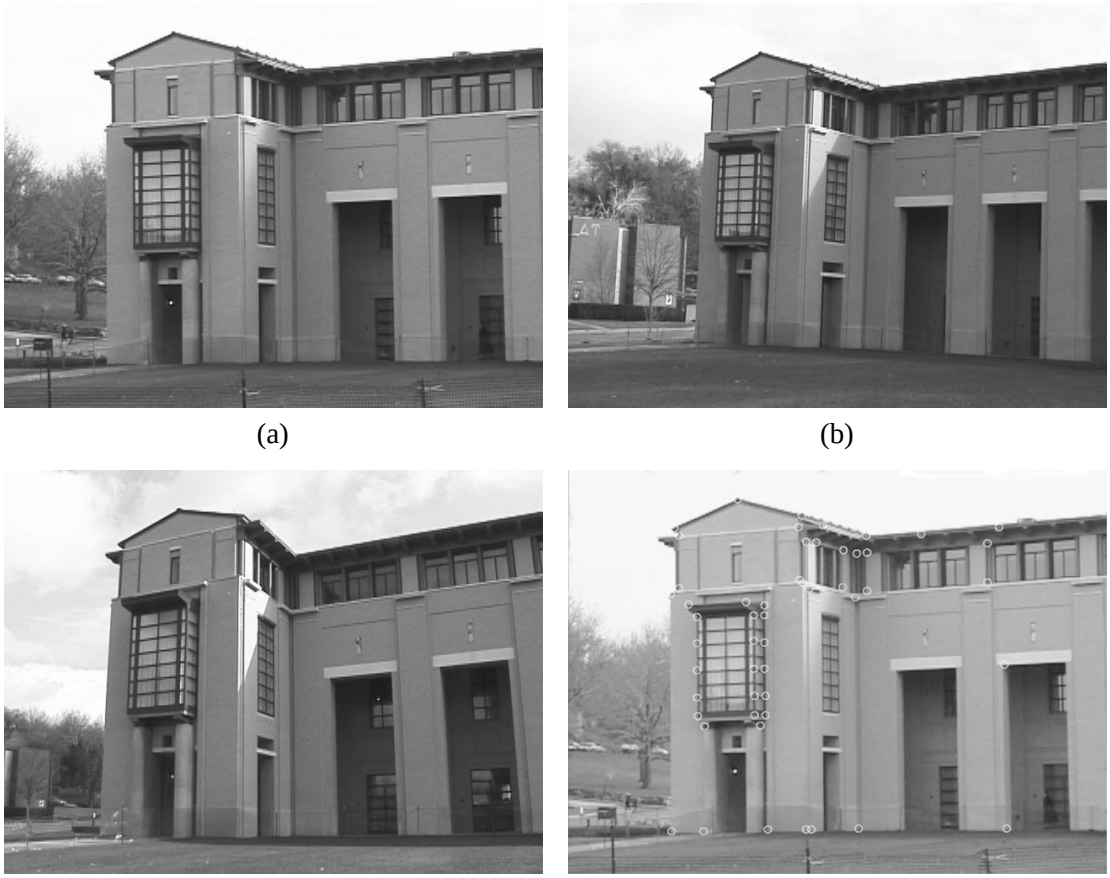


Figure 2.4: **Building sequence:** Focal lengths of the building sequence recovered by the uncalibrated reconstruction method. The recovered values are changing with the camera motion as expected.

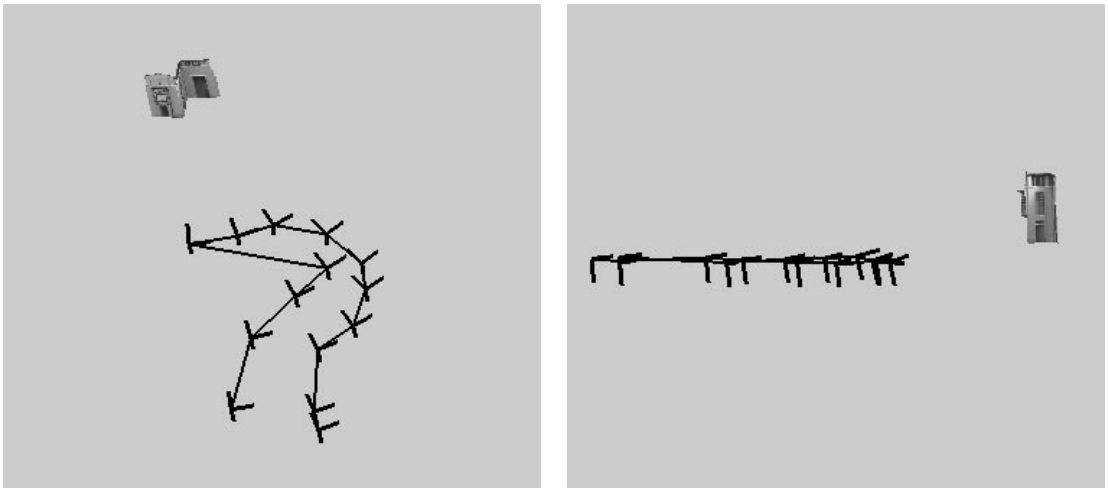
Figure 2.6(a) shows the reconstructed building and camera trajectories. The top view shows that the recovered camera moves toward the building and then away again as expected. The recovered camera positions and orientations shown in the side view demonstrate that all the cameras have the almost



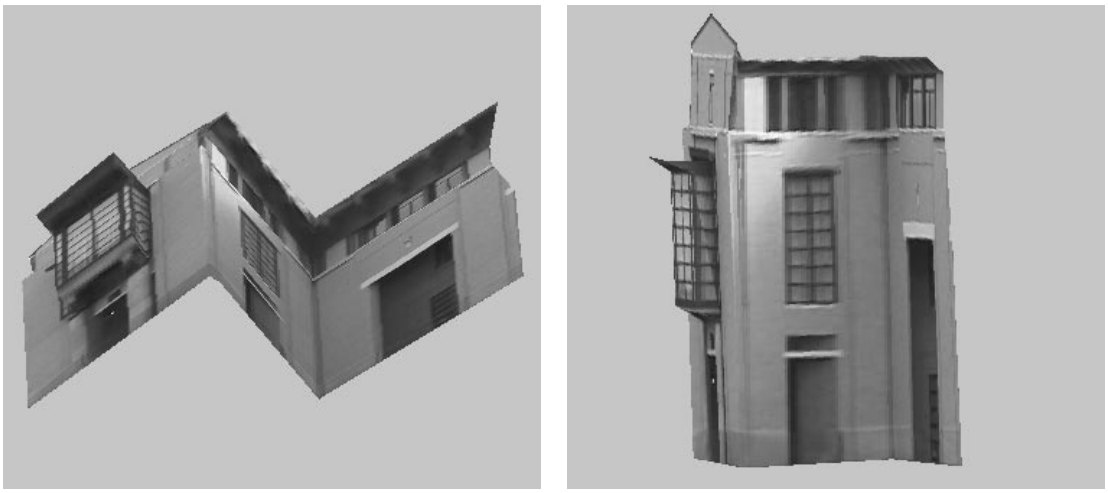
(a) (b)

(c) (d)

Figure 2.5: **Building sequence input:** (a) 1st image, (b) 4th image, (c) 9th image of the building sequence. (d) 1st image of the building sequence with the feature points overlaid.



(a)



(b)

Figure 2.6: **Building sequence results:** (a)Top and side view of the reconstruction, the 3-axis figures denote the recovered cameras. The top view shows that the recovered camera moves toward the building, then away again as expected. The side view shows that the recovered locations of the cameras are at the same height and the orientations are tilted upward. (b)Bottom and side view of the reconstructed building with texture mapping.

same height and tilt upward a little bit, which are the expected values that the same person took the sequence while walking in front of the building. Figure 2.6(b) shows the reconstructed building with texture mapping. To quantify the results, we measure the orthogonality and parallelism of the lines composed of the recovered feature points. The average angle between pairs of expected parallel lines is 0.89° and the average angle between pairs of expected perpendicular lines is 91.74°. Figure 2.4 plots the recovered focal lengths, which shows that the focal lengths are changing with the camera motion as we expected.

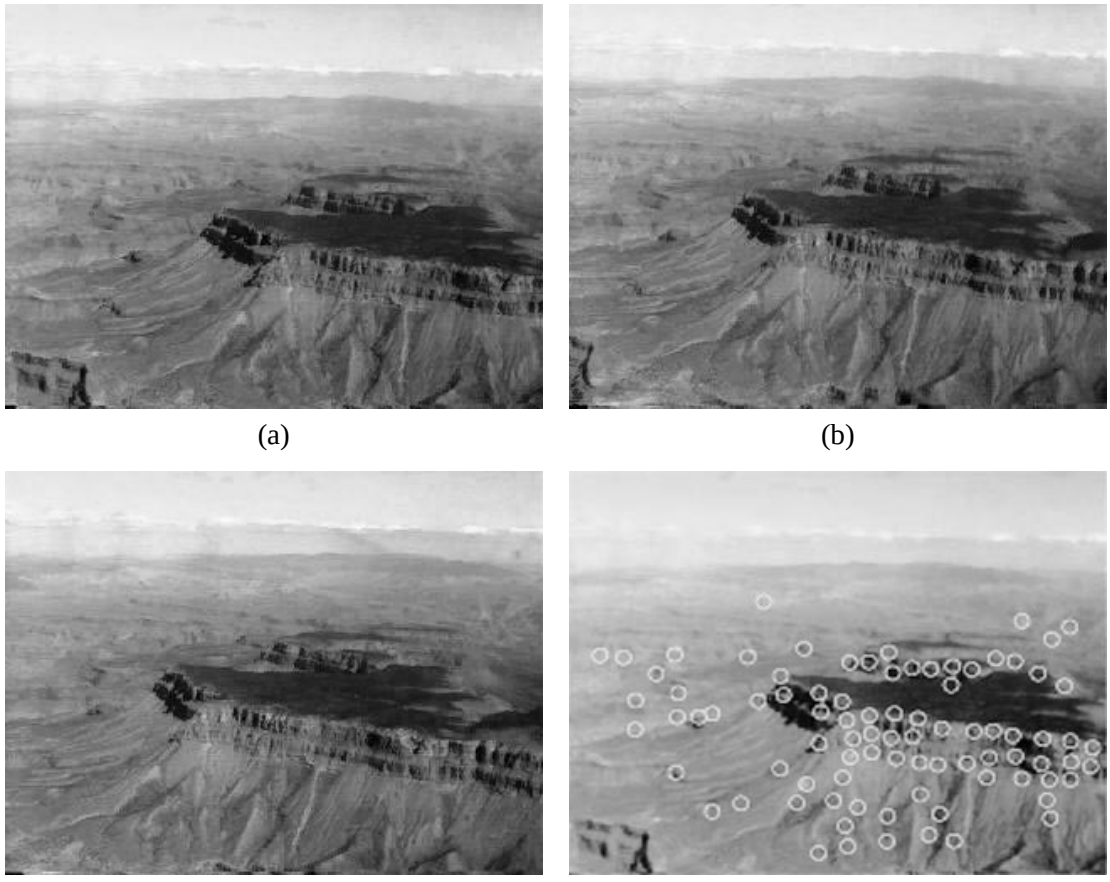
2.4.3 Real example 2: Grand Canyon sequence

The second example is an aerial image sequence taken from a small airplane flying over the Grand Canyon. The plane changed its altitude as well as its roll, pitch and yaw angles during the sequence. The sequence consists of 97 images, and 86 feature points were tracked through the sequence. Three frames from the sequence are shown in Figure 2.7(a)-(c), and the tracked feature points are shown in Figure 2.7(d). We assume that the focal lengths and the principal point are unknown, but that the principal point is fixed over the sequence. The normalization algorithm of Section 2.3.3 is used here. Figures 2.8(a) and (b) show the reconstructed camera trajectories and terrain map. The camera focal lengths changed little when taking the sequence. Figure 2.9 is a plot of the recovered focal lengths, and shows that the focal lengths are relatively constant. The principal point recovered by the reconstruction method is $(159, 119)$ (with the image size of 320×240).

2.4.4 Real example 3: Calibration setup

In this experiment we test our method on a setup for multi-camera calibration. In this setup 51 cameras are placed in a dome, and a bar of LEDs is moved around under the dome. The bar is imaged by each camera as it is moved through a series of known positions. Since the intrinsic parameters of each camera do not change as the bar is moved, the images taken by one camera are combined into one image containing multiple bars. This composite image includes 232 feature points (LED positions). Therefore, the setup generates 51 images, each contains 232 features, which are to be used as calibration data for the cameras. Tsai's calibration algorithm [Tsai, 1987] is used on this setup to calibrate the 51 cameras. The calibration results of Tsai's algorithm are compared with the results of the uncalibrated reconstruction method.

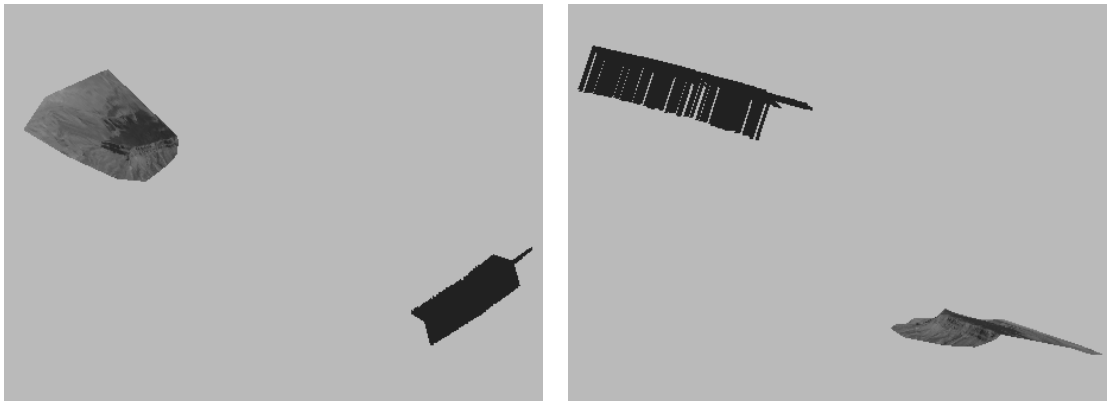
In this example we assume that all the intrinsic parameters (except the skews) are unknown, and may differ from camera to camera. The normalization algorithm described in Section 2.3.4 is applied. We initialize the aspect ratios to 1 and initialize the principal points to the middle of the images. Figure 2.10 shows the reconstructed LED positions and the reconstructed camera orientations and locations. The reconstructed LED positions are compared with their known positions. The maximum distance



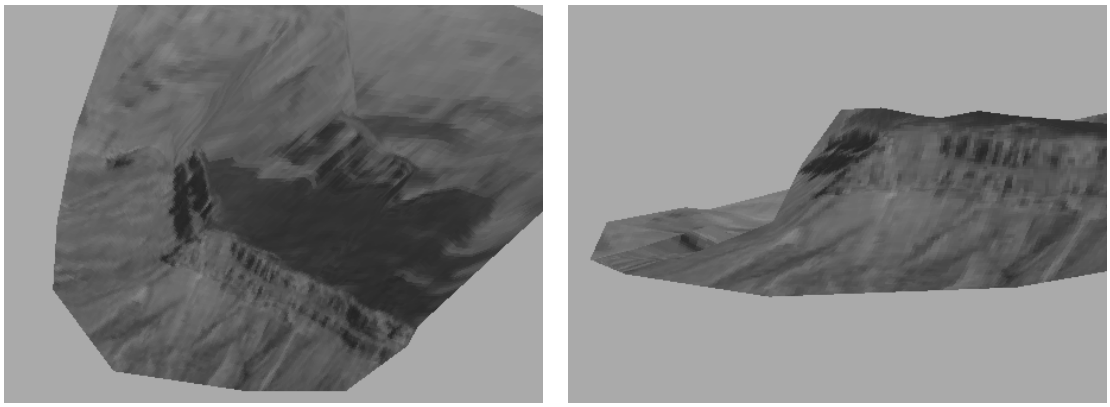
(a) (b)

(c) (d)

Figure 2.7: **Grand Canyon sequence input:** (a) 1st image, (b) 46th image, (c) 91st image of the Grand Canyon sequence. (d) 1st image of the Grand Canyon sequence with the feature points overlaid.



(a)



(b)

Figure 2.8: **Grand Canyon sequence results:** (a)Top and side view of the reconstruction, the 3-axis figures denote the recovered cameras. (b)Top and side view of the reconstructed Grand Canyon with texture mapping.

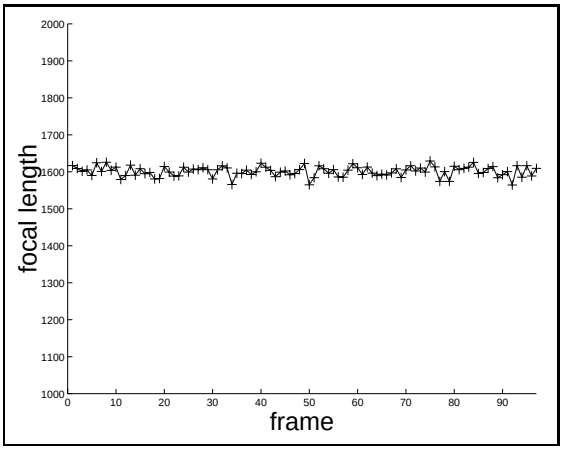


Figure 2.9: **Grand Canyon sequence:** Focal lengths of the Grand Canyon sequence recovered by the uncalibrated reconstruction method. The recovered values are relatively constant as expected.

is 20mm which is about 0.61% of the bar length. The recovered camera locations and orientations are compared with Tsai's calibration results. The maximum distance between the recovered camera locations by the two methods is 32mm which is about 0.98% of the bar length, the maximum angle between the recovered camera orientations is 0.3°.

Figure 2.11 are plots of the differences of the focal lengths, the principal points and the aspect ratios recovered by the uncalibrated reconstruction method and by Tsai's calibration algorithm. The plots show that the calibration results of these two methods are very close.

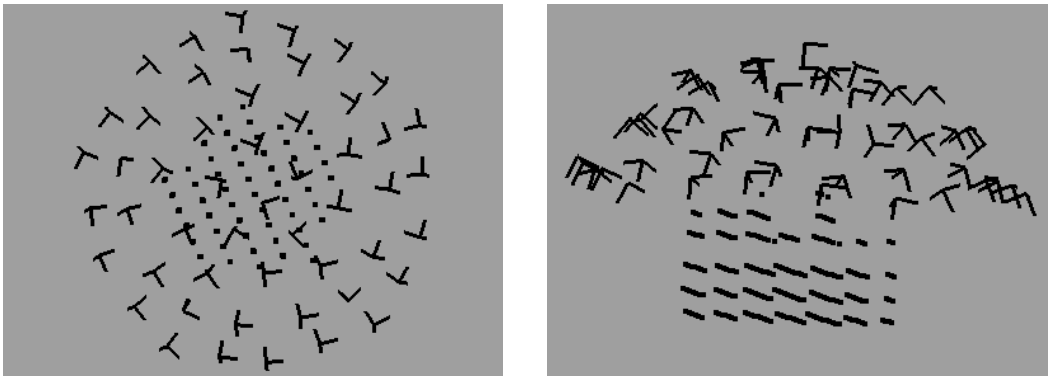
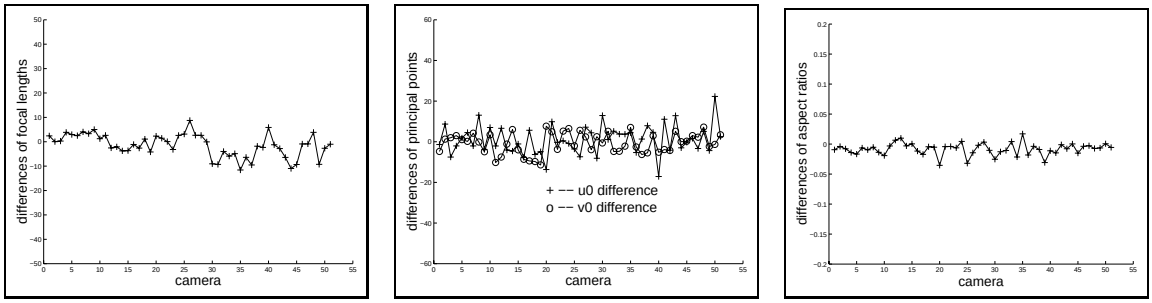


Figure 2.10: **Calibration setup results:** Top and side view of the reconstruction of the calibration setup, the points denote the recovered LED positions, the 3-axis figures are the recovered cameras.



(a) (b) (c)

Figure 2.11: **Calibration setup:** Differences of (a)the focal lengths (b) the principal points (u_0, v_0) (c) the aspect ratios of the calibration setup data recovered by the uncalibrated reconstruction method and by Tsai's calibration algorithm.

Chapter 3

Multiple Motion Scene Reconstruction

In this chapter we describe a method [Han and Kanade, 2000b] for reconstruction of scenes containing an unknown number of moving objects. We refer to such scenes as *multiple motion scenes*. The multiple motion scene reconstruction method recovers the scene structure, the trajectories of the moving objects and the camera motion simultaneously from monocular image sequences. The number of the moving objects is automatically detected without prior motion segmentation. We also discuss solutions to the degenerate cases when the scene structure or the motion space is degenerate. Extensions of the multiple motion scene reconstruction method to weak perspective and perspective projections are presented as well. Experiments on synthetic and real image sequences show that the multiple motion scene reconstruction method is reliable under noise.

3.1 Feature points representation

We propose a unified representation of the static scene and the moving objects. Assuming that m feature points are tracked over n images, some of them static and the others moving linearly with constant speeds, we regard every feature point as a moving point with constant velocity: the static points simply have zero velocity. Any point $\mathbf{p}_{ij}$ is represented by,

$$\mathbf{p}_{ij} = \mathbf{s}_j + i\mathbf{v}_j \tag{3.1}$$

in a world coordinate system, where $i = 1 \cdots n$ and $j = 1 \cdots m$. n is the number of frames and m is the number of feature points. $\mathbf{s}_j$ is the point position at frame 0 (i.e., when the 0th frame is taken) and $\mathbf{v}_j$ is its motion velocity.

We first use the orthographic camera model for the derivations. We describe its extensions to weak perspective and perspective camera models in Section 3.4. If a point $\mathbf{p}_{ij}$ is observed in frame i at image

coordinates $(u_{ij} \;\; v_{ij})$, then,

$$\begin{aligned} u_{ij} &= \mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi} \\ v_{ij} &= \mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi} \end{aligned} \tag{3.2}$$

$\mathbf{i}_i$ and $\mathbf{j}_i$ are the rotation axes of the ith camera. t_{xi} and t_{yi} are its translations. Therefore,

$$\begin{aligned} u_{ij} &= \mathbf{i}_i \cdot \mathbf{s}_j + i\,\mathbf{i}_i \cdot \mathbf{v}_j + t_{xi} \\ v_{ij} &= \mathbf{j}_i \cdot \mathbf{s}_j + i\,\mathbf{j}_i \cdot \mathbf{v}_j + t_{yi} \end{aligned} \tag{3.3}$$

We put all the feature points coordinates $(u_{ij} \;\; v_{ij})$ in a $2n \times m$ measurement matrix W,

$$W = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1m} \\ v_{11} & v_{12} & \dots & v_{1m} \\ & \vdots & \vdots & \\ u_{n1} & u_{n2} & \dots & u_{nm} \\ v_{n1} & v_{n2} & \dots & v_{mn} \end{bmatrix} \tag{3.4}$$

Each column of W contains the observations for a single point, and each row contains the observed u-coordinates or v-coordinates for a single frame. We have,

$$W = MS + T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{3.5}$$

with the rotation matrix,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{x2} & \mathbf{n}_{y2} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} \end{bmatrix}^{\mathrm{T}} \tag{3.6}$$

where

$$\begin{aligned} \mathbf{m}_{xi} &= \mathbf{i}_i \qquad \mathbf{n}_{xi} = i\,\mathbf{i}_i \\ \mathbf{m}_{yi} &= \mathbf{j}_i \qquad \mathbf{n}_{yi} = i\,\mathbf{j}_i \end{aligned} \tag{3.7}$$

and the shape matrix,

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \tag{3.8}$$

The translation vector T is,

$$T = \begin{bmatrix} t_{x1} & t_{y1} & t_{x2} & t_{y2} & \cdots & t_{xn} & t_{yn} \end{bmatrix}^{\mathrm{T}} \tag{3.9}$$

The constraints of the objects moving linearly with constant speeds enable the unified representation of the motion matrix M, composed of the rotation axes ($\mathbf{m}_{xi}$ and $\mathbf{m}_{yi}$) and the scaled rotation axes ($\mathbf{n}_{xi}$ and $\mathbf{n}_{yi}$), and of the shape matrix, composed of the scene structure ($\mathbf{s}_j$) and the motion velocities ($\mathbf{v}_j$).

3.2 Scene reconstruction

In this section we describe the multiple motion scene reconstruction method [Han and Kanade, 1999a] based on the unified representation of the static scene and the moving objects. The method factors the measurement matrix into the product of the unified motion matrix, which is a combination of the rotation and the scaled rotation axes, and the unified shape matrix, which is a combination of the initial positions of the feature points and their velocities.

3.2.1 Moving world coordinate system location

As a set of points are either static or moving linearly at constant speeds, the center of gravity of all the points is moving linearly at a constant speed as well. The velocity of the center of gravity is equal to the average of all the velocities ($\mathbf{v}_j$). We transform the 3D representation to a **moving** world coordinate system whose origin is at the center of gravity of all the feature points and with a fixed orientation (such as being aligned with the first camera). Therefore,

$$\sum_{j=1}^{m} \mathbf{p}_{ij} = 0 \tag{3.10}$$

From Equation (3.2), we have,

$$\begin{aligned} \sum_{j=1}^{m} u_{ij} &= \sum_{j=1}^{m} (\mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi}) = \mathbf{i}_i \sum_{j=1}^{m} \mathbf{p}_{ij} + m t_{xi} = m t_{xi} \\ \sum_{j=1}^{m} v_{ij} &= \sum_{j=1}^{m} (\mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi}) = \mathbf{j}_i \sum_{j=1}^{m} \mathbf{p}_{ij} + m t_{yi} = m t_{yi} \end{aligned} \tag{3.11}$$

We can compute the translation vector directly from Equation (3.11),

$$\begin{aligned} t_{xi} &= \frac{1}{m} \sum_{j=1}^{m} u_{ij} \\ t_{yi} &= \frac{1}{m} \sum_{j=1}^{m} v_{ij} \end{aligned} \tag{3.12}$$

3.2.2 Decomposition

Once the translation vector T is known, we subtract it from W in Equation (3.5),

$$\hat{W} = W - T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} = \hat{M}\hat{S} = \hat{M}AA^{-1}\hat{S} = MS \tag{3.13}$$

where $M = \hat{M}A$ and $S = A^{-1}\hat{S}$. According to the representations of M and S in Equations (3.6) and (3.8), we know that the rank of the matrix $\hat{W}$ is at most 6 no matter how many moving objects are there. We perform a SVD on $\hat{W}$ and get the best possible rank 6 approximation of $\hat{W}$ as $\hat{M}\hat{S}$, where $\hat{M}$ is a $2n \times 6$ matrix and $\hat{S}$ is a $6 \times m$ matrix. This decomposition is not unique. Any non-singular 6×6 matrix A could be inserted between $\hat{M}$ and $\hat{S}$ to get another motion and shape pair.

3.2.3 Normalization

Metric constraints are imposed to translate the current pair of motion ($\hat{M}$) and shape ($\hat{S}$) to the Euclidean solutions through recovering the linear transformation A. This process is called *normalization*. We recover this 6×6 matrix A by observing that the rows of the motion matrix M consist of the rotation axes and the scaled ones (Equation (3.6)),

$$|\mathbf{m}_{xi}|^2 = 1 \quad |\mathbf{m}_{yi}|^2 = 1 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \tag{3.14}$$

$$|\mathbf{n}_{xi}|^2 = i^2 \quad |\mathbf{n}_{yi}|^2 = i^2 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \tag{3.15}$$

$$\mathbf{m}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \tag{3.16}$$

The above equations impose linear constraints on the elements of MM^{T}. Since

$$MM^{\mathrm{T}} = \hat{M}AA^{\mathrm{T}}\hat{M}^{\mathrm{T}} \tag{3.17}$$

these constraints are linear on the elements of the symmetric matrix $Q = AA^{\mathrm{T}}$.

Define

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{3.18}$$

where A is 6×6 matrix and A_1, A_2 are both 6×3 matrices. Since $M = \hat{M}A$,

$$\begin{aligned} \hat{M}A_1 &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \end{bmatrix}^{\mathrm{T}} \\ \hat{M}A_2 &= \begin{bmatrix} \mathbf{n}_{x1} & \mathbf{n}_{y1} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} \end{bmatrix}^{\mathrm{T}} \\ &= N \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{3.19}$$

where

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & n \end{bmatrix} \tag{3.20}$$

according to Equation (3.7). Therefore,

$$\hat{M}A_2 = N\hat{M}A_1 \tag{3.21}$$

The matrix A_2 is over constrained given A_1 and $\hat{M}$ by,

$$A_2 = KA_1 \tag{3.22}$$

where

$$K = \hat{M}^{-1}N\hat{M} \tag{3.23}$$

and $\hat{M}^{-1}$ is the pseudo inverse matrix which is $6 \times 2n$ and uniquely defined when $n \geq 3$.

From Equation (3.19), we see that Equation (3.14) imposes constraints on the 21 unknown elements of the 6×6 symmetric matrix $Q_1 = A_1A_1^{\mathrm{T}}$, while Equation (3.15) imposes constraints on the 21 elements of $Q_2 = A_2A_2^{\mathrm{T}}$. From the relation of A_1 and A_2 (Equation (3.22)), we have,

$$Q_2 = A_2A_2^{\mathrm{T}} = KA_1A_1^{\mathrm{T}}K^{\mathrm{T}} = KQ_1K^{\mathrm{T}} \tag{3.24}$$

which translates the constraints on Q_2 to the constraints on Q_1.

Equation (3.16) imposes constraints on $Q_3 = A_2A_1^{\mathrm{T}}$ which can also be translated into constraints on Q_1,

$$Q_3 = A_2A_1^{\mathrm{T}} = KA_1A_1^{\mathrm{T}} = KQ_1 \tag{3.25}$$

Therefore, each frame contributes 8 constraints (Equations (3.14) to (3.16)) on Q_1. In total, we have $8n$ equations on the 21 unknown elements of the symmetric matrix Q_1. Linear least squares solutions are computed. We then compute the matrix A_1 from Q_1 by rank 3 matrix decomposition and A_2 by Equation (3.22), so we recover the linear transformation $A = [A_1 \;\; A_2]$.

3.2.4 Motion and shape reconstruction

Once the matrix A has been found, the shape matrix is computed using $S = A^{-1}\hat{S}$ and the motion matrix is $M = \hat{M}A$. We compute the camera rotation axes as,

$$\mathbf{i}_i = \mathbf{m}_{xi} \quad \mathbf{j}_i = \mathbf{m}_{yi} \quad \mathbf{k}_i = \mathbf{m}_{xi} \times \mathbf{m}_{yi} \tag{3.26}$$

The shape matrix consists of the scene structure and the velocities (represented in the moving world coordinate system). We need to transform the representation back to a fixed world coordinate system with the origin at the center of gravity of all the points at frame 1.

First we compute the velocity of the moving coordinate system. Since the system is moving at the average velocity of all the moving points, the static points must have the same velocity which is the negative value of the average velocity. It is often the case that there are more static points than moving points from each moving object, so we let every point vote for a "common" velocity (denoted as $\mathbf{v}_c$). The velocity with the most votes is taken as the negative velocity of the moving coordinate system. The points with the "common" velocity are automatically classified as static and the scene structure is computed as:

$$\mathbf{sc}_j = \mathbf{s}_j + \mathbf{v}_c \tag{3.27}$$

where $\mathbf{sc}_j$ denotes the scene point position represented in the fixed coordinate system. According to Equation (3.1), $\mathbf{s}_j$ is the point position at frame 0.

The points which do not have the "common" velocity are the moving points. The number of the moving objects is therefore detected. Their starting positions represented in the fixed coordinate system are:

$$\mathbf{sm}_j = \mathbf{s}_j + \mathbf{v}_c \tag{3.28}$$

and their velocities are:

$$\mathbf{vm}_j = \mathbf{v}_j - \mathbf{v}_c \tag{3.29}$$

3.2.5 Summary of algorithm

We summarize the reconstruction method as follows:

1. Compute the camera translations T from the matrix W according to Equation (3.12);
2. Subtract T from W to generate $\hat{W}$ according to Equation (3.13);
3. Perform SVD on $\hat{W}$ and get $\hat{M}$ and $\hat{S}$;
4. Set up linear equations of the 21 unknown elements of the symmetric matrix Q_1 by imposing constraints in Equations (3.14) to (3.16);

5. Factor Q_1 to get A_1 from $Q_1 = A_1 A_1^{\mathrm{T}}$;
6. Compute A_2 from $A_2 = K A_1$;
7. Combine A_1 and A_2 to generate the linear transformation matrix $A = [A_1 \;\; A_2]$;
8. Recover the shape matrix using $S = A^{-1}\hat{S}$ and motion matrix using $M = \hat{M}A$;
9. Recover the camera rotation axes as in Equation (3.26);
10. Detect the moving objects, reconstruct the scene structure and the trajectories of the moving objects according to Equations (3.27) to (3.29).

3.3 Degenerate cases

The method described in Section 3.2 solves the case where the "registered" measurement matrix (the matrix generated by subtraction of translations from the original measurement matrix) has the full rank 6, that is, where the static structure and the motion space of the objects are both rank 3. Equivalently, this is the case that the scene is three dimensional and the velocities of the moving objects span a three dimensional space. In this section we discuss degenerate cases.

If the scene has a degenerate shape, such as all the points lie in a plane, the plane plus parallax method [Irani *et al.*, 1998] can detect the situation and solve for the scene structure (plane position), the camera motion and the motion segmentation [Anandan *et al.*, 1994, Irani *et al.*, 1997]. The motion trajectories can be recovered using the method proposed by Avidan and Shashua [Avidan and Shashua, 1999], given the reconstruction of the camera motion. Therefore, in this section we only discuss the solutions to the degenerate motion space of the objects.

We classify the degenerate situations into three classes:

1. Rank-3 case: The matrix $\hat{W}$ has rank 3. This corresponds to the situation where there is no moving object in the scene. The one-object factorization method [Tomasi and Kanade, 1992] is used to recover the scene structure and the camera motion.

2. Rank-4 case: The matrix $\hat{W}$ has rank 4. This corresponds to the situation where there is one moving object or multiple objects moving in the same and/or the opposite direction (not necessarily the same 3D line). Section 3.3.2 describes a linear algorithm for this case.

3. Rank-5 case: The matrix $\hat{W}$ has rank 5. This corresponds to the situation where the velocities of the objects lie in a two dimensional space (not necessarily the same 3D plane). Section 3.3.3 gives a nonlinear solution to this case.

3.3.1 Rank approximation

Given tracked feature points, we first need to decide which case (full rank or one of the above three degenerate cases) is the best approximation. The rank of the matrix $\hat{W}$ is one important clue. However, finding the rank of $\hat{W}$ is not straightforward. Both inaccuracies in feature locations and approximation of perspective projection using orthographic or weak perspective projections induce noises in the rank computation.

We use an algorithm similar to [Boult and Brown, 1991] and [Irani, 1999] to detect the rank of $\hat{W}$. We first estimate the noise level of the input images and approximate the rank using the singular values of $\hat{W}$ and the noise level. We refer to this method as *direct rank approximation*. In [Gear, 1998], Gear proposed a maximum likelihood method to estimate the grouping of points in the presence of noise. One of the core techniques of the method is rank approximation. He evaluated the grouping errors of all the possible rank values based on the statistical noise model. The rank value with the minimum error is chosen as the best rank approximation. We applied Gear's idea to the multiple motion scene reconstruction method where the rank of $\hat{W}$ can only be any value in $\{3, 4, 5, 6\}$, which is determined by the motion space of the objects and is not dependent on the number of moving objects. Compared with Gear's method [Gear, 1998] and Costeira and Kanade's method [Costeira and Kanade, 1998], in which the rank value is used to detect the number of moving objects and is affected by degenerate shapes, this rank estimation has much less computation. For each rank value in $\{3, 4, 5, 6\}$, we perform the multiple motion scene reconstruction and measure the error in orthogonality of the recovered camera rotation matrices as well as the discrepancies of the feature points back projections. The best rank approximation is the one with the minimum error. The results show that the direct rank approximation method gives reliable estimations of the rank at most times.

3.3.2 Rank-4 case

When only one moving object is in the scene, or when all the moving objects travel in the same or the opposite direction, the motion space is one dimensional and the rank of the "registered" measurement matrix is 4. In this case we align the $\mathbf{x}$ direction of the world coordinate system with the motion direction. The system is still moving with the constant velocity. Therefore, the motion and shape matrices are (compare with Equations (3.6) and (3.8)),

$$\begin{aligned} M &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ i_{x1} & j_{x1} & 2i_{x2} & 2j_{x2} & \cdots & ni_{xn} & nj_{xn} \end{bmatrix}^{\mathrm{T}} \\ S &= \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \end{bmatrix} \end{aligned} \tag{3.30}$$

where i_{xi} and j_{xi} represent the x-elements of the ith rotation axes, v_{xj} denotes the x-element of the velocity of the jth feature point. We apply similar derivations as in the full rank case to the computation of T (Equation (3.12)) and the decomposition of $\hat{W}$ (Equation(3.13)). In this case the rank of $\hat{W}$ is 4. We perform a rank 4 matrix decomposition on $\hat{W}$ and get a $2n \times 4$ matrix $\hat{M}$ and a $4 \times m$ matrix $\hat{S}$. Now the linear transformation matrix A is 4×4. Similarly, we define

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{3.31}$$

where A_1 is 4×3, A_2 is 4×1 and we have,

$$A_2 = K(A_1)_1 \tag{3.32}$$

where $(A_1)_1$ is the first column of A_1 and K is defined in Equation(3.23). Since the matrix M consists of the rotation axes and only the x-elements of the scaled rotation axes, the constraints in Equations (3.15) and (3.16) cannot be represented as linear constraints on the elements of MM^{T}. However, the constraints in Equation (3.14) still hold and provide full rank linear equations on the 10 unknown elements of the symmetric 4×4 matrix $Q_1 = A_1 A_1^{\mathrm{T}}$. Least squares solutions are computed. We then compute A_1 by rank 3 matrix decomposition of Q_1. This decomposition is up to a three dimensional rotation R since the matrix Q_1 is symmetric. When the motion space is full rank, any rotation matrix R provides a valid reconstruction with a different orientation of the world coordinate system. We usually fix the matrix R by aligning the world coordinate system with the first camera orientation. However, when the motion space is degenerate, the alignment is constrained to make the orientation of the world coordinate system consistent with the motion direction(s).

In rank-4 case, we need to align the $\mathbf{x}$ direction of the world coordinate system as the motion direction before we compute A_2 according to Equation (3.32). The matrix R is determined by aligning the matrix $\hat{M}KA_1$ with the matrix $N\hat{M}A_1$.

Therefore, the linear transformation A is,

$$A = \begin{bmatrix} A_1 R & K(A_1 R)_1 \end{bmatrix} \tag{3.33}$$

where $()_1$ denotes the first column of the matrix. We apply a derivation similar to the one in Section 3.2.4 to recover the motion and shape.

3.3.3 Rank-5 case

When the velocities of all the moving objects lie in a two dimensional space, we assume that the $\mathbf{x} - \mathbf{y}$ plane of the world coordinate system is aligned with the two dimensional motion space. The

system is still moving with constant velocity. Therefore, the motion and shape matrices are,

$$
\begin{aligned}
M &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ i_{x1} & j_{x1} & 2i_{x2} & 2j_{x2} & \cdots & ni_{xn} & nj_{xn} \\ i_{y1} & j_{y1} & 2i_{y2} & 2j_{y2} & \cdots & ni_{yn} & nj_{yn} \end{bmatrix}^{\mathrm{T}} \\
S &= \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \\ v_{y1} & v_{y2} & \cdots & v_{ym} \end{bmatrix}
\end{aligned} \tag{3.34}
$$

where i_{xi} and j_{xi} represent the x-elements of the ith rotation axes, i_{yi} and j_{yi} represent their y-elements, v_{xj} denotes the x-element of the velocity of the jth feature point and v_{yj} is its y-element. Therefore, the rank of $\hat{W}$ is 5. Similar derivations apply to the computation of T (Equation (3.12)) and the decomposition of $\hat{W}$ (Equation(3.13)). In this case we perform a rank 5 matrix decomposition on $\hat{W}$ and get a $2n \times 5$ matrix $\hat{M}$ and a $5 \times m$ matrix $\hat{S}$. The linear transformation matrix A is 5×5. Similarly, we define

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{3.35}$$

where A_1 is 5×3 and A_2 is 5×2,

$$A_2 = K(A_1)_{12} \tag{3.36}$$

where $(A_1)_{12}$ denotes the first two columns of A_1 and K is defined in Equation (3.23). Here only the constraints in Equation (3.14) can be represented as linear constraints on the elements of $Q_1 = A_1 A_1^{\mathrm{T}}$. In this case the constraints are not sufficient to solve for the 15 unknown elements of the symmetric 5×5 matrix Q_1 linearly.

The constraints in Equations (3.15) and (3.16) can be represented as constraints on the elements of Q_1 and the five elements of the third column of A_1, which is a 5×1 vector denoted by $\mathbf{c}$. According to Equation (3.36),

$$\begin{bmatrix} A_2 & K\mathbf{c} \end{bmatrix} = KA_1 \tag{3.37}$$

we have,

$$A_2 A_2^{\mathrm{T}} = KA_1 A_1^{\mathrm{T}} K^{\mathrm{T}} - K\mathbf{c}\mathbf{c}^{\mathrm{T}} K^{\mathrm{T}} = KQ_1 K^{\mathrm{T}} - K\mathbf{c}\mathbf{c}^{\mathrm{T}} K^{\mathrm{T}} \tag{3.38}$$

and

$$
\begin{aligned}
\begin{bmatrix} A_2 & i\mathbf{c} \end{bmatrix} A_1^{\mathrm{T}} &= A_2 (A_1)_{12}^{\mathrm{T}} + i\mathbf{c}\mathbf{c}^{\mathrm{T}} \\
&= KA_1 A_1^{\mathrm{T}} - K\mathbf{c}\mathbf{c}^{\mathrm{T}} + i\mathbf{c}\mathbf{c}^{\mathrm{T}} \\
&= KQ_1 - K\mathbf{c}\mathbf{c}^{\mathrm{T}} + i\mathbf{c}\mathbf{c}^{\mathrm{T}}
\end{aligned} \tag{3.39}
$$

Since,

$$\begin{array}{rclcrcl} \mathbf{m}_{xi}^{\mathrm{T}} & = & \hat{\mathbf{m}}_x^{(i)\mathrm{T}} A_1 & & \mathbf{n}_{xi}^{\mathrm{T}} & = & \left[\begin{array}{cc} \hat{\mathbf{m}}_x^{(i)\mathrm{T}} A_2 & i\,\hat{\mathbf{m}}_x^{(i)\mathrm{T}}\mathbf{c} \end{array} \right] \\ \mathbf{m}_{yi}^{\mathrm{T}} & = & \hat{\mathbf{m}}_y^{(i)\mathrm{T}} A_1 & & \mathbf{n}_{yi}^{\mathrm{T}} & = & \left[\begin{array}{cc} \hat{\mathbf{m}}_y^{(i)\mathrm{T}} A_2 & i\,\hat{\mathbf{m}}_y^{(i)\mathrm{T}}\mathbf{c} \end{array} \right] \end{array} \tag{3.40}$$

according to Equation (3.19). $\hat{\mathbf{m}}_x^{(i)}$ and $\hat{\mathbf{m}}_y^{(i)}$ represent the ith x and y rows of the matrix $\hat{M}$. They are both 5×1 vectors. We translate the constraints in Equation (3.15) to the constraints on Q_1 and $\mathbf{c}$ according to Equation (3.38),

$$\begin{aligned} |\mathbf{n}_{xi}|^2 &= \hat{\mathbf{m}}_x^{(i)\mathrm{T}} A_2 A_2^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} \\ &= \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K Q_1 K^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K \mathbf{c}\mathbf{c}^{\mathrm{T}} K^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} = i^2 \end{aligned} \tag{3.41}$$

and,

$$\begin{aligned} |\mathbf{n}_{yi}|^2 &= \hat{\mathbf{m}}_y^{(i)\mathrm{T}} K Q_1 K^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_y^{(i)\mathrm{T}} K \mathbf{c}\mathbf{c}^{\mathrm{T}} K^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} + i^2 \hat{\mathbf{m}}_y^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} = i^2 \\ \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} &= \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K Q_1 K^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K \mathbf{c}\mathbf{c}^{\mathrm{T}} K^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} + i^2 \hat{\mathbf{m}}_x^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} = 0 \end{aligned} \tag{3.42}$$

Similarly, we translate the constraints in Equation (3.16) to the constraints on Q_1 and $\mathbf{c}$ according to Equation (3.39),

$$\begin{aligned} \mathbf{m}_{xi} \cdot \mathbf{n}_{yi} &= \hat{\mathbf{m}}_y^{(i)\mathrm{T}} \left[\begin{array}{cc} A_2 & i\mathbf{c} \end{array} \right] A_1^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} \\ &= \hat{\mathbf{m}}_y^{(i)\mathrm{T}} K Q_1 \hat{\mathbf{m}}_x^{(i)} - \hat{\mathbf{m}}_y^{(i)\mathrm{T}} K \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} + i \hat{\mathbf{m}}_y^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_x^{(i)} = 0 \end{aligned} \tag{3.43}$$

and

$$\mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K Q_1 \hat{\mathbf{m}}_y^{(i)} - \hat{\mathbf{m}}_x^{(i)\mathrm{T}} K \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} + i \hat{\mathbf{m}}_x^{(i)\mathrm{T}} \mathbf{c}\mathbf{c}^{\mathrm{T}} \hat{\mathbf{m}}_y^{(i)} = 0 \tag{3.44}$$

Therefore, we get linear equations of the 15 unknown elements of Q_1 and the 15 unknown elements of $\mathbf{c}\mathbf{c}^{\mathrm{T}}$. Since these equations cannot provide full rank constraints on the 30 unknowns, there is no linear solutions of Q_1 and $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ directly. However, the constraints are full rank on the elements of Q_1 if $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ is given. That is, if $\mathbf{c}$ can be computed, we can get a linear solution of Q_1. In this way we change the problem to a small scale nonlinear optimization on the 5 elements of $\mathbf{c}$. Once the vector $\mathbf{c}$ is computed, the matrix Q_1 is computed by least squares solutions. A_1 is then calculated from Q_1.

Same to the rank-4 case, we need to align the $\mathbf{x} - \mathbf{y}$ plane of the world coordinate system with the two dimensional motion space before we compute A_2 according to Equation (3.36). The matrix R is also determined by aligning the matrix $\hat{M} K A_1$ with the matrix $N \hat{M} A_1$. The alignment problem is solved by the least eigenvalue method.

Therefore, the linear transformation A is,

$$A = \begin{bmatrix} A_1 R & K(A_1 R)_{12} \end{bmatrix} \tag{3.45}$$

We apply a derivation similar to the one in Section 3.2.4 to recover the motion and shape.

3.4 Extensions to weak perspective and perspective projections

3.4.1 Scene reconstruction under weak perspective projection

Based on the unified representation of the static scene and the moving objects proposed in Section 3.1, any point $\mathbf{p}_{ij}$ is defined as,

$$\mathbf{p}_{ij} = \mathbf{s}_j + i\mathbf{v}_j \tag{3.46}$$

in a world coordinate system, where $i = 1 \cdots n$ and $j = 1 \cdots m$. n is the number of frames and m is the number of feature points. $\mathbf{s}_j$ is the point position at frame 0 and $\mathbf{v}_j$ is its motion velocity.

The image coordinates $(u_{ij} \;\; v_{ij})$ of a point $\mathbf{p}_{ij}$ in frame i under weak perspective projection are,

$$\begin{aligned} u_{ij} &= \frac{\mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi}}{z_i} \\ v_{ij} &= \frac{\mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi}}{z_i} \end{aligned} \tag{3.47}$$

$\mathbf{i}_i$ and $\mathbf{j}_i$ are the rotation axes of the ith camera. t_{xi} and t_{yi} are the translations. z_i is the distance between the ith camera optical center and the center of gravity of all the feature points. Therefore,

$$\begin{aligned} u_{ij} &= \frac{\mathbf{i}_i}{z_i} \cdot \mathbf{s}_j + i\,\frac{\mathbf{i}_i}{z_i} \cdot \mathbf{v}_j + \frac{t_{xi}}{z_i} \\ v_{ij} &= \frac{\mathbf{j}_i}{z_i} \cdot \mathbf{s}_j + i\,\frac{\mathbf{j}_i}{z_i} \cdot \mathbf{v}_j + \frac{t_{yi}}{z_i} \end{aligned} \tag{3.48}$$

We put all the feature points coordinates $(u_{ij} \;\; v_{ij})$ in a $2n \times m$ measurement matrix W,

$$W = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1m} \\ v_{11} & v_{12} & \dots & v_{1m} \\ & \vdots & \vdots & \\ u_{n1} & u_{n2} & \dots & u_{nm} \\ v_{n1} & v_{n2} & \dots & v_{mn} \end{bmatrix} \tag{3.49}$$

We have,

$$W = MS + T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \tag{3.50}$$

with the rotation matrix,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{x2} & \mathbf{m}_{y2} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{x2} & \mathbf{n}_{y2} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} \end{bmatrix}^{\mathrm{T}} \tag{3.51}$$

where

$$\begin{array}{rclrcl} \mathbf{m}_{xi} & = & \frac{\mathbf{i}_i}{z_i} & \mathbf{n}_{xi} & = & i\,\frac{\mathbf{i}_i}{z_i} \\ \mathbf{m}_{yi} & = & \frac{\mathbf{j}_i}{z_i} & \mathbf{n}_{yi} & = & i\,\frac{\mathbf{j}_i}{z_i} \end{array} \tag{3.52}$$

and the shape matrix,

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \tag{3.53}$$

The translation vector T is,

$$T = \begin{bmatrix} \frac{t_{x1}}{z_1} & \frac{t_{y1}}{z_1} & \frac{t_{x2}}{z_2} & \frac{t_{y2}}{z_2} & \cdots & \frac{t_{xn}}{z_n} & \frac{t_{yn}}{z_n} \end{bmatrix}^{\mathrm{T}} \tag{3.54}$$

Now the unified representation of the rotation matrix M is composed of the rotation axes scaled by the object depth z_i ($\mathbf{m}_{xi}$ and $\mathbf{m}_{yi}$) and their scaled versions by the frame number i ($\mathbf{n}_{xi}$ and $\mathbf{n}_{yi}$). The unified representation of the shape matrix is composed of the scene structure ($\mathbf{s}_j$) and the motion velocities ($\mathbf{v}_j$), which is same as that under orthographic projection.

Moving world coordinate system location

As in Section 3.2.1, we transform the 3D representation to a moving world coordinate system with fixed orientation and the origin at the center of gravity of all the feature points. Therefore,

$$\sum_{j=1}^{m} \mathbf{p}_{ij} = 0 \tag{3.55}$$

From Equation (3.47), we have,

$$\begin{array}{l} \sum_{j=1}^{m} u_{ij} = \sum_{j=1}^{m} (\frac{\mathbf{i}_i}{z_i} \cdot \mathbf{p}_{ij} + \frac{t_{xi}}{z_i}) = \frac{\mathbf{i}_i}{z_i} \sum_{j=1}^{m} \mathbf{p}_{ij} + m\frac{t_{xi}}{z_i} = m\frac{t_{xi}}{z_i} \\ \sum_{j=1}^{m} v_{ij} = \sum_{j=1}^{m} (\frac{\mathbf{j}_i}{z_i} \cdot \mathbf{p}_{ij} + \frac{t_{yi}}{z_i}) = \frac{\mathbf{j}_i}{z_i} \sum_{j=1}^{m} \mathbf{p}_{ij} + m\frac{t_{yi}}{z_i} = m\frac{t_{yi}}{z_i} \end{array} \tag{3.56}$$

We get the vector T from Equation (3.56),

$$\begin{aligned} \frac{t_{xi}}{z_i} &= \frac{1}{m}\sum_{j=1}^{m} u_{ij} \\ \frac{t_{yi}}{z_i} &= \frac{1}{m}\sum_{j=1}^{m} v_{ij} \end{aligned} \tag{3.57}$$

Decomposition

We subtract the translation vector T from W in Equation (3.50),

$$\hat{W} = W - T \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} = \hat{M}\hat{S} = \hat{M}AA^{-1}\hat{S} = MS \tag{3.58}$$

where $M = \hat{M}A$ and $S = A^{-1}\hat{S}$. According to the representations of M and S in Equations (3.51) and (3.53), we know that the rank of the matrix $\hat{W}$ is at most 6. We perform a rank 6 SVD on $\hat{W}$ and get the best possible rank 6 approximation of $\hat{W}$ as $\hat{M}\hat{S}$, where $\hat{M}$ is a $2n \times 6$ matrix and $\hat{S}$ is a $6 \times m$ matrix. This decomposition is not unique since any non-singular 6×6 matrix A could be inserted between $\hat{M}$ and $\hat{S}$ to get another motion and shape pair.

Normalization

Metric constraints are imposed to translate the current pair of motion ($\hat{M}$) and shape ($\hat{S}$) to the Euclidean solutions through recovering the linear transformation A. We recover this 6×6 matrix A by observing that the rows of the motion matrix M consist of the scaled rotation axes and their corresponding scaled versions (Equation (3.51)),

$$|\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \tag{3.59}$$

$$|\mathbf{n}_{xi}|^2 = i^2|\mathbf{m}_{xi}|^2 \quad |\mathbf{n}_{yi}|^2 = i^2|\mathbf{m}_{yi}|^2 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \tag{3.60}$$

$$\mathbf{m}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \tag{3.61}$$

The above equations impose linear constraints on the elements of MM^{T}. Since

$$MM^{\mathrm{T}} = \hat{M}AA^{\mathrm{T}}\hat{M}^{\mathrm{T}} \tag{3.62}$$

these constraints are linear on the elements of the symmetric matrix $Q = AA^{\mathrm{T}}$.

The derivations to get the linear transformation A are same as described in Section 3.2.3. The same steps are also followed to solve for the degenerate cases under weak perspective projection.

Motion and shape reconstruction

Once the matrix A has been found, the shape matrix is computed using $S = A^{-1}\hat{S}$ and the motion matrix is $M = \hat{M}A$. We compute the depth z_i first,

$$z_i = \frac{1}{|\mathbf{m}_{xi}|} \tag{3.63}$$

then the camera rotation axes as,

$$\mathbf{i}_i = z_i\mathbf{m}_{xi} \quad \mathbf{j}_i = z_i\mathbf{m}_{yi} \quad \mathbf{k}_i = \mathbf{i}_i \times \mathbf{j}_i \tag{3.64}$$

and the translations are,

$$t_{xi} = \frac{z_i}{m}\sum_{j=1}^{m} u_{ij} \quad t_{yi} = \frac{z_i}{m}\sum_{j=1}^{m} v_{ij} \tag{3.65}$$

The shape matrix consists of the scene structure and the velocities represented in the moving world coordinate system. We need to transform the representation back to a fixed coordinate system with the origin at the center of gravity of all the points at frame 1. The moving objects are automatically detected at the same time. This process is same as described in Section 3.2.4.

3.4.2 Scene reconstruction under perspective projection

Based on the same unified representation of feature points, the image coordinates $(u_{ij} \;\; v_{ij})$ of a point $\mathbf{p}_{ij}$ in frame i under perspective projection are,

$$\begin{aligned} u_{ij} &= \frac{\mathbf{i}_i \cdot \mathbf{p}_{ij} + t_{xi}}{\mathbf{k}_i \cdot \mathbf{p}_{ij} + t_{zi}} \\ v_{ij} &= \frac{\mathbf{j}_i \cdot \mathbf{p}_{ij} + t_{yi}}{\mathbf{k}_i \cdot \mathbf{p}_{ij} + t_{zi}} \end{aligned} \tag{3.66}$$

$\mathbf{i}_i$, $\mathbf{j}_i$ and $\mathbf{k}_i$ are the rotation axes of the ith camera. t_{xi}, t_{yi} and t_{zi} are the translations. We divide both the numerators and the denominators of the above equations by t_{zi},

$$\begin{aligned} u_{ij} &= \frac{\frac{\mathbf{i}_i \cdot \mathbf{p}_{ij}}{t_{zi}} + \frac{t_{xi}}{t_{zi}}}{1 + \epsilon_{ij}} \\ v_{ij} &= \frac{\frac{\mathbf{j}_i \cdot \mathbf{p}_{ij}}{t_{zi}} + \frac{t_{yi}}{t_{zi}}}{1 + \epsilon_{ij}} \end{aligned} \tag{3.67}$$

where

$$\epsilon_{ij} = \frac{\mathbf{k}_i \cdot \mathbf{p}_{ij}}{t_{zi}} \tag{3.68}$$

Iterations of weak perspective reconstruction

Given tracked feature points, the perspective reconstruction can be regarded as non-linear parameter fitting of Equation (3.67) with camera motion and scene structure as parameters. The numerators in Equation (3.67) are the weak perspective projections. Christy and Horaud [Christy and Horaud, 1996a] presented the perspective factorization method by incremental weak perspective reconstructions. Their method worked on the scenes without moving objects. We applied their idea to the perspective reconstruction of multiple motion scenes. Whenever the objects are at some reasonable distance from the camera, the ϵ_{ij}'s are far less than 1. We compute the parameter fitting by iterations of the weak perspective approximations starting with $\epsilon_{ij} = 0$, that is, we perform the multiple motion scene weak perspective reconstruction method described in Section 3.4.1 on the measurement matrix W,

$$W = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ v_{11} & v_{12} & \cdots & v_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ u_{n1} & u_{n2} & \cdots & u_{nm} \\ v_{n1} & v_{n2} & \cdots & v_{nm} \end{bmatrix} \tag{3.69}$$

where n is the number of cameras and m is the number of feature points. The recovered motion parameters are denoted by $\mathbf{i'}_i$, $\mathbf{j'}_i$, $\mathbf{k'}_i$ and t'_{xi}, t'_{yi}, t'_{zi}. The recovered feature points are denoted by $\mathbf{p'}_{ij}$. We then use these current parameters to generate a new measurement matrix W':

$$W' = \begin{bmatrix} u'_{11} & u'_{12} & \cdots & u'_{1m} \\ v'_{11} & v'_{12} & \cdots & v'_{1m} \\ \cdots & \cdots & \cdots & \cdots \\ u'_{n1} & u'_{n2} & \cdots & u'_{nm} \\ v'_{n1} & v'_{n2} & \cdots & v'_{nm} \end{bmatrix} \tag{3.70}$$

where

$$\begin{aligned} u'_{ij} &= \frac{\mathbf{i'}_i \cdot \mathbf{p'}_{ij} + t'_{xi}}{\mathbf{k'}_i \cdot \mathbf{p'}_{ij} + t'_{zi}} \\ v'_{ij} &= \frac{\mathbf{j'}_i \cdot \mathbf{p'}_{ij} + t'_{yi}}{\mathbf{k'}_i \cdot \mathbf{p'}_{ij} + t'_{zi}} \end{aligned} \tag{3.71}$$

The process of generating the new measurement matrix is equivalent to the back projection process of other non-linear optimization methods. The new measurement matrix W' provides a criterion to choose between the two ambiguous reconstructions which are up to a mirror-symmetry transformation. The difference of W' from the original measurement matrix W also gives the convergence error. A

new iteration of the weak perspective reconstruction is performed on the current measurement matrix W'. The goal of the parameter fitting is to iteratively find the reconstructions which make the back projections consistent with the image measurements.

Choice between mirror-symmetric shapes

It is well known that there is an inherent ambiguity problem with any affine reconstruction method, that is, after any affine reconstruction we can get two mirror-symmetric shapes and the corresponding "mirror-symmetric" camera motions. Define

$$\epsilon_{ij}^{l} = \frac{\mathbf{k}_i^l \cdot \mathbf{p}_{ij}^l}{t_{zi}} \quad l = 1, 2 \tag{3.72}$$

and

$$\begin{array}{lll} k_{xi}^1 = -k_{xi}^2 & k_{yi}^1 = -k_{yi}^2 & k_{zi}^1 = k_{zi}^2 \\ p_{xij}^1 = p_{xij}^2 & p_{yij}^1 = p_{yij}^2 & p_{zij}^1 = -p_{zij}^2 \end{array} \tag{3.73}$$

According to Equation (3.72),

$$\epsilon_{ij}^1 = -\epsilon_{ij}^2 \tag{3.74}$$

For objects at reasonable distance from the camera, such as 5 to 20 times the object size, the weak perspective reconstruction method generates relatively correct reconstruction without considering the perspective effects. In the two new measurement matrices computed by Equations (3.70) and (3.71) for the two symmetric reconstructions, the perspective effects are taken care of by ϵ_{ij}'s. The ratio between the corresponding items of two W''s is $\frac{1+\epsilon_{ij}}{1-\epsilon_{ij}}$ which is large enough to distinguish the right shape from its mirror one. Based on this analysis, we keep only one set of the motion and shape parameters in each iteration, which is computation efficient.

Error measurement

We use the Frobenius norm of the difference matrix of the selected new measurement matrix and the original one as error E,

$$E = \| W' - W \|_{\mathrm{F}} \tag{3.75}$$

Perspective reconstruction method outline

The multiple motion scene perspective reconstruction method is summarized as follows:

1. Set $\epsilon_{ij} = 0$, for $i = 1 \cdots n$ and $j = 1 \cdots m$;
2. Generate the original measurement matrix W by equation (3.69) and start the iterations with $W' = W$;
3. Perform the multiple motion scene weak perspective reconstruction method on W' and generate two pairs of motion and shape which are mirror symmetric;
4. Calculate the two new measurement matrices with the sign reversal motions and shapes by Equations (3.70) and (3.71);
5. Compute the error E between the new measurement matrices and the original W as in Equation (3.75);
6. Choose the set of parameters with smaller error as the refined motion and shape, and define the corresponding measurement matrix as W';
7. If this error is close to zero, stop; else go to step 3.

3.5 Experiments

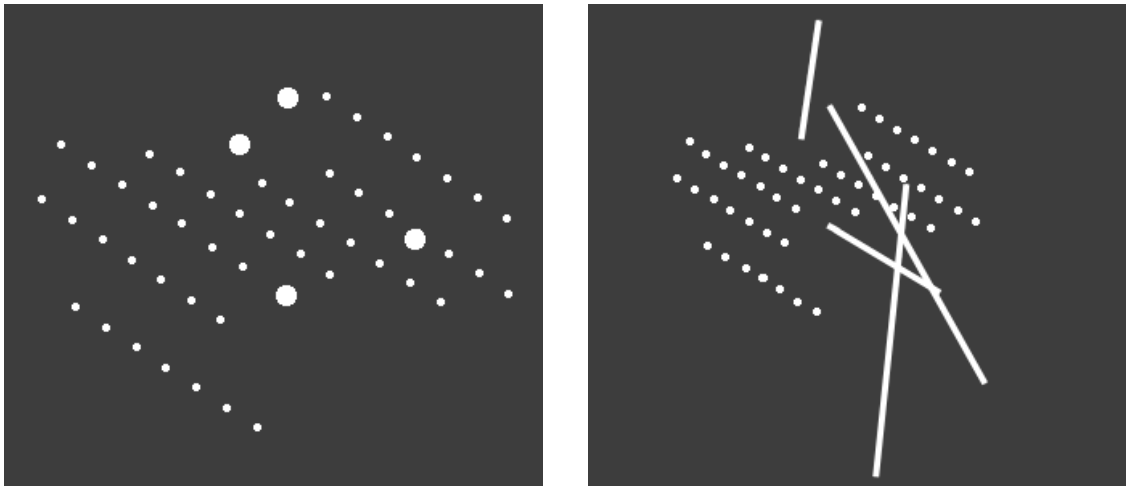
A number of experiments have been performed to test the effectiveness of the multiple motion scene reconstruction method presented in this chapter. First some synthetic images are used to evaluate the quality of the method. Then two experiments are conducted on real image sequences. The first sequence was taken by a hand-held camera of an indoor scene, and the reconstruction results are compared with the ground truth. The second sequence was taken by a small plane flying over the buildings. The weak perspective reconstruction method is used in the experiments described in this section because weak perspective projection is a better approximation to perspective projection than orthographic, and it is more reliable and efficient than the iterative perspective method.

3.5.1 Synthetic examples

We generate sequences of 100 frames with 49 feature points from the static scene and 0 to 9 objects moving in random directions. The shape of the static scene is a sweep of the sin curve in the space. The camera rotates randomly at 30 to 50 degrees around the scene. The distance between the camera and the scene is 15 to 50 times the static scene size. We add 2 pixels noise to the feature locations (the size of the image is 640×480).

Figure 3.1 illustrates the case where 4 objects are moving randomly in 3D space. The method

automatically detects the number of the moving objects as 4, reconstructs the static scene and the initial positions of the 4 moving objects, as shown in Figure 3.1(a). Figure 3.1(b) shows the trajectories of the moving objects as well as the static scene.



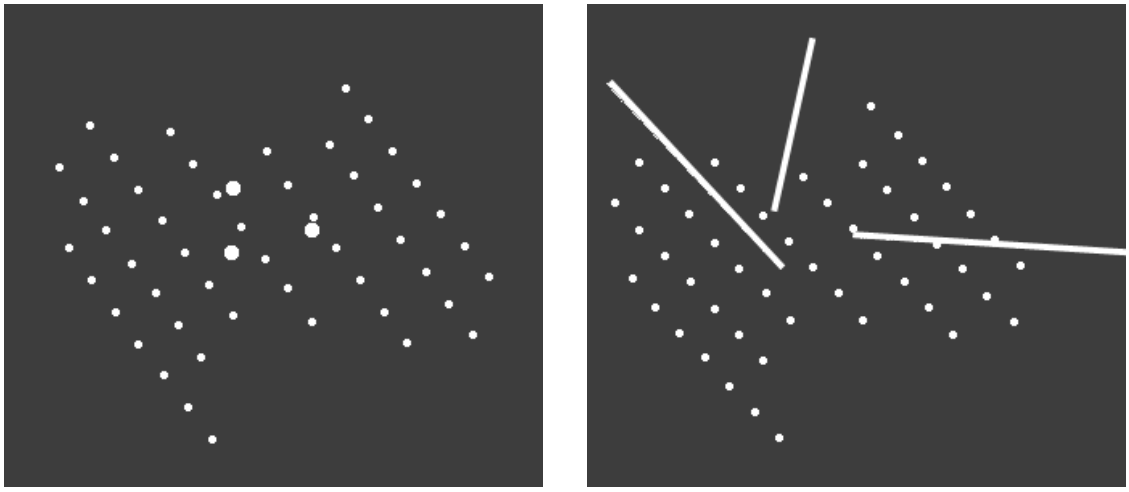
(a) (b)

Figure 3.1: **Full rank case:** A scene with a three dimensional motion space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.

We perform experiments on the case that there are two moving objects whose directions are on a plane. The method detects that the rank as 5 and recovers the scene structure and the two motion trajectories correctly. We also try the case that there are three moving objects but their motion directions lie in a two dimensional space. The method gets the right rank approximation (5) and the accurate reconstructions (shown in Figure 3.2).

We also conduct experiments on rank-4 cases that there is only one moving object, and that there are multiple moving objects which are moving in the same or the opposite direction. The method detects the rank as 4 in both cases. For the case that there is no moving object, the method correctly detects the rank as 3 and recovers the scene structure.

In all cases, we measure the reconstruction error by comparison with the ground truth. Since the reconstruction from monocular image sequences is up to a scale, we assume that the size of the static shape is 1. With 2 pixel standard noise, the maximum distance between the recovered static points and their known positions is 1.0%, the maximum error of the reconstructed initial positions of the moving objects is 1.2% and the velocity error is less than 1.1%. We also assess the quality of the camera motion reconstruction. The maximum distance between the recovered camera locations and the ground truth values is 1.4% and the maximum angle between the recovered camera orientations and the known



(a) (b)

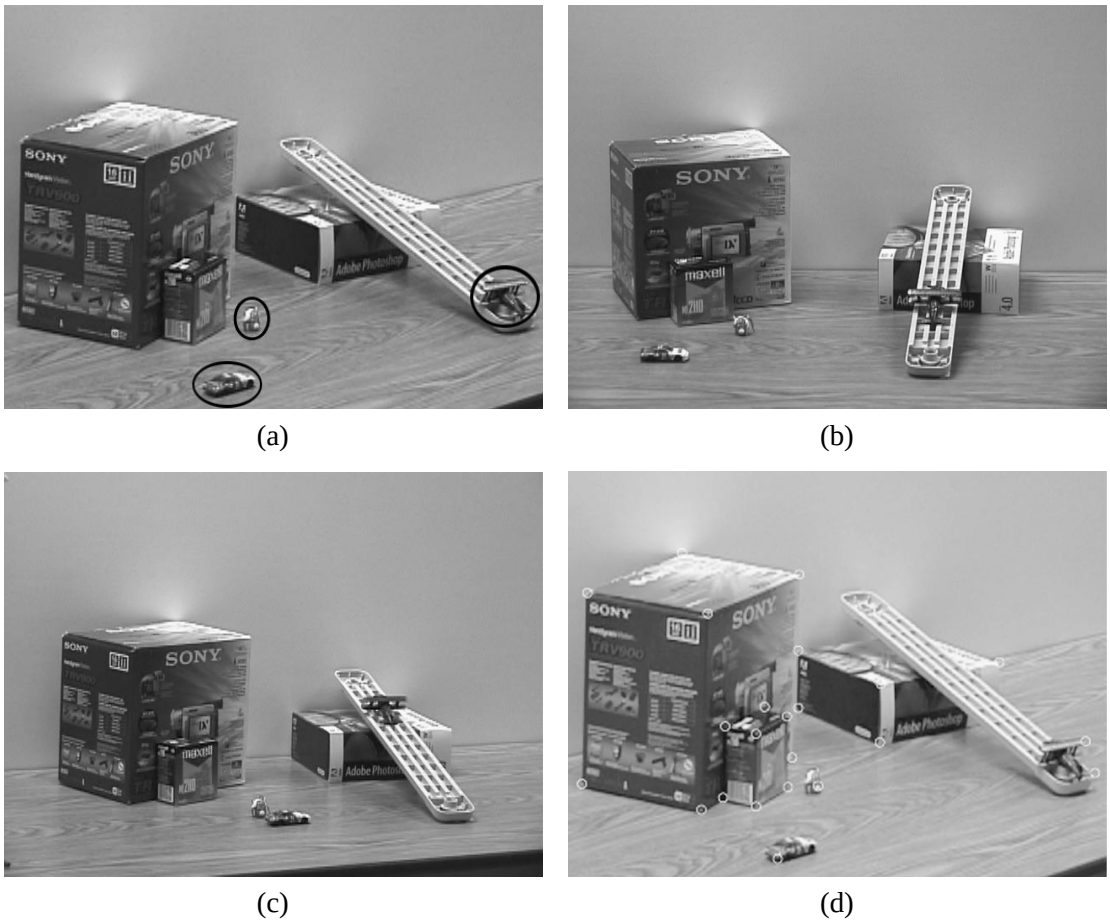
Figure 3.2: **Rank-5 case:** A scene with three motion trajectories which lie in a two dimensional space. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.

values is 0.1°.

3.5.2 Real example 1: Toy sequence

This sequence was taken of an indoor scene by a hand-held camera. Three objects, a car, a plane and a toy person, were moving linearly with constant speeds. The car and the person were moving on the floor, and the speed of the car was three times of the speed of the person. Their motion directions were perpendicular with each other. The plane was taking off on a slope and moved two times as fast as the car. The boxes represented the static scene. 24 images were taken. Three of them are shown in Figure 3.3(a)-(c). 23 feature points were manually selected and tracked, which are overlaid on the first image shown in Figure 3.3(d). We use the first 18 frames to perform the reconstruction. The shapes of the boxes, the starting positions of the moving objects and the motion velocities are recovered and demonstrated in Figure 3.4(a) (with texture mapping) and (b) (with wireframe), the motion trajectories are overlaid in the images. Figure 3.4 (c) show the recovered camera locations and orientations.

We assess the quality of the reconstruction by comparison with the ground truth. The angle between the motion direction of the car and that of the person is 90.15°, the ratio between the speeds is 3.05 which is close to the expected value 3.0. The ratio of the speed of the plane to that of the car is 1.97. The maximum distance between the positions of the recovered static points and the ground truth positions is 2mm. The recovered motion direction of the plane is 20° tilted upward from the floor, which is close to the expected value.



(a) (b)

(c) (d)

Figure 3.3: **Toy sequence input:** (a) 1st image, (b) 7th image, (c) 18th image of the indoor sequence, the moving objects are circled in the 1st image. (d) 1st image of the indoor sequence with the feature points overlaid.

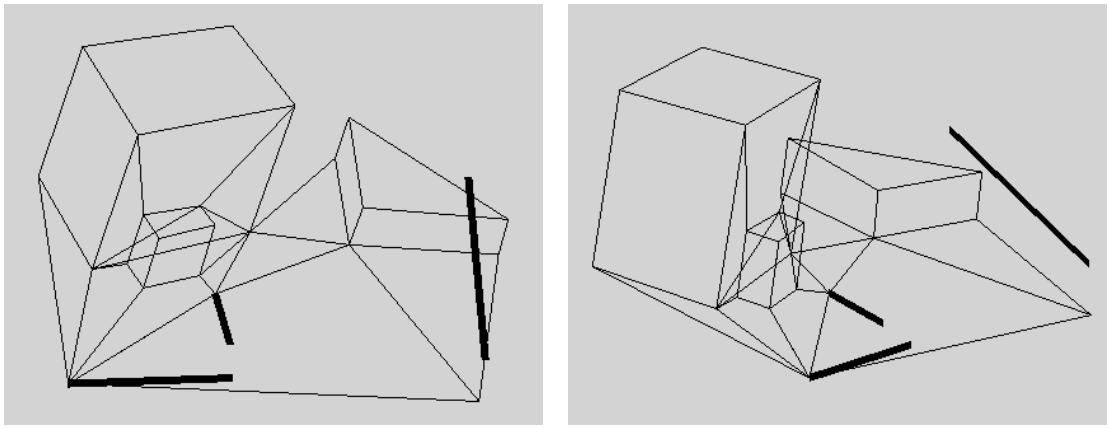
We project the motion trajectories back to the images and measure the discrepancies of the tracked objects and the back projections in the last 7 frames. The maximum discrepancy is 2 pixels.

3.5.3 Real example 2: Smith Hall sequence

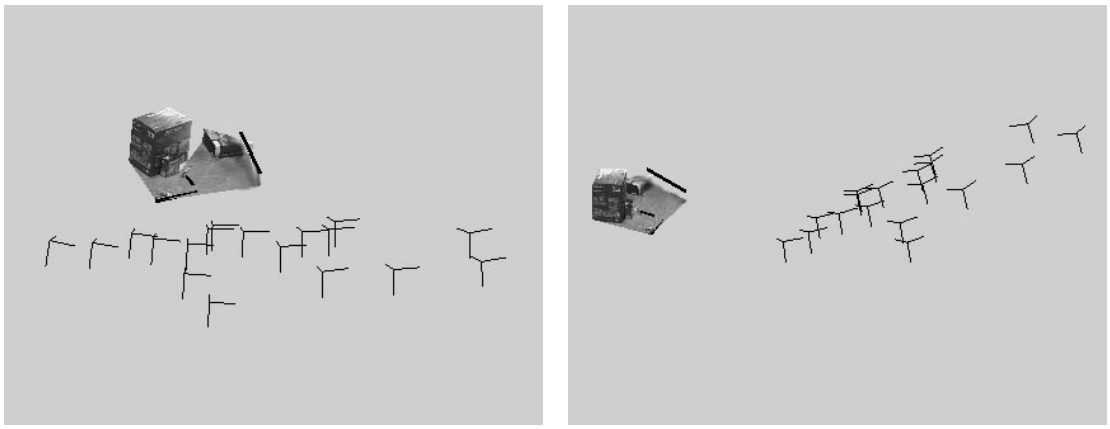
This sequence was taken by a small airplane flying over a scene with multiple moving cars. The first 80 frames of a 90 frame sequence are used, three of these frames are shown in Figure 3.5(a)-(c). 35 feature points were manually selected in the first frame corresponding to the buildings and the two moving cars as shown in Figure 3.5(d). These points were automatically tracked in the remaining frames. The method estimates the rank of $\hat{W}$ as 4 because the two cars were moving in opposite directions. Figures 3.6(a) and (b) show the recovered buildings as well as the motion trajectories. Since

the resolution of the input images is very low, the texture mapping is not very clear. Similar to the experiment in Section 3.5.2, we measure the discrepancies of the back projections of the cars and the tracked cars for the final 10 frames. The maximum discrepancies are 4 pixels for the white car and 5 pixels for the black car.

(a)

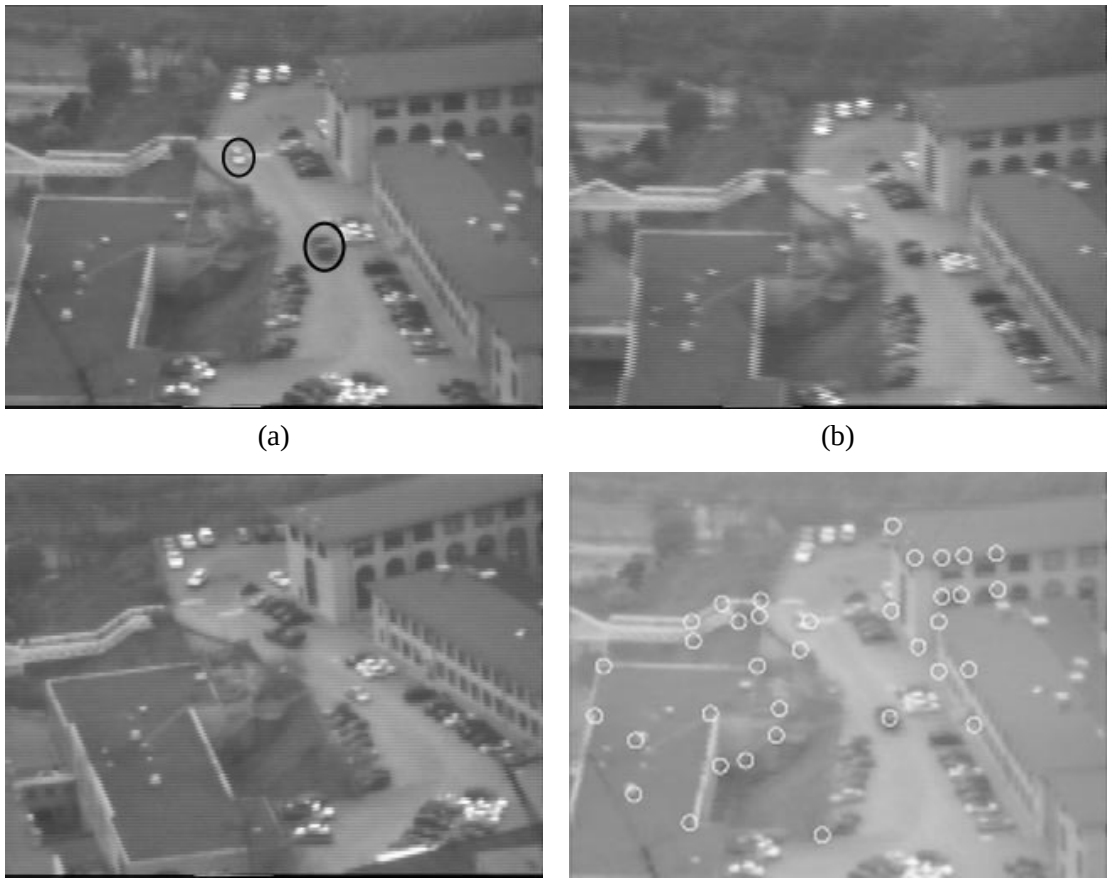


(b)



(c)

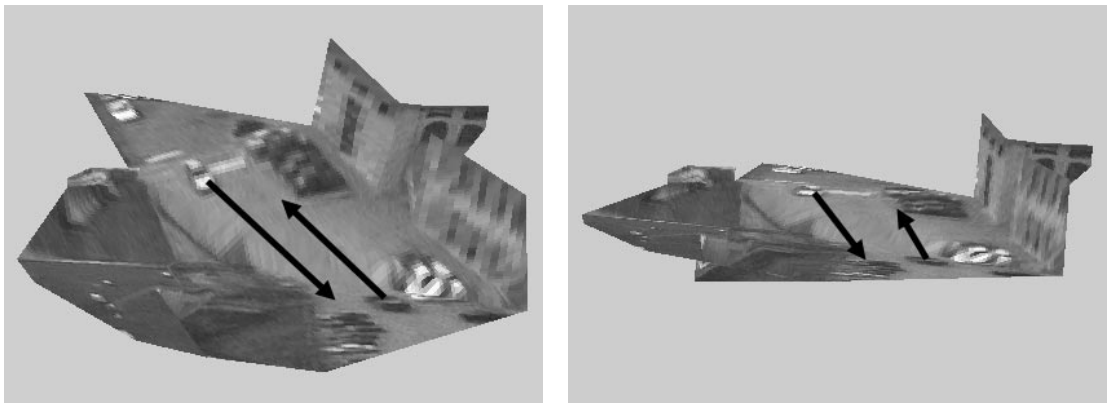
Figure 3.4: **Toy sequence results:** (a) Two views of the reconstruction with texture mapping, the black lines denote the recovered motion trajectories, the arrows show the motion directions. (b) Two views of the reconstruction with wireframe, the black lines denote the recovered motion trajectories. (c) Two views of the reconstruction, the 3-axis figures are the recovered cameras.



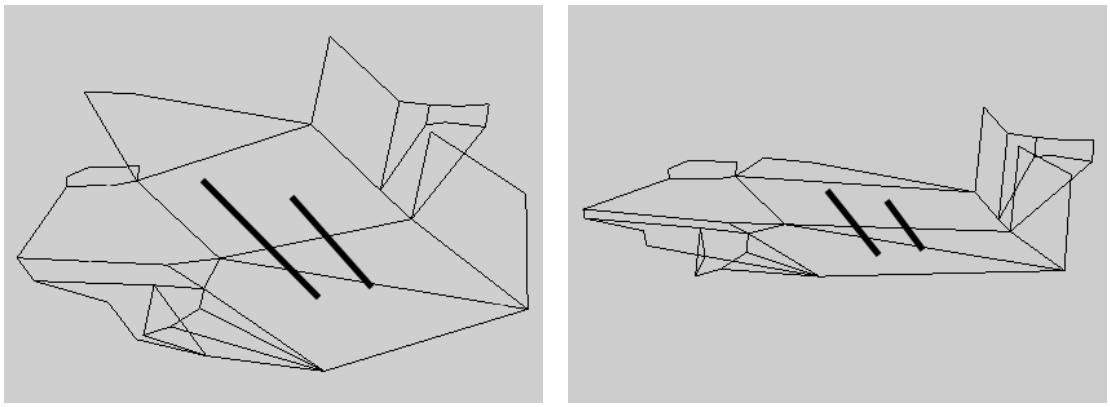
(a) (b)

(c) (d)

Figure 3.5: **Smith Hall sequence input:** (a) 1st image, (b) 33th image, (c) 80th image from the outdoor sequence, the moving objects are circled in the 1st image. (d) 1st image of the outdoor sequence with the feature points overlaid.



(a)



(b)

Figure 3.6: **Smith Hall sequence results:** (a) Two views of the reconstruction with texture mapping, the black lines denote the recovered motion trajectories, the arrows show the motion directions. (b) Two views of the reconstruction with wireframe, the black lines denote the recovered motion trajectories.

Chapter 4

Multiple Motion Scene Reconstruction with Uncalibrated Cameras

Chapter 3 presents a reconstruction method for multiple motion scenes under affine projections. It requires the cameras be intrinsically calibrated. In practice, many image sequences are taken with uncalibrated cameras. In this chapter we present the multiple motion scene reconstruction method from uncalibrated views [Han and Kanade, 2001]. Assuming that the objects are moving linearly with constant speeds, the method recovers the scene structure, the trajectories of the moving objects, the camera motion together with the camera intrinsic parameters. The method detects the moving objects automatically without prior motion segmentation.

4.1 Projective reconstruction

Given tracked feature points from uncalibrated views, we first perform a projective reconstruction. Perspective projection P_i, $i = 1 \cdots n$ and n is the number of frames, is represented by a 3×4 matrix,

$$P_i \sim K_i \left[\begin{array}{cc} R_i & \mathbf{t}_i \end{array} \right] \tag{4.1}$$

where

$$K_i = \left[\begin{array}{ccc} f_i & 0 & u_{0i} \\ 0 & \alpha_i f_i & v_{0i} \\ 0 & 0 & 1 \end{array} \right] \quad R_i = \left[\begin{array}{c} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \\ \mathbf{k}_i^{\mathrm{T}} \end{array} \right] \quad \mathbf{t}_i = \left[\begin{array}{c} t_{xi} \\ t_{yi} \\ t_{zi} \end{array} \right]$$

The upper triangular calibration matrix K_i encodes the intrinsic parameters of the ith camera: f_i represents the focal length, (u_{0i}, v_{0i}) is the principal point and α_i is the aspect ratio. We assume that the cameras have zero skews. R_i is the ith rotation matrix with $\mathbf{i}_i$, $\mathbf{j}_i$ and $\mathbf{k}_i$ denoting the rotation axes.

$\mathbf{t}_i$ is the ith translation vector. m feature points $\mathbf{x}_{ij}$, $j = 1 \cdots m$, are represented by homogeneous coordinates,

$$\mathbf{x}_{ij} \sim \begin{bmatrix} \mathbf{p}_{ij}^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{4.2}$$

where $\mathbf{p}_{ij}$ is defined as the unified representation of point (Chapter 3),

$$\mathbf{p}_{ij} = \mathbf{s}_j + i\mathbf{v}_j \tag{4.3}$$

where $\mathbf{s}_j$ is the point position at frame 0 and $\mathbf{v}_j$ is its motion velocity.

The image coordinates are represented by $(u_{ij} \;\; v_{ij})$ and the following hold,

$$\begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} \sim P_i \mathbf{x}_{ij} \quad \text{or} \quad \lambda_{ij} \begin{bmatrix} u_{ij} \\ v_{ij} \\ 1 \end{bmatrix} = P_i \mathbf{x}_{ij} \tag{4.4}$$

where λ_{ij} is a non-zero scale factor called projective depth. According to Equations (4.1) to (4.3),

$$\begin{aligned} P_i \mathbf{x}_{ij} &\sim K_i \left(R_i \, \mathbf{p}_{ij} + \mathbf{t}_i \right) \\ &= K_i \left(R_i \, \mathbf{s}_j + i \, R_i \, \mathbf{v}_j + \mathbf{t}_i \right) \\ &= K_i \begin{bmatrix} R_i & iR_i & \mathbf{t}_i \end{bmatrix} \begin{bmatrix} \mathbf{s}_j^{\mathrm{T}} & \mathbf{v}_j^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \\ &\sim \tilde{P}_i \tilde{\mathbf{x}}_j \end{aligned} \tag{4.5}$$

where

$$\tilde{P}_i \sim K_i \begin{bmatrix} R_i & iR_i & \mathbf{t}_i \end{bmatrix} \quad \tilde{\mathbf{x}}_j \sim \begin{bmatrix} \mathbf{s}_j^{\mathrm{T}} & \mathbf{v}_j^{\mathrm{T}} & 1 \end{bmatrix}^{\mathrm{T}} \tag{4.6}$$

$\tilde{P}_i$ is a 3×7 matrix which is the product of the ith calibration matrix and the unified motion matrix composed of the camera rotation, the scaled camera rotation by the frame number and the camera translation. $\tilde{\mathbf{x}}_j$ is a 7×1 vector which is the homogeneous representation of the unified scene structure including the initial point position and its velocity. The equivalent matrix form is,

$$W_s = \begin{bmatrix} \lambda_{11} \begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1m} \begin{bmatrix} u_{1m} \\ v_{1m} \\ 1 \end{bmatrix} \\ \vdots & \vdots & \vdots \\ \lambda_{n1} \begin{bmatrix} u_{n1} \\ v_{n1} \\ 1 \end{bmatrix} & \cdots & \lambda_{nm} \begin{bmatrix} u_{nm} \\ v_{nm} \\ 1 \end{bmatrix} \end{bmatrix} \tag{4.7}$$

$$= \begin{bmatrix} \tilde{P}_1 \\ \vdots \\ \tilde{P}_n \end{bmatrix} [\tilde{\mathbf{x}}_1 \ \cdots \ \tilde{\mathbf{x}}_m] = \tilde{P}\tilde{X} \tag{4.8}$$

where W_s is the *scaled measurement matrix*. We call the $3n \times 7$ matrix $\tilde{P}$ as motion matrix and the $7 \times m$ matrix $\tilde{X}$ as shape matrix. The constraint of the objects moving with constant velocities enables the unified representation of the motion matrix $\tilde{P}$ and the shape matrix $\tilde{X}$. They are both at most rank 7, therefore, the rank of the scaled measurement matrix W_s is at most **7** (instead of **rank 4** when the scene does not contain moving objects).

We apply the following bilinear factorization algorithm to get the projective reconstruction. The algorithm is similar to the iterative algorithm presented in Section 2.2 with the difference that a **rank 7** matrix factorization is performed at step 3. It iteratively applies factorization to the current scaled measurement matrix.

Iterative Projective Factorization Algorithm

1. Set $\lambda_{ij} = 1$, for $i = 1 \cdots n$ and $j = 1 \cdots m$;
2. Compute the current scaled measurement matrix W_s by Equation (4.7);
3. Perform **rank 7** factorization on W_s, generate the projective motion $\hat{P}$ and shape $\hat{X}$;
4. Reset $\lambda_{ij} = \hat{P}_i^{(3)}\hat{\mathbf{x}}_j$, where $\hat{P}_i^{(3)}$ denotes the third row of the projection matrix $\hat{P}_i$;
5. If λ_{ij}'s are the same as the previous iteration, stop; else go to step 2.

4.2 Euclidean reconstruction

The factorization of Equation (4.8) recovers the motion and shape up to a 7×7 linear projective transformation H,

$$W_s = \hat{P}\hat{X} = \hat{P}HH^{-1}\hat{X} = \tilde{P}\tilde{X} \tag{4.9}$$

where $\tilde{P} = \hat{P}H$ and $\tilde{X} = H^{-1}\hat{X}$. $\hat{P}$ and $\hat{X}$ are referred to as the projective motion and the projective shape. Any non-singular $\mathbf{7} \times \mathbf{7}$ matrix could be inserted between $\hat{P}$ and $\hat{X}$ to get another motion and shape pair. For the multiple motion scene reconstruction method presented in Chapter 3, the size of the linear transformation matrix is $\mathbf{6} \times \mathbf{6}$ which works on calibrated cameras and therefore does not encode the camera translation information. The goal of the Euclidean reconstruction is to impose metric constraints on the projective motion and shape in order to recover the linear transformation H,

from which we can simultaneously reconstruct the intrinsic parameters and the Euclidean motion and shape. This is the *normalization* process.

In this section we present the normalization algorithm for the case that only the focal lengths are unknown and varying. It is straightforward to derive the normalization algorithms for the other cases where more or all of the intrinsic parameters are unknown (except skews) following the same line of work presented in this chapter. However, due to the increased size of the transformation matrix, the normalization processes are less stable and practical (such as solving a linear system of more than 400 unknowns) to solve for the other intrinsic parameters as in Chapter 2. The normalization algorithm for the case with unknown focal lengths is linear.

When the focal lengths are the only unknown intrinsic parameters, we have,

$$u_{0i} = 0 \quad v_{0i} = 0 \quad \alpha_i = 1 \tag{4.10}$$

Therefore, according to Equation (4.6),

$$\tilde{P} = \begin{bmatrix} M & T \end{bmatrix} \tag{4.11}$$

where

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \\ \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{z1} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} & \mathbf{n}_{zn} \end{bmatrix}^{\mathrm{T}} \tag{4.12}$$

$$T = \begin{bmatrix} T_{x1} & T_{y1} & T_{z1} & \cdots & T_{xn} & T_{yn} & T_{zn} \end{bmatrix}^{\mathrm{T}} \tag{4.13}$$

and

$$\begin{array}{lll} \mathbf{m}_{xi} = \mu_i f_i \mathbf{i}_i & \mathbf{n}_{xi} = i\mu_i f_i \mathbf{i}_i & T_{xi} = \mu_i f_i t_{xi} \\ \mathbf{m}_{yi} = \mu_i f_i \mathbf{j}_i & \mathbf{n}_{yi} = i\mu_i f_i \mathbf{j}_i & T_{yi} = \mu_i f_i t_{yi} \\ \mathbf{m}_{zi} = \mu_i \mathbf{k}_i & \mathbf{n}_{zi} = i\mu_i \mathbf{k}_i & T_{zi} = \mu_i t_{zi} \end{array} \tag{4.14}$$

The shape matrix is represented by,

$$\tilde{X} \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{4.15}$$

where

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix} \tag{4.16}$$

and

$$\tilde{\mathbf{x}}_j = \begin{bmatrix} \nu_j \mathbf{s}_j^{\mathrm{T}} & \nu_j \mathbf{v}_j^{\mathrm{T}} & \nu_j \end{bmatrix}^{\mathrm{T}} \tag{4.17}$$

μ_i and ν_i are the scale factors for the homogeneous representations in Equation (4.6). The normalization process follows the same line of work as in Section 2.3. However, we are now working on a higher dimensional space of motion (4.12) and shape (4.16) because the scene contains multiple independently moving objects.

4.2.1 Moving world coordinate system location

As the points are either static or moving linearly with constant speeds, the center of gravity of all the feature points is also moving linearly with constant speed. So is the center of gravity of all the **scaled** points ($\nu_j \mathbf{p}_{ij}$). Here we transform the 3D representations to a moving world coordinate system with fixed orientation (such as being aligned with the first camera) and the origin at the center of gravity of all the scaled feature points. Therefore,

$$\sum_{j=1}^{m} \nu_j \mathbf{p}_{ij} = 0 \tag{4.18}$$

We get,

$$\begin{aligned}
\sum_{j=1}^{m} \lambda_{ij} u_{ij} &= \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + \mathbf{n}_{xi} \cdot \nu_j \mathbf{v}_j + \nu_j T_{xi}) \\
&= \sum_{j=1}^{m} (\mathbf{m}_{xi} \cdot \nu_j \mathbf{s}_j + i\mathbf{m}_{xi} \cdot \nu_j \mathbf{v}_j + \nu_j T_{xi}) \\
&= \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j (\mathbf{s}_j + i\mathbf{v}_j) + T_{xi} \sum_{j=1}^{m} \nu_j \\
&= \mathbf{m}_{xi} \cdot \sum_{j=1}^{m} \nu_j \mathbf{p}_{ij} + T_{xi} \sum_{j=1}^{m} \nu_j \\
&= T_{xi} \sum_{j=1}^{m} \nu_j
\end{aligned} \tag{4.19}$$

Similarly,

$$\sum_{j=1}^{m} \lambda_{ij} v_{ij} = T_{yi} \sum_{j=1}^{m} \nu_j \qquad \sum_{j=1}^{m} \lambda_{ij} = T_{zi} \sum_{j=1}^{m} \nu_j \tag{4.20}$$

Define the 7×7 projective transformation H as,

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{4.21}$$

where A is 7×6 and B is 7×1.

Since $\tilde{P} = \hat{P}H$,

$$\left[\begin{array}{cc} M & T \end{array} \right] = \hat{P} \left[\begin{array}{cc} A & B \end{array} \right] \tag{4.22}$$

according to Equation (4.11). We have,

$$T_{xi} = \hat{P}_{xi}B \quad T_{yi} = \hat{P}_{yi}B \quad T_{zi} = \hat{P}_{zi}B \tag{4.23}$$

From Equations (4.19) and (4.20),

$$\frac{T_{xi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} u_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \quad \frac{T_{yi}}{T_{zi}} = \frac{\sum_{j=1}^{m} \lambda_{ij} v_{ij}}{\sum_{j=1}^{m} \lambda_{ij}} \tag{4.24}$$

we set up $2n$ linear equations of the 7 unknown elements of the matrix B. Linear least squares solutions are then computed.

4.2.2 Normalization

We recover the 7×6 matrix A by observing that the rows of the matrix M consist of $\mathbf{m}_i$, which are the scaled rotation axes by μ_i and focal length f_i, and $\mathbf{n}_i$, which are the scaled $\mathbf{m}_i$ by frame number i (Equation (4.14)),

orthogonality of $\mathbf{m}_i$:

$$\begin{gathered} |\mathbf{m}_{xi}|^2 = |\mathbf{m}_{yi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{m}_{yi} = 0 \quad \mathbf{m}_{xi} \cdot \mathbf{m}_{zi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{m}_{zi} = 0 \end{gathered} \tag{4.25}$$

orthogonality of $\mathbf{n}_i$:

$$\begin{gathered} |\mathbf{n}_{xi}|^2 = |\mathbf{n}_{yi}|^2 \\ \mathbf{n}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{n}_{xi} \cdot \mathbf{n}_{zi} = 0 \quad \mathbf{n}_{yi} \cdot \mathbf{n}_{zi} = 0 \end{gathered} \tag{4.26}$$

relation of $\mathbf{m}_i$ and $\mathbf{n}_i$:

$$\begin{gathered} |\mathbf{n}_{xi}|^2 = i^2 |\mathbf{m}_{xi}|^2 \quad |\mathbf{n}_{yi}|^2 = i^2 |\mathbf{m}_{yi}|^2 \quad |\mathbf{n}_{zi}|^2 = i^2 |\mathbf{m}_{zi}|^2 \\ \mathbf{m}_{xi} \cdot \mathbf{n}_{yi} = 0 \quad \mathbf{m}_{xi} \cdot \mathbf{n}_{zi} = 0 \\ \mathbf{m}_{yi} \cdot \mathbf{n}_{xi} = 0 \quad \mathbf{m}_{yi} \cdot \mathbf{n}_{zi} = 0 \\ \mathbf{m}_{zi} \cdot \mathbf{n}_{xi} = 0 \quad \mathbf{m}_{zi} \cdot \mathbf{n}_{yi} = 0 \end{gathered} \tag{4.27}$$

The above equations impose linear constraints on the elements of MM^{T}. We add one more equation assuming $\mu_1 = 1$,

$$|\mathbf{m}_{z1}|^2 = 1 \tag{4.28}$$

Since $M = \hat{P}A$,

$$MM^{\mathrm{T}} = \hat{P}AA^{\mathrm{T}}\hat{P}^{\mathrm{T}} \tag{4.29}$$

these constraints are linear on the elements of the symmetric matrix $Q = AA^{\mathrm{T}}$. From Equations (4.25) to (4.27) we can see that these constraints are different from those under calibrated cameras (Equations (3.14) to (3.16)) since the motion matrix used here is composed of camera motion and **camera intrinsic parameters**.

Define,

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{4.30}$$

where A is 7×6 matrix and A_1, A_2 are both 7×3 matrices. We get,

$$\begin{aligned} \hat{P}A_1 &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \end{bmatrix}^{\mathrm{T}} \\ \hat{P}A_2 &= \begin{bmatrix} \mathbf{n}_{x1} & \mathbf{n}_{y1} & \mathbf{n}_{z1} & \cdots & \mathbf{n}_{xn} & \mathbf{n}_{yn} & \mathbf{n}_{zn} \end{bmatrix}^{\mathrm{T}} \\ &= N \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{4.31}$$

where

$$N = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & n & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & n \end{bmatrix} \tag{4.32}$$

according to Equation (4.14). Therefore,

$$\hat{P}A_2 = N\hat{P}A_1 \tag{4.33}$$

A_2 is over constrained given A_1 and $\hat{P}$,

$$A_2 = KA_1 \tag{4.34}$$

where

$$K = \hat{P}^{-1}N\hat{P} \tag{4.35}$$

and $\hat{P}^{-1}$ is the generalized inverse matrix which is $7 \times 3n$ and uniquely defined when $n \geq 3$.

Using similar derivations of Section 3.2, we see that Equation (4.25) imposes linear constraints on the 28 unknown elements of the 7×7 symmetric matrix $Q_1 = A_1 A_1^{\mathrm{T}}$, while Equation (4.26) imposes constraints on the 28 unknown elements of $Q_2 = A_2 A_2^{\mathrm{T}}$. From Equation (4.34) we have,

$$Q_2 = A_2 A_2^{\mathrm{T}} = K A_1 A_1^{\mathrm{T}} K^{\mathrm{T}} = K Q_1 K^{\mathrm{T}} \tag{4.36}$$

which translates the constraints on Q_2 to constraints on Q_1. Equation (4.27) imposes constraints on $Q_3 = A_2 A_1^{\mathrm{T}}$ which can also be translated into constraints on Q_1,

$$Q_3 = A_2 A_1^{\mathrm{T}} = K A_1 A_1^{\mathrm{T}} = K Q_1 \tag{4.37}$$

Therefore, each frame contributes 17 constraints (Equations (4.25) to (4.27)) on Q_1. In total, we get $17n+1$ linear equations on the 28 unknown elements of the symmetric matrix Q_1. Linear least squares solutions are computed. We then compute the matrix A_1 from Q_1 by rank 3 matrix decomposition and A_2 by Equation (4.34), so we recover the linear transformation A.

4.2.3 Camera calibration and scene reconstruction

Once the matrix A has been found, the projective transformation is $[A \;\; B]$. The shape matrix is computed as $\tilde{X} = H^{-1}\hat{X}$ and the motion matrix as $\tilde{P} = \hat{P}H$. We first compute the scale factors μ_i,

$$\mu_i = |\mathbf{m}_{zi}| \tag{4.38}$$

We then compute the focal lengths as,

$$f_i = \frac{|\mathbf{m}_{xi}| + |\mathbf{m}_{yi}|}{2\mu_i} \tag{4.39}$$

Therefore, the camera motion parameters are,

$$\begin{array}{lll} \mathbf{i}_i = \frac{\mathbf{m}_{xi}}{\mu_i f_i} & \mathbf{j}_i = \frac{\mathbf{m}_{yi}}{\mu_i f_i} & \mathbf{k}_i = \frac{\mathbf{m}_{zi}}{\mu_i} \\ t_{xi} = \frac{T_{xi}}{\mu_i f_i} & t_{yi} = \frac{T_{yi}}{\mu_i f_i} & t_{zi} = \frac{T_{zi}}{\mu_i} \end{array} \tag{4.40}$$

The shape matrix consists of the scene structure and the velocities represented in the moving world coordinate system. We need to transform the representation back to a fixed coordinate system with the origin at the center of gravity of all the points at frame 1 and detect the moving objects automatically. This process is same as described in Section 3.2.4.

4.2.4 Algorithm outline

We summarize the reconstruction method as follows:

1. Perform SVD on W_s, get the projective motion $\hat{P}$ and the projective shape $\hat{X}$;
2. Sum up each row of W_s and compute the ratios between them as in Equation (4.24);
3. Set up $2n$ linear equations of the 7 unknown elements of the matrix B based on the ratios from step 2 and compute B;
4. Set up $17n + 1$ linear equations of the 28 unknown elements of the symmetric matrix Q_1 by imposing constraints in Equations (4.25) to (4.27);
5. Factor Q_1 to get A_1 from $Q_1 = A_1 A_1^{\mathrm{T}}$;
6. Compute A_2 from $A_2 = K A_1$;
7. Combine A_1 and A_2 to generate the linear transformation matrix $A = [A_1 \ A_2]$;
8. Put the matrices A and B together and get the projective transformation $H = [A \ \ B]$;
9. Recover the shape matrix using $\tilde{X} = H^{-1}\hat{X}$ and motion matrix using $\tilde{P} = \hat{P}H$;
10. Recover the focal lengths, the camera rotation axes and the translation vectors according to Equations (4.39) and (4.40).
11. Detect the moving objects, reconstruct the scene structure and the trajectories of the moving objects as presented in Section 3.2.4.

4.3 Experiments

In this section we present the experimental results on synthetic and real images. The first set of experiments use synthetic images to evaluate the method quantitatively. The second experiment is conducted on a real image sequence taken by a hand-held camera of an indoor scene, and the reconstruction results are compared with the ground truth values.

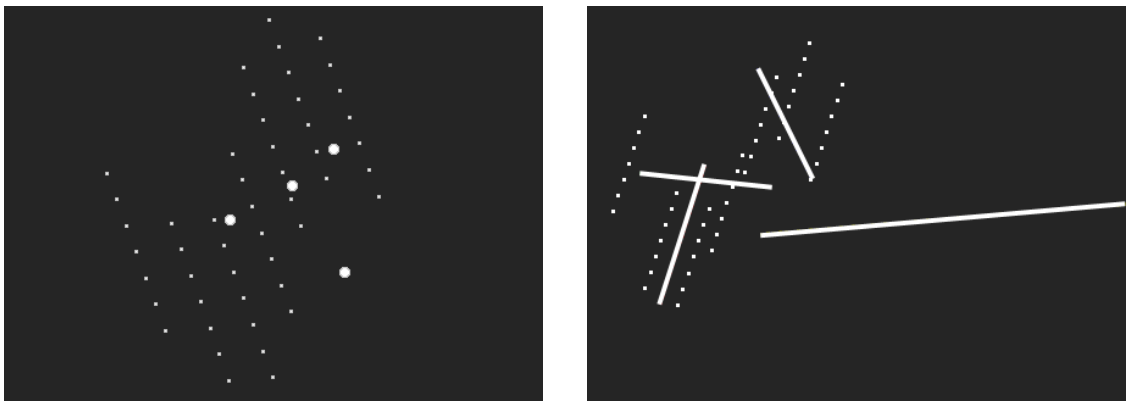
4.3.1 Synthetic examples

We generate 100 image sequences of the scene with 8 to 49 static feature points and 3 to 8 points moving in random directions. The frame number is 4 to 60. The shape of the static scene is a sweep of the sin curve in 3D space. The camera is rotating randomly through 30 to 60 degrees for each or any of roll, pitch and yaw. The distance between the camera and the center of gravity of all the static points is varied from 4 to 20 times the object size. We add 2 pixel standard noise to the feature locations from 640×480 images.

Figure 4.1 illustrates the case where 4 objects are moving randomly in 3D space. The method automatically detects the number of the moving objects as 4, reconstructs the static scene and the initial positions of the 4 moving objects, as shown in Figure 4.1(a). Figure 4.1(b) shows the trajectories of the moving objects with the static scene. There are 49 points from the static scene and 60 frames are taken.

Figure 4.2 plots the focal lengths recovered by the method and their ground truth values. The maximum error is 7.2% of the true value.

We also apply the multiple motion scene reconstruction method for weak perspective cameras to the same sequence using the true values of the focal lengths. The results are shown in Figure 4.3. It is easy to see that the reconstruction results have distortions which are caused by the approximation of perspective cameras with weak perspective cameras.



(a) (b)

Figure 4.1: **Synthetic sequence:** Reconstruction of a scene with four moving objects by the uncalibrated multiple motion scene reconstruction method. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories.

To evaluate the quality of the reconstruction method, we measure the reconstruction error by comparison with the ground truth. Since the reconstruction from monocular image sequences is up to a scale, we assume that the size of the static shape is 1. The maximum distance between the recovered static points and their true values is 3.2%, the maximum error of the reconstructed initial positions of the moving objects is 4.1% and the velocity error is less than 1.9%. The maximum distance between the recovered camera locations and the ground truth values is 5.4% and the maximum angle between the recovered camera orientations and the known values is 0.12°. The maximum reconstruction error of the focal lengths is 8.11% of the ground truth values.

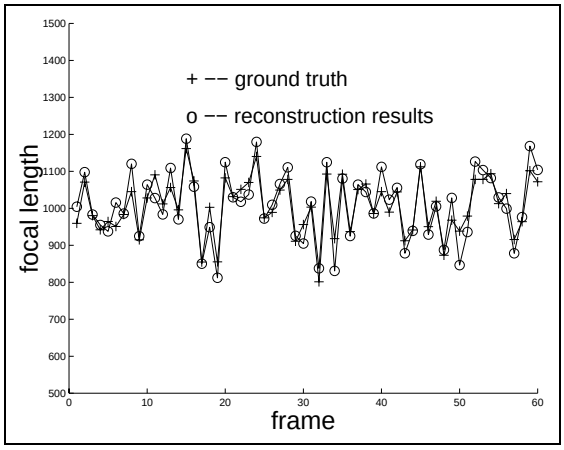
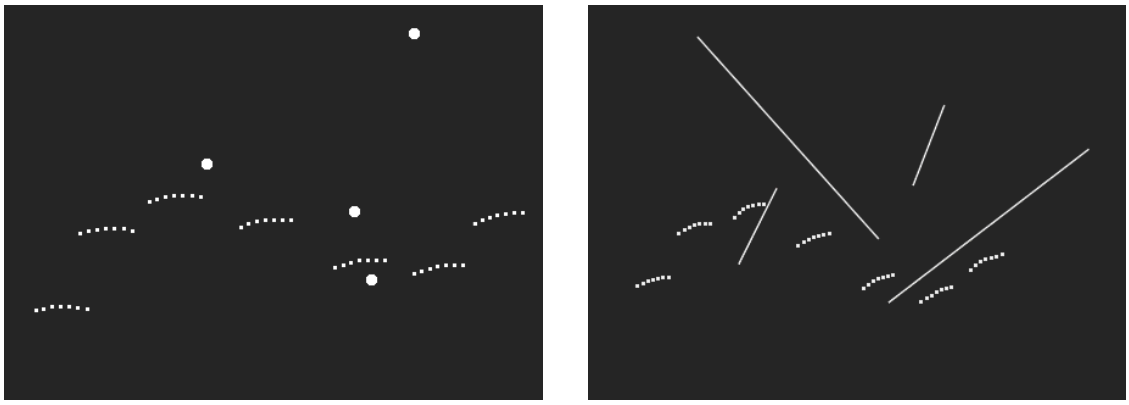


Figure 4.2: **Synthetic sequence:** Comparison of the focal lengths recovered by the uncalibrated multiple motion scene reconstruction method and their ground truth values for the synthetic sequence. The maximum error is 7.2% of the true value.



(a) (b)

Figure 4.3: **Synthetic sequence :** Reconstruction of a scene with four moving objects by the multiple motion scene weak perspective method. (a) The reconstructed scene structure and the initial positions of the moving objects. (b) The reconstructed scene and the motion trajectories. The distortions are caused by the approximation of perspective cameras with weak perspective cameras.

4.3.2 Real example

This sequence was taken by a hand-held camera. There were three objects moving in the scene, including a toy car, a toy bird and a toy person. The objects were moving linearly with constant speeds. The car and the person were moving on the table. The speed of the car was $3.5cm$ per frame and the speed of the person was $2.5cm$ per frame. The bird was climbing the pole and moved $3.0cm$ per frame. The books and the box represented the static scene. The camera was zoomed out at the beginning and gradually zoomed in as it moved around the scene. The focal length was changed every two frames. 10 images were taken. Three of them are shown in Figure 4.4(a)-(c). 29 feature points were manually selected and tracked as shown in Figure 4.4(d). Each moving object had one feature point selected.

The shapes of the books and the box, the starting positions of the toys and the motion velocities are recovered and demonstrated in Figure 4.5(a), the motion trajectories are overlaid in the images. Figure 4.5 (b) shows the recovered camera locations and orientations with the scene reconstruction. Figure 4.6 plots the recovered focal lengths, which shows that the focal lengths are changing with the camera motion as we expected. The largest focal length almost doubles the smallest one, which is correct for the $2\times$ optical lens.

We assess the quality of the reconstruction by comparison with the ground truth. The ratio between the speeds of the moving toys are $2.5 : 3.77 : 2.91$ which are close to the expected value $2.5 : 3.5 : 3.0$. The maximum distance between the positions of the recovered static points and the ground truth positions is 5mm. The angle between the recovered motion direction of the bird and the floor is 91.2°, which is close to the expected value.

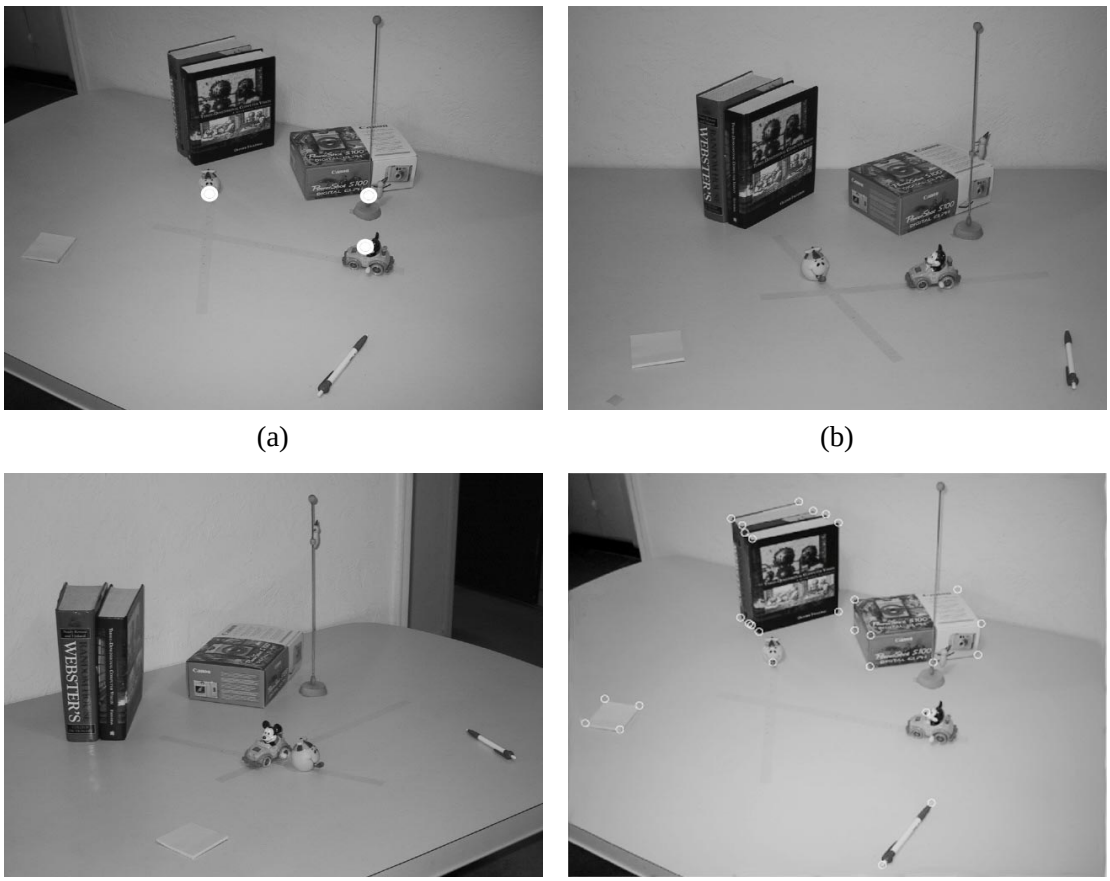
4.4 Degenerate cases

The method described in Sections 4.1 and 4.2 solves the full rank case where the static structure and the motion space of the objects are both rank 3. In other words, the scene is three dimensional and the velocities of the moving objects span a three dimensional space. Degenerate cases, however, exist because either or both of shape and motion spaces are degenerate. The shape space is degenerate, for example, when all the points lie in a plane. The motion space of the moving objects is degenerate, when:

1. There is no moving object in the scene.

2. There is one moving object or multiple objects moving in the same and/or the opposite direction (not necessarily the same 3D line).

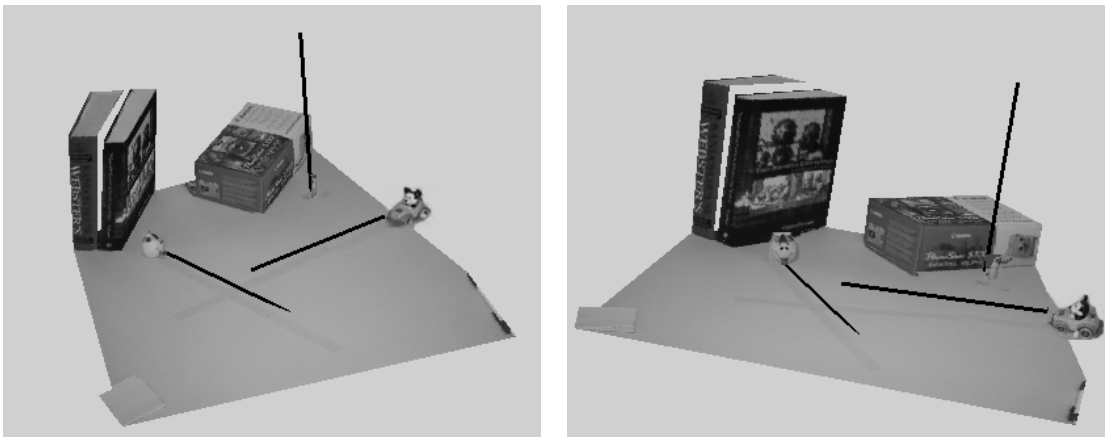
3. The velocities of the objects lie in a two dimensional space (not necessarily the same 3D plane).



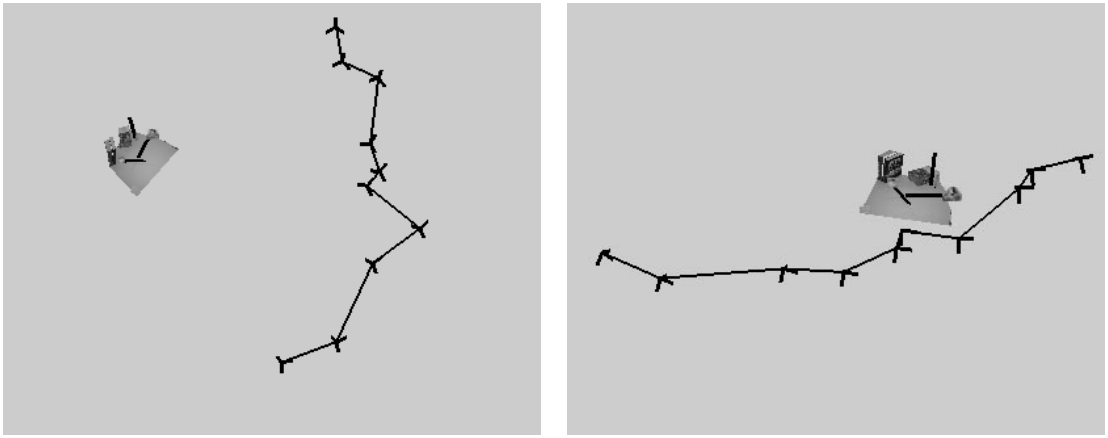
(a) (b)

(c) (d)

Figure 4.4: **Real sequence input:** (a) 1st image, (b) 5th image, (c) 10th image of the indoor sequence. The white circles in the 1st image show the feature points selected on the moving objects. (d) 1st image of the sequence with the feature points overlaid.



(a)



(b)

Figure 4.5: **Real sequence results:** (a) Two views of the scene reconstruction with texture mapping, the black lines denote the recovered motion trajectories. (b) Two views of the scene reconstruction and the camera positions/orientations, the 3-axis figures are the recovered cameras.

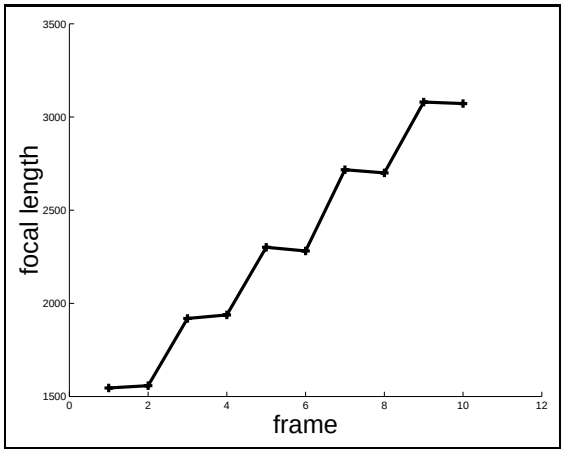


Figure 4.6: **Real sequence:** Focal lengths of the real sequence recovered by the uncalibrated multiple motion scene reconstruction method. The recovered values change every two frames as expected.

When cameras are intrinsically calibrated, there are solutions to these degenerate cases as shown in Section 3.3. Following the same line of work, we design the reconstruction algorithms for the degenerate cases with uncalibrated cameras. However, the rank of the measurement matrix can not be used as a clue about which case is the best approximation under perspective projections. The measurement matrix is always full rank. Therefore, we assume that the rank approximation information is given though there is no requirement for prior motion segmentation and the rank does not depend on how many objects are moving. In this section we describe the reconstruction algorithms for the case 2 and case 3 mentioned above. They are referred to as rank-4 case and rank-5 case, respectively, as in Section 3.3.

4.4.1 Rank-4 case

When only one moving object is in the scene, or when all the moving objects travel in the same or the opposite direction, the motion space is one dimensional. We refer to this case as rank-4 case.

Projective reconstruction

We align the $\mathbf{x}$ direction of the world coordinate system with the motion direction. Therefore, the motion and shape matrices are,

$$M = \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \\ n_{x1_x} & n_{y1_x} & n_{z1_x} & \cdots & n_{xn_x} & n_{yn_x} & n_{zn_x} \end{bmatrix}^{\mathrm{T}}$$

$$S = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \end{bmatrix} \tag{4.41}$$

where n_{xi_x}, n_{yi_x} and n_{zi_x} represent the x-elements of the vectors $\mathbf{n}_{xi}$, $\mathbf{n}_{yi}$ and $\mathbf{n}_{zi}$ respectively, v_{xj} denotes the x-element of the velocity of the jth feature point. $\mathbf{m}$ and $\mathbf{n}$ vectors are defined in Equation (4.14). We have,

$$\tilde{P} = \begin{bmatrix} M & T \end{bmatrix} \quad \tilde{X} \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{4.42}$$

and T is defined as in Equation (4.13). Therefore, $\tilde{P}$ is a $3n \times 5$ matrix and $\tilde{X}$ is a $5 \times m$ matrix. Following the same derivation as in Section 4.1, the rank of the scaled measurement matrix is 5. We apply a similar bilinear factorization algorithm with the only difference that we perform a **rank 5** matrix factorization on W_s instead of rank 7 as in Section 4.1.

Euclidean reconstruction

Define the 5×5 projective transformation H as,

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{4.43}$$

where A is 5×4 and B is 5×1. Similar derivations apply to the computation of the 5 unknown elements of the matrix B as in Equation (4.24). Similarly, we have

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{4.44}$$

where A_1 is 5×3, A_2 is 5×1 and,

$$A_2 = K(A_1)_1 \tag{4.45}$$

where $(A_1)_1$ is the first column of A_1 and K is defined in Equation(4.35). Since the matrix M consists of the rotation axes and only the x-elements of the scaled rotation axes, the constraints in Equations (4.26) and (4.27) cannot be represented as linear constraints on the elements of MM^{T}. However, the constraints in Equation (4.25) still hold and provide full rank linear equations on the 15 unknown elements of the symmetric 5×5 matrix $Q_1 = A_1 A_1^{\mathrm{T}}$. Least squares solutions are computed. We then compute A_1 by rank 3 matrix decomposition of Q_1. This decomposition is up to a three dimensional rotation R which is constrained to make the $\mathbf{x}$ direction of the world coordinate system as the motion direction. The matrix R is determined by aligning the matrix $\hat{M}KA_1$ with the matrix $N\hat{M}A_1$.

Therefore, the linear transformation A is,

$$A = \begin{bmatrix} A_1 R & K(A_1 R)_1 \end{bmatrix} \tag{4.46}$$

where $()_1$ denotes the first column of the matrix. We apply a derivation similar to the one in Section 4.2.3 to recover the camera focal lengths, the camera motion and the scene structure.

4.4.2 Rank-5 case

When the velocities of all the moving objects lie in a two dimensional space, the motion space is two dimensional. We refer to this case as rank-5 case.

Projective reconstruction

We assume that the $\mathbf{x}-\mathbf{y}$ plane of the world coordinate system is aligned with the two dimensional motion space. Therefore, the motion and shape matrices are,

$$\begin{aligned} M &= \begin{bmatrix} \mathbf{m}_{x1} & \mathbf{m}_{y1} & \mathbf{m}_{z1} & \cdots & \mathbf{m}_{xn} & \mathbf{m}_{yn} & \mathbf{m}_{zn} \\ n_{x1_x} & n_{y1_x} & n_{z1_x} & \cdots & n_{xn_x} & n_{yn_x} & n_{zn_x} \\ n_{x1_y} & n_{y1_y} & n_{z1_y} & \cdots & n_{xn_y} & n_{yn_y} & n_{zn_y} \end{bmatrix}^{\mathrm{T}} \\ S &= \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 & \cdots & \mathbf{s}_m \\ v_{x1} & v_{x2} & \cdots & v_{xm} \\ v_{y1} & v_{y2} & \cdots & v_{ym} \end{bmatrix} \end{aligned} \tag{4.47}$$

where n_{xi_x}, n_{yi_x} and n_{zi_x} represent the x-elements of the vectors $\mathbf{n}_{xi}$, $\mathbf{n}_{yi}$ and $\mathbf{n}_{zi}$ respectively, n_{xi_y}, n_{yi_y} and n_{zi_y} are their y-elements, v_{xj} denotes the x-element of the velocity of the jth feature point and v_{yj} is its y-element. $\mathbf{m}$ and $\mathbf{n}$ vectors are defined in Equation (4.14). We have,

$$\tilde{P} = \begin{bmatrix} M & T \end{bmatrix} \qquad \tilde{X} \sim \begin{bmatrix} S \\ \mathbf{1} \end{bmatrix} \tag{4.48}$$

and T is defined as in Equation (4.13). Therefore, $\tilde{P}$ is a $3n \times 6$ matrix and $\tilde{X}$ is a $6 \times m$ matrix. Following the same derivation as in Section 4.1, the rank of the scaled measurement matrix is 6. We apply a similar bilinear factorization algorithm with the only difference that we perform a **rank 6** matrix factorization on W_s.

Euclidean reconstruction

Define the 6×6 projective transformation H as,

$$H = \begin{bmatrix} A & B \end{bmatrix} \tag{4.49}$$

where A is 6×5 and B is 6×1. Similar derivations apply to the computation of the 6 unknown elements of the matrix B as in Equation (4.24). Define,

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{4.50}$$

where A_1 is 6×3, A_2 is 6×2 and,

$$A_2 = K(A_1)_{12} \tag{4.51}$$

where $(A_1)_{12}$ denotes the first two columns of A_1 and K is defined in Equation (4.35). Here only the constraints in Equation (4.25) can be represented as linear constraints on the elements of $Q_1 = A_1 A_1^{\mathrm{T}}$. In this case the constraints are not sufficient to solve for the 21 unknown elements of the symmetric 6×6 matrix Q_1 linearly.

The constraints in Equations (4.26) and (4.27) can be represented as constraints on the elements of Q_1 and the six elements of the third column of A_1, which is a 6×1 vector denoted by $\mathbf{c}$, as in Section 3.3.3. Therefore, we get linear equations of the 21 unknown elements of Q_1 and the 21 unknown elements of $\mathbf{c}\mathbf{c}^{\mathrm{T}}$. Since these equations cannot provide full rank constraints on the 42 unknowns, there is no linear solutions of Q_1 and $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ directly. However, the constraints are full rank on the elements of Q_1 if $\mathbf{c}\mathbf{c}^{\mathrm{T}}$ is given. That is, if $\mathbf{c}$ can be computed, we can get a linear solution of Q_1. In this way we change the problem to a small scale nonlinear optimization on the 6 elements of $\mathbf{c}$. Once the vector $\mathbf{c}$ is computed, the matrix Q_1 is computed by least squares solutions. A_1 is then calculated from Q_1.

Same to the rank-4 case, we need to align the $\mathbf{x} - \mathbf{y}$ plane of the world coordinate system with the two dimensional motion space. The matrix R is also determined by aligning the matrix $\hat{M} K A_1$ with the matrix $N \hat{M} A_1$. The alignment problem is solved by the least eigenvalue method.

Therefore, the linear transformation A is,

$$A = \begin{bmatrix} A_1 R & K(A_1 R)_{12} \end{bmatrix} \tag{4.52}$$

We apply a derivation similar to the one in Section 4.2.3 to recover the camera intrinsic parameters, the camera motion and the scene structure.

Chapter 5

Reconstruction Analysis

In this chapter we address two important issues of reconstruction methods: minimum data requirement and gauge selection. Reconstruction reliability is related to the minimum number of views and features required for reconstruction. We describe the theoretical analysis and the empirical results of the minimum data requirement of the reconstruction methods presented in this thesis. Gauge selection is the process of specifying the coordinate frame and representing the recovered geometry in the chosen frame. We analyze the gauge selection techniques used in the reconstruction methods described in this thesis and show that the techniques make the reconstruction methods reliable.

5.1 Minimum data requirement

The main advantage of the factorization-based methods is using the heavily redundant information from multiple image features and views. However, it is equally important to compute the minimum data requirement of these methods in order to analyze the practicality and reliability of the methods. In this section we discuss the minimum number of views and image features required by the reconstruction methods presented in Chapters 2, 3 and 4.

The low bound of data requirement is determined by the number of degrees of freedom of the reconstruction and the number of constraints given by each feature in each view, which is presented in Section 5.1.1. The minimum number of views and/or features is also constrained by the solution process. Section 5.1.2 lists the number of variables and the number of corresponding equations used in the reconstruction processes. These two computations of the minimum data only provide necessary conditions to carry out the reconstruction. There is no guarantee that the reconstruction results are reasonably accurate and stable with the theoretical results of the minimum data, especially for the non-linear optimization methods. In Section 5.1.3 we describe the empirical results of the minimum data required by the reconstruction methods.

5.1.1 Counting the arguments

In this section we specify the number of views and image features required to carry out the reconstruction. This analysis is related to counting the number of degrees of freedom of the reconstruction and the number of constraints give by each feature in each view. Hence comes the title "counting the arguments". Hartley and Zisserman's discussion about the minimum data required for tensor computation [Hartley and Zisserman, 2000] and Long Quan's analysis about the minimum number of line segments for the factorization method from line correspondences [Quan and Kanade, 1997] are two examples of this line of work.

Number of constraints

The input for the reconstruction methods presented in this thesis are the feature correspondences. Each feature point has two image coordinates in each view. Therefore, the number of constraints given by all the correspondences is $2nm$, where n is the number of views (or frames) and m is the number of feature points. Table 5.1.1 describes the analysis results about the minimum data requirement by counting the arguments. The column with the title **"Known #"** lists the numbers of constraints provided by the feature points for different reconstruction methods, all of which are $2nm$.

Number of degrees of freedom

The number of degrees of freedom of reconstruction depends on the size of the space composed of all the possible reconstructions. The output of the reconstruction methods consists of the scene structure, which includes the trajectories of the moving objects for multiple motion scenes, the camera motion and the camera intrinsic parameters for uncalibrated views. The number of degrees of freedom of each of the reconstruction output is summarized as follows. Reconstruction from monocular image sequences is up to a rigidity transformation, therefore, the total number of degrees of freedom should be subtracted by the number of ambiguities caused by the transformation. In Table 5.1.1, the column of **"Unknown #"** presents the number of degrees of freedom for each reconstruction method.

- Scene structure

  We refer the scenes without moving objects as static scenes and the scenes containing moving objects as multiple motion scenes. The number of degrees of freedom for static scenes is $3m$, where m is the number of feature points, since each feature point has three coordinates $(x,\ y,\ z)$ to represent its 3D position. For multiple motion scenes each feature point has three coordinates $(x,\ y,\ z)$ to represent its 3D position (the initial position) and three coordinates $(v_x,\ v_y,\ v_z)$ to denote its velocity (static points have zero velocities). We use the velocities of the feature points to distinguish moving features from static ones because we do not require prior motion

segmentation. Therefore, the number of degrees of freedom for multiple motion scenes is $6m$ for full rank case. Following the similar derivation, the number of degrees of freedom is $4m$ for rank-4 case (three coordinates for the initial position and one coordinate for the velocity since the motion space is one-dimensional) and $5m$ for rank-5 case (three coordinates for the initial position and two for the velocity as the motion space is two-dimensional), respectively.

- Camera motion

  The camera motion is determined by its rotation and translation. Weak perspective and perspective cameras have 6 degrees of freedom: 3 for rotation and 3 for translation. There is no information about the translations of orthographic cameras along their optical axes, therefore, the number of degrees of freedom for orthographic cameras is 5: 3 for rotation and 2 for translation. In total, the number of degrees of freedom for orthographic cameras is $5n$, where n is the number of views, and the number of degrees of freedom for weak perspective and perspective (calibrated or uncalibrated) cameras is $6n$.

- Camera intrinsic calibration

  When the cameras are not intrinsically calibrated, the number of degrees of freedom of the reconstruction is increased by the number of the unknown intrinsic parameters. For the static scene reconstruction method dealing with case 1, where the focal lengths are unknown and varying, the number of degrees of freedom for the intrinsic parameters is n. For case 2, where the focal lengths and the constant principal point are unknown, the number of degrees of freedom is $n+2$. For case 3, where the focal lengths, the principal points and the aspect ratios are all unknown and varying, the number of degrees of freedom for the camera intrinsic parameters is $4n$. The uncalibrated Euclidean reconstruction method for multiple motion scenes handles the case where the focal lengths are the only unknown intrinsic parameters, therefore, it has n degrees of freedom for the camera intrinsic parameters.

- Ambiguity

  Euclidean reconstruction from monocular image sequences is up to a rigidity transformation which has 6 degrees of freedom: 3 for rotation and 3 for translation. This number should be subtracted from the total number of degrees of freedom of the reconstruction. There is one more ambiguity for the reconstructions under weak perspective and perspective (calibrated or uncalibrated) cameras: the scale of the reconstruction. Therefore, 6 degrees of freedom is subtracted from the total number of degrees of freedom for orthographic cameras and 7 is subtracted for weak perspective and perspective cameras.

Methods			Known #	Unknown #	Minimum data
Static scene	Orthographic		$2nm$	$5n+3m-6$	$n=2 \quad m=4$
	Weak perspective		$2nm$	$6n+3m-7$	$n=3 \quad m=4$
	Uncalibrated perspective	Case 1	$2nm$	$7n+3m-7$	$n=3 \quad m=5$
		Case 2	$2nm$	$7n+3m-5$	$n=3 \quad m=6$
		Case 3	$2nm$	$10n+3m-7$	$n=3 \quad m=8$
Multiple motion scene	Orthographic	Full rank	$2nm$	$5n+6m-6$	$n=4 \quad m=7$
		Rank-4	$2nm$	$5n+4m-4$	$n=4 \quad m=4$
		Rank-5	$2nm$	$5n+5m-5$	$n=4 \quad m=5$
	Weak perspective	Full rank	$2nm$	$6n+6m-7$	$n=5 \quad m=6$
		Rank-4	$2nm$	$6n+4m-5$	$n=4 \quad m=5$
		Rank-5	$2nm$	$6n+5m-6$	$n=4 \quad m=6$
	Perspective	Full rank	$2nm$	$6n+6m-7$	$n=5 \quad m=6$
		Rank-4	$2nm$	$6n+4m-5$	$n=4 \quad m=5$
		Rank-5	$2nm$	$6n+5m-6$	$n=4 \quad m=6$
	Uncalibrated perspective	Full rank	$2nm$	$7n+6m-7$	$n=5 \quad m=7$
		Rank-4	$2nm$	$7n+4m-5$	$n=4 \quad m=6$
		Rank-5	$2nm$	$7n+5m-6$	$n=3 \quad m=15$

Table 5.1: **Minimum data requirement:** Counting the arguments. n denotes the number of views and m denotes the number of feature points.

Minimum number of views and features

We list the minimum number of views and feature points required by the reconstruction methods through counting the arguments in the column **"Minimum data"** of Table 5.1.1. We take the uncalibrated reconstruction method dealing with case 1 for static scenes as an example to illustrate how we compute the minimum data requirement.

The constraint is,

$$2nm \geq 7n + 3m - 7 \tag{5.1}$$

where n is the number of views and m is the number of feature points. Compute the positive integer solutions of n and m,

$$n = 2 \quad m = 7 \quad \text{OR} \quad n = 3 \quad m = 5 \tag{5.2}$$

In most cases the solutions are not unique. We choose the solution to be listed in the table according to two principles. The first one is that we prefer the solution which is comparable with the constraints of "analyzing the solution" (refer to Section 5.1.2). These two constraints both give the necessary requirement of the minimum data. Their intersection provides the low bound of the requirement. Therefore, we choose the comparable solution for easier computation of the intersection. The second principle is that we are in favor of less views. In practice, it is easier to get more feature points than to get more

views.

Another thing to point out is that this table only gives the low bound of the requirement. There is no guarantee that the reconstruction exits with the minimum data. For example, by counting the arguments, 2 views and 4 features are the minimum data required for orthographic reconstruction. However, there is actually the *bas relief* ambiguity that we cannot get a unique Euclidean reconstruction from any two orthographic views [Kahl and Triggs, 1999].

5.1.2 Analyzing the solution

The minimum data requirement is also determined by the solution process. The reconstruction methods presented in the thesis are based on linear and bilinear subspace constraints, that is, the solution process includes solving linear and bilinear equations. Therefore, the solution process requires the number of equations be larger than the number of variables. This is the constraint given by analyzing the solution. In Table 5.1.2 the numbers of equations for different reconstruction methods are listed in the column of **"Equation #"** and the numbers of variables are in the column of **"Variable #**". The last column **"Minimum data"** presents the minimum data required by analyzing the solution process. We take the orthographic reconstruction method for multiple motion scenes (full rank case) and the uncalibrated reconstruction method for static scenes (case 2) as examples of the reconstruction methods with calibrated and uncalibrated cameras, respectively, to illustrate how we set up Table 5.1.2.

Reconstruction with calibrated cameras

The reconstruction process with calibrated cameras is composed of matrix decomposition (SVD), normalization and recovery of shape and motion. The first two steps, decomposition and normalization, provide constraints on the minimum data requirement while the last step is directly derived as long as the first two steps succeed.

Take the full rank case of the orthographic reconstruction method for multiple motion scenes as example. We first fix the moving world coordinate system and compute the translation vector as the mean of the measurement matrix W. Then we perform a rank 6 SVD on the new measurement matrix $\hat{W}$ which is generated by subtracting the translation vector from W. The rank 6 SVD decomposes the matrix $\hat{W}$ into the product of a $2n \times 6$ matrix and a $6 \times m$ matrix. The decomposition is up to a 6×6 linear transformation. Therefore, the total number of variables for SVD is $12n + 6m - 35$. Since the size of $\hat{W}$ is $2n \times m$, we get constraints,

$$2nm \geq 12n + 6m - 35 \tag{5.3}$$

Methods				Equation #	Variable #	Minimum data
Static scene	Orthographic	SVD		$2nm$	$6n+3m-8$	$n=2 \quad m=4$
		Normalization		$3n$	6	
	Weak perspective	SVD		$2nm$	$6n+3m-8$	$n=3 \quad m=4$
		Normalization		$2n+1$	6	
	Uncalibrated perspective	Case 1	Projective	$2nm$	$11n+3m-15$	$n=3 \quad m=6$
			Euclidean	$4n+1$	10	
		Case 2	Projective	$2nm$	$11n+3m-15$	$n=5 \quad m=6$
			Euclidean	$13n+1$	55	
		Case 3	Projective	$2nm$	$11n+3m-15$	$n=3 \quad m=6$
			Euclidean	$3n+2$	10	
Multiple motion scene	Orthographic	Full rank	SVD	$2nm$	$12n+6m-35$	$n=4 \quad m=7$
			Normalization	$8n$	21	
		Rank-4	SVD	$2nm$	$8n+4m-15$	$n=4 \quad m=5$
			Normalization	$3n$	10	
		Rank-5	SVD	$2nm$	$10n+5m-24$	$n=3 \quad m=6$
			Normalization	$8n$	20	
	Weak perspective	Full rank	SVD	$2nm$	$12n+6m-35$	$n=4 \quad m=7$
			Normalization	$7n+1$	21	
		Rank-4	SVD	$2nm$	$8n+4m-15$	$n=5 \quad m=5$
			Normalization	$2n+1$	10	
		Rank-5	SVD	$2nm$	$10n+5m-24$	$n=3 \quad m=6$
			Normalization	$7n+1$	20	
	Perspective	Full rank	SVD	$2nm$	$12n+6m-35$	$n=4 \quad m=7$
			Normalization	$7n+1$	21	
		Rank-4	SVD	$2nm$	$8n+4m-15$	$n=5 \quad m=5$
			Normalization	$2n+1$	10	
		Rank-5	SVD	$2nm$	$10n+5m-24$	$n=3 \quad m=6$
			Normalization	$7n+1$	20	
	Uncalibrated perspective	Full rank	Projective	$2nm$	$20n+6m-48$	$n=4 \quad m=16$
			Euclidean	$17n+1$	28	
		Rank-4	Projective	$2nm$	$14n+4m-24$	$n=4 \quad m=8$
			Euclidean	$4n+1$	15	
		Rank-5	Projective	$2nm$	$17n+5m-35$	$n=3 \quad m=16$
			Euclidean	$17n+1$	27	

Table 5.2: **Minimum data requirement:** Analyzing the solution. n denotes the number of views and m denotes the number of feature points.

The minimum data requirement for SVD is,

$$n = 4 \quad m = 7 \tag{5.4}$$

The goal of the normalization process is to recover the 6×6 affine transformation A by imposing metric constraints on the matrix decomposition results. A is recovered by decomposing A as,

$$A = \begin{bmatrix} A_1 & A_2 \end{bmatrix} \tag{5.5}$$

and solving the 21 unknown elements of the symmetric matrix Q_1,

$$Q_1 = A_1 A_1^{\mathrm{T}} \tag{5.6}$$

Therefore, the number of variables is 21 and the number of equations is $8n$ which are the metric constraints. We get the constraint on the minimum data requirement for the normalization process,

$$n \geq 3 \tag{5.7}$$

Combining the above two constraints (Equation (5.4) and Equation (5.7)), we have $n = 4 \;\; m = 7$ as the minimum data required by the full rank orthographic reconstruction method for multiple motion scenes.

Reconstruction with uncalibrated cameras

The reconstruction with uncalibrated cameras consists of projective reconstruction and Euclidean reconstruction. Taking the uncalibrated Euclidean reconstruction method for static scenes (case 2) as example, a total of $2nm$ measurements are available to estimate the projective motion and shape. Each camera projection is represented by a 3×4 matrix which has 11 variables because of the homogeneous representation. Each feature point is represented by a 4×1 vector which has 3 variables. Since the projective reconstruction is up to an unknown 4×4 projective transformation, the total number of variables is $11n + 3m - 15$. The constraint of the projective reconstruction is,

$$2nm \geq 11n + 3m - 15 \tag{5.8}$$

Therefore, the minimum data requirement is,

$$n = 3 \quad m = 6 \tag{5.9}$$

Since we get the decomposition results from the projective reconstruction, the normalization pro-

cess is the only step having constraints on the minimum data requirement in the Euclidean reconstruction. In order to recover the 4×4 projective transformation H linearly, we set up $13n + 1$ equations of the 55 unknown elements of the matrix $\mathbf{q}\mathbf{q}^{\mathrm{T}}$, where each $\mathbf{q}$ is a 10×1 vector composed of the unknown elements of the symmetric matrix HH^{T}. Therefore, the number of equations is $13n + 1$ and the number of variables is 55, we get,

$$n \geq 5 \tag{5.10}$$

Combining the above two constraints, the minimum data required by the case 2 method is $n = 5 \quad m = 6$. This constraint is a low bound of the minimum data requirement because we do not analyze if the equations are independent.

5.1.3 Empirical results

We conduct a number of synthetic experiments to determine the minimum number of views and feature points required by the reconstruction methods presented in the thesis. We synthesize a cube as the static scene from which the feature points are chosen at generic locations, that is, any 4 points are not co-planar. The camera undergoes random motions whose rotation goes through a total of 30 to 50 degrees. The distance between the moving camera and the cube is about 15 to 20 times the cube size for orthographic projections, 10 to 15 times for weak perspective projections and 4 to 10 times for perspective projections. The image size is 640×480. The focal lengths are random numbers from 1000 to 2000 pixels. The principal points are shifted from the center of the images by 0 to 8 pixels. The aspect ratios are randomly set as any value between 0.8 and 1.2.

We try different values of n (number of views) and m (number of features) from the low bound generated by counting the arguments (Section 5.1.1) and analyzing the solution (Section 5.1.2), and choose the pair of n and m with the smallest value of n and its corresponding smallest value of m. We test the chosen pair of values by generating 10 sequences with different locations of the feature points and random camera motions. We confirm the pair of values and list them in Table 5.1.3 as the empirical results only if 10 sequences all generate reasonable results, which means,

- The maximum reconstruction error of the feature locations is less than 5% of the cube size;
- The maximum distance between the recovered camera positions and the ground truth values is less than 10% of the cube size;
- The recovered camera orientations are within 5° of the true orientations;
- The recovered focal lengths are within 10% of the ground truth values;
- The maximum reconstruction error of the principal points is less than 2 pixels;

Methods			Minimum data
Static scene	Orthographic		$n = 3 \quad m = 4$
	Weak perspective		$n = 3 \quad m = 4$
	Uncalibrated perspective	Case 1	$n = 3 \quad m = 6$
		Case 2	$n = 5 \quad m = 6$
		Case 3	$n = 8 \quad m = 10$
Multiple motion scene	Orthographic	Full rank	$n = 5 \quad m = 7$
		Rank-4	$n = 4 \quad m = 5$
		Rank-5	$n = 8 \quad m = 10$
	Weak perspective	Full rank	$n = 5 \quad m = 7$
		Rank-4	$n = 5 \quad m = 5$
		Rank-5	$n = 8 \quad m = 10$
	Perspective	Full rank	$n = 5 \quad m = 7$
		Rank-4	$n = 5 \quad m = 5$
		Rank-5	$n = 8 \quad m = 10$
	Uncalibrated perspective	Full rank	$n = 4 \quad m = 16$
		Rank-4	$n = 4 \quad m = 8$
		Rank-5	$n = 8 \quad m = 10$

Table 5.3: **Minimum data requirement:** Empirical results. n denotes the number of views and m denotes the number of feature points.

- The recovered aspect ratios are within 10% of the true values.

One interesting thing about the minimum data requirement for multiple motion scenes is that the minimum number of features is not dependent on how many of them are moving as long as the rank of the measurement matrix is same. Taking the full rank orthographic reconstruction for multiple motion scenes as an example, the minimum data requirement is $n = 5 \quad m = 7$. Synthetic experiments show that these 7 feature points can be composed of 4 static points and 3 moving points, or 3 static and 4 moving ones, or 2 static and 5 moving points, or 1 static and 6 moving ones, or even 0 static and 7 moving points. As long as the static points and the initial positions of the moving points are in a 3D space and the motion velocities span in a 3D space as well, the full rank reconstruction method works on any 7 points from 5 views. We have same results for the degenerate cases. For example, the rank-4 orthographic reconstruction method works on 4 views and 5 points no matter what the number of the static points is. It can be any value from 0 to 4 as long as the moving points are all moving in the same (or the opposite) direction. Wexler and Shashua's scene synthesis method [Wexler and Shashua, 2000] can work on the scene where all the feature points are moving, so can the multiple motion scene reconstruction methods presented in this thesis.

5.2 Gauge selection

Scene reconstruction from monocular image sequences is up to a similarity transform. For the linear and bilinear subspace methods presented in this thesis, we place the origin of the world coordinate system at the center of gravity of all the feature points, which is actually a moving origin when some of the feature points are moving, and align the orientation of the coordinate system with the first camera. The process is called *gauge selection* [Heyden, 1997, Morris *et al.*, 1999]. Recently, gauge theory [McLauchlan, 1999, Morris *et al.*, 1999, Triggs *et al.*, 2000, McLauchlan, 2000, Kanatani and Morris, 2000, Morris *et al.*, 2000a] has been developed to deal with the reconstruction ambiguities. In this section, we focus on analyzing the gauge selection process of the reconstruction methods presented in the thesis and demonstrate that fixing the gauge for the calibrated reconstruction methods saves computation cost and improves reliability of the reconstruction results.

5.2.1 Gauge selection in static scene reconstruction

Gauge selection is to determine the similarity transform which has 6 degrees of freedom: 3 for translation and 3 for rotation. The translation is decided when we fix the origin of the world coordinate system and the rotation is determined when we align the orientation of the world coordinate system.

Orthographic and weak perspective projections

We first use Tomasi and Kanade's orthographic factorization method [Tomasi and Kanade, 1992] as an example to demonstrate that fixing the origin of the world coordinate system decreases the number of variables and improves reliability of the reconstruction. We outline Tomasi and Kanade's factorization method as follows and illustrate the gauge selection techniques used in their method.

1. The world coordinate system location

   The method first places the origin of the world coordinate system at the center of gravity of all the feature points, based on which it calculates the camera translations. At this point, the **translation ambiguity** is solved.

2. Decomposition

   Subtract the translations from the measurement matrix and get the "registered" measurement matrix. A rank 3 SVD is performed on the "registered" matrix to generate the pair of motion and shape up to a $\mathbf{3} \times \mathbf{3}$ affine transformation A.

3. Normalization

Set up $3n$ linear equations to solve the $\mathbf{6}$ unknown elements of the $\mathbf{3} \times \mathbf{3}$ symmetric matrix $Q = AA^{\mathrm{T}}$. Once Q is computed, we get A by rank 3 matrix decomposition of Q. This decomposition is up to a three dimensional rotation because the matrix Q is symmetric. We can solve the **rotation ambiguity** by aligning the world coordinate system with any orientation, such as the first camera orientation.

4. Motion and shape recovery

   Once A has been found, we can recover the Euclidean motion and shape.

In Tomasi and Kanade's method, the translation ambiguity is solved at the beginning by fixing the origin of the world coordinate system. However, it is not necessary to perform the above four steps in this order. We reformulate the method to solve the translation ambiguity at the same time as computing the linear transformation. Costeira and Kanade [Costeira and Kanade, 1998] used a similar process for easier incorporation of multiple moving objects. The outline of the reformulated method is as follows:

1. Decomposition

   We perform a rank $\mathbf{4}$ SVD on the measurement matrix W and generate the pair of motion $\hat{M}$ and shape $\hat{S}$ up to a $\mathbf{4} \times \mathbf{4}$ linear transformation H.

2. Rotation constraints

   We decompose the transformation H according to $H = [A \;\; B]$, where A is $\mathbf{4} \times \mathbf{3}$ and B is $\mathbf{4} \times \mathbf{1}$. Imposing the same metric constraints as in Tomasi and Kanade's method, we set up $3n$ equations to solve the $\mathbf{10}$ unknown elements of the $\mathbf{4} \times \mathbf{4}$ symmetric matrix $Q = AA^{\mathrm{T}}$. A is computed from Q by matrix decomposition. Same as in Tomasi and Kanade's method, this decomposition is up to a 3D rotation. The **rotation ambiguity** is therefore solved in this step by fixing the orientation of the world coordinate system.

3. Translation constraints

   We can place the origin of the world coordinate system at arbitrary locations by setting B to any values which make the matrix $H = [A \;\; B]$ non-singular. Therefore, the **translation ambiguity** is solved without extra computation. We can also place the origin at the center of gravity by computing B as,

   $$\overline{\mathbf{w}} = \hat{M}B \tag{5.11}$$

   where $\overline{\mathbf{w}}$ is the mean vector of the measurement matrix W. B is overconstrained when there are two or more than two views,

   $$B = \hat{M}^{-1}\overline{\mathbf{w}} \tag{5.12}$$

   where $\hat{M}^{-1}$ is the general inverse of the matrix $\hat{M}$.

4. Motion and shape recovery

   Once A and B have been found, we can recover the Euclidean motion and shape by $H = [A \;\; B]$.

Comparing these two formulations of the orthographic factorization method, it is clear that Tomasi and Kanade's method has less variables to solve. The number of variables is 6 in Tomasi and Kanade's method while it is 10 in the other formulation. Tomasi and Kanade's method deals with a smaller space which is rank 3 instead of rank 4. Same analysis applies to the weak perspective factorization method. We perform a set of synthetic experiments to compare the reliability of these two formulations. We synthesize 60 sequences with increased noise at feature locations. Each sequence has 100 frames and 50 feature points. The feature noise is from 0 to 3 pixels. The image size is 640×480. We measure the shape error as the average of the distances between the recovered feature points and their corresponding true values. The value of the average shape error shown in Figure 5.1 is the ratio between the average error and the object size. Figure 5.1 shows that the shape errors reconstructed by the two orthographic formulations increase with the feature noise and Tomasi and Kanade's formulation is more reliable.

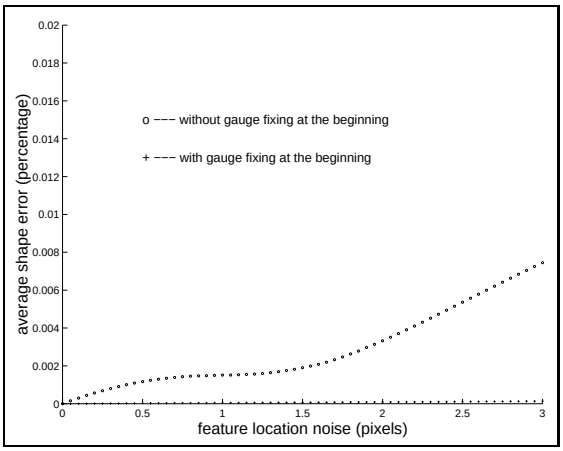


Figure 5.1: **Gauge selection for orthographic projection:** Average shape errors recovered by the two formulations for orthographic reconstruction. It shows that the shape errors increase with the feature noise and the formulation which fixes the gauge at the beginning (Tomasi and Kanade's method) is more reliable.

Uncalibrated perspective projections

The uncalibrated Euclidean reconstruction method described in Chapter 2 places the origin of the world coordinate system at the center of gravity of the feature points, therefore, it solves the translation ambiguity at the beginning of the Euclidean reconstruction process. However, the camera translation vector cannot be computed directly from the scaled measurement matrix generated by projective reconstruction. Therefore, it is impossible to decrease the number of variables by fixing the origin as

for affine projections. The conclusion is that we can either fix the origin at the center of gravity and compute B vector of the linear transformation $H = [A \ B]$ as presented in Chapter 2, or set B to any value which corresponds to an arbitrary 3D location of the origin. The computation cost for normalization and reliability of the results are same. Figure 5.2 shows the reconstructed shape errors by the two formulations for the case 1 normalization algorithm, which are almost same.

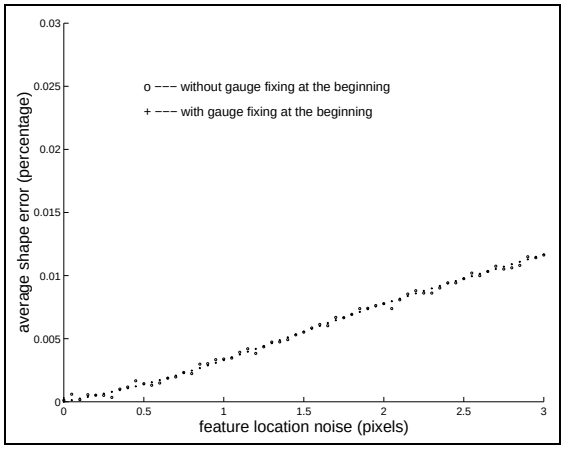


Figure 5.2: **Gauge selection for uncalibrated perspective projection:** Average shape errors recovered by the two formulations for uncalibrated Euclidean reconstruction. It shows that the shape errors increase with the feature noise and the results from the two formulations are very close.

5.2.2 Gauge selection in multiple motion scene reconstruction

The reconstruction methods for multiple motion scenes are based on the unified representation of feature points no matter if they are moving or not. This representation induces the difficulty of fixing the gauge because the center of gravity of all the feature points is moving. It is interesting to notice that the center of gravity is moving linearly with constant speed because we assume that the moving points have constant velocities. Therefore, we define the world coordinate system as a moving system with its origin at the moving center of gravity and its orientation fixed. In this section we demonstrate that the design of the **moving** world coordinate system enables the reconstruction process work on a smaller space so that the results are more stable.

Table 5.4 compares the orthographic reconstruction processes for multiple motion scenes with and without solving the translation ambiguity at the first step. We can see that the method fixing the origin at the first step, which is the method presented in Chapter 3, has less computation. Figure 5.3 shows the reconstruction errors of the two formulations under orthographic projections. The average shape error evaluates the reconstruction errors of the static feature points and the initial positions of the moving feature points. We can see that the formulation which fixes the moving origin at the center of gravity at

With gauge fixing at the first step	Without gauge fixing at the first step
1. World coordinate system location Fix the origin of the moving world coordinate system at the center of gravity of all the feature points and compute the camera translations. This step solves the **translation ambiguity**.	**1. Decomposition** Perform a rank $\mathbf{7}$ SVD on the measurement matrix and get the pair of motion and shape up to a $\mathbf{7} \times \mathbf{7}$ linear transformation $H = [A\ B]$.
2. Decomposition Subtract the camera translations from the measurement matrix and perform a rank $\mathbf{6}$ SVD on the "registered" measurement matrix to generate the pair of motion and shape up to a $\mathbf{6} \times \mathbf{6}$ linear transformation H.	**2. Rotation constraints** Set up $8n$ linear equations of the $\mathbf{28}$ unknown elements of the $\mathbf{7} \times \mathbf{7}$ symmetric matrix Q. A is computed from Q by rank 3 matrix decomposition which is up to a 3D rotation. We solve the **rotation ambiguity** by aligning the world coordinate system with first camera (or any) orientation.
3. Normalization Set up $8n$ linear equations of the $\mathbf{21}$ unknown elements of the $\mathbf{6} \times \mathbf{6}$ symmetric matrix Q. H is computed from Q by rank 3 matrix decomposition which is up to a 3D rotation. We solve the **rotation ambiguity** by aligning the world coordinate system with first camera (or any) orientation.	**3. Translation constraints** Set B to any values which make H non-singular. It solves the **translation ambiguity** by placing the origin of the world coordinate system at arbitrary locations. Or we can solve B by $B = \hat{M}^{-1}\overline{\mathbf{w}}$ to place the origin at the center of gravity of all the feature points.
4. Motion and shape recovery Once H has been recovered, the motion and shape are computed from H and the moving features are automatically detected.	**4. Motion and shape recovery** Once A and B are recovered, we get $H = [A\ B]$. The motion and shape are computed from H and the moving features are detected.

Table 5.4: **Gauge selection:** Comparison of two orthographic reconstruction processes for multiple motion scenes with and without gauge fixing at the first step of reconstruction.

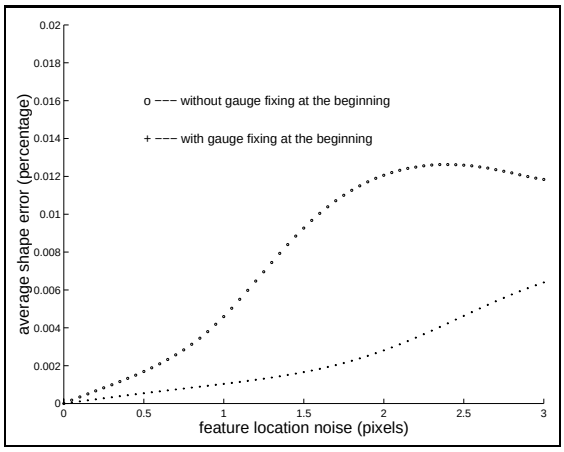


Figure 5.3: **Gauge selection for multiple motion scenes under orthographic projection:** Average shape errors recovered by the two formulations for multiple motion scenes orthographic reconstruction. It shows that the shape errors increase with the feature noise and the formulation which fixes the gauge at the beginning (the multiple motion scene orthographic reconstruction method presented in Chapter 3) is more reliable.

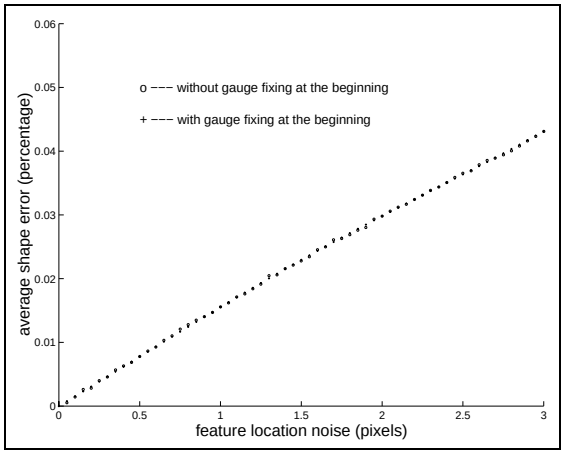


Figure 5.4: **Gauge selection for multiple motion scenes under uncalibrated perspective projection:** Average shape errors recovered by the two formulations for multiple motion scenes uncalibrated reconstruction. It shows that the shape errors increase with the feature noise and the results from the two formulations are very close.

the beginning is more reliable. We demonstrate the full rank case only though the analysis applies to the degenerate cases as well.

Following the same analysis for static scenes, the uncalibrated reconstruction methods for multiple motion scenes have the similar computation cost and reliability for the two formulations. Figure 5.4 shows the results.

Chapter 6

Conclusion

When a camera moves around in a scene, the images taken contain information about the camera and the scene structure. We address two interesting problems in the area of Structure from Motion. One is about the camera. We work on the image sequences taken with uncalibrated cameras. The other is about the scene. We deal with the scenes rich with moving objects.

We present three linear and bilinear subspace methods in this thesis. The uncalibrated Euclidean reconstruction method works on image sequences of static scenes taken with uncalibrated cameras. The multiple motion scene reconstruction method with calibrated cameras and the multiple motion scene reconstruction method with uncalibrated cameras both deal with image sequences of scenes rich with moving objects.

We also discuss two important issues of the reconstruction methods: minimum data requirement and gauge selection. The theoretical analysis and the empirical results are presented.

In this chapter we first summarize the contributions of this thesis in terms of theoretical work, system work and potential applications. Then we identify the directions for future work to continue this line of research.

6.1 Contributions

6.1.1 Theories

1. Decouple the uncalibrated reconstruction process into projective reconstruction and Euclidean reconstruction.

   Given tracked feature points from multiple uncalibrated views, we first perform a bilinear projective reconstruction process to generate the scaled image measurements from which we get the projective motion and shape. The Euclidean reconstruction then converts the projective results

into the Euclidean ones by enforcing metric constraints. These two steps are relatively independent, that is, the Euclidean reconstruction method can work on the output from any projective reconstruction method while the projective results can be fed into any self-calibration method.

2. Embed the camera intrinsic parameters recovery within the Euclidean reconstruction.

   Image measurements are generated by the projection of 3D scene structure to 2D images. The projection is determined by the camera motion and the camera intrinsic parameters. We embed the unknown camera intrinsic parameters within the camera motion representation for uncalibrated cases, based on which we present the Euclidean reconstruction method.

3. Recover the multiple motion scenes with the assumption that the objects are moving linearly with constant speeds.

   Assuming the objects are moving linearly with constant speeds, we proposed a unified geometrical representation of the static scene and the moving objects. The representation incorporates the motion information within the scene representation, which naturally leads to a factorization-based method.

6.1.2 Systems

1. A uncalibrated reconstruction method which recovers the Euclidean shape, the camera motion and the camera intrinsic parameters from multiple uncalibrated perspective views.

2. A multiple motion scene reconstruction method which reconstructs a scene containing multiple moving objects together with the camera motion from monocular image sequences.

3. A uncalibrated multiple motion scene reconstruction method which recovers the scene structure, the trajectories of the moving objects, the camera motion and the camera intrinsic parameters simultaneously.

We build three reconstruction systems based on the above three methods respectively. A series of experiments on synthetic and real image sequences are conducted. We also address the issues of minimum data requirement and gauge selection of the reconstruction methods with the theoretical analysis and the empirical results.

6.1.3 Applications

1. Multi-camera calibration.

   Obtaining the ground truth is difficult and time-consuming in camera calibration. The uncalibrated reconstruction method provides a good way to calibrate multi-camera systems. Instead of

carefully putting objects at accurate positions, a person can wave a stick with LEDs randomly in the room. The LEDs enable fast and easy computation of correspondences. Given these tracked feature points, the reconstruction method can be applied to recover the camera extrinsic and intrinsic parameters simultaneously.

2. Video analysis.

   Scene modeling can be regarded as an efficient way of representing the large amount of information in image sequences, especially the feature-based modeling presented in this thesis. It can be applied to video editing, image based rendering, video compression, video retrieval and summarization.

6.2 Future work

6.2.1 Critical motion sequences

Sequences of camera motions that lead to inherent ambiguities in uncalibrated Euclidean reconstruction or self-calibration are referred to as *critical motion sequences* [Sturm, 1997a, Sturm, 1997b], that is, there are situations in which any uncalibrated reconstruction method fails or is exceptionally weak. The critical situations are often independent of the specific camera intrinsic parameters. They are related to certain types of camera motions which prevent unique Euclidean reconstruction.

In this thesis we present a collection of reconstruction methods and conduct the experiments under generic camera motions. In practice, it is important to analyze the critical motion sequences for the methods so that we can detect and avoid the critical and "close to critical" situations.

- Static scene reconstruction with uncalibrated cameras

  Kahl et al. [Kahl *et al.*, 2000] applied subgroup approach to self-calibration constraints when some of the intrinsic parameters can vary. They proved that *given the plane at infinity and known skew, aspect ratio and principal point, then a motion is critical if and only if there is only one viewing direction.* The explicit geometric descriptions of the corresponding critical motion sequences are: (i) arbitrary rotations about the optical axis and translations, (ii) arbitrary rotations about at most two centers, (iii) forward-looking motions along an ellipse and/or a corresponding hyperbola in an orthogonal plane. These are the critical motion sequences for case 1 of the uncalibrated Euclidean reconstruction method presented in Chapter 2. They also analyze the case with zero skew and unit aspect ratio which covers case 2 of the uncalibrated reconstruction method. The critical motion sequences for this case are *there are at most two viewing directions*. Sturm [Sturm, 1999] described the critical motion sequences for stereo systems with varying focal lengths. We have not seen any work done for the case when all of the intrinsic parameters

are unknown and varying except skews (case 3 of the uncalibrated reconstruction method). It is necessary to analyze the critical motion sequences for this case in order to determine if a multi-camera set up is possible to be self-calibrated. We can start with extensions of the subgroup approach presented by Kahl et al. [Kahl *et al.*, 2000] to the case where the intrinsic parameters are varying.

- Multiple motion scene reconstruction with calibrated cameras

  Kahl and Triggs [Kahl and Triggs, 1999] investigated the critical motion sequences for intrinsically calibrated orthographic and perspective cameras. Their conclusions included: (i) for any two orthographic and weak perspective cameras, there is a one parameter family of possible structures corresponding to the bas relief ("flattening") ambiguity, (ii) for any two calibrated perspective cameras, there is always a two-fold ambiguity corresponding to a "twisted pair". The twisted pair duality is caused by the rotation of one of the cameras by 180° around the axis joining the two optical centers. These conclusions are based on the assumption that the scenes are static. The minimum number of views required for multiple motion scene reconstruction is larger than 2, however, we are dealing with a much larger reconstruction space than static scenes. For example, the affine transformation space is 6D for multiple motion scenes (full rank case) while it is 3D for static scenes. We need to explore if there are ambiguities for more than two views due to the moving objects, and if the critical motions depend on the directions of the moving objects.

- Multiple motion scene reconstruction with uncalibrated cameras

  Some research has been done on the analysis about the critical motion sequences of systems with varying focal lengths [Sturm, 1999, Kahl *et al.*, 2000]. Most work is limited to static scenes. It is interesting to apply the static scene results to the uncalibrated multiple motion scene reconstruction in order to figure out if the critical motion sequences for static scenes are still critical for multiple motion scenes. As mentioned above, we also want to work on if there are critical motion sequences caused by the moving objects, and if the critical motion sequences are related to the directions of the moving objects. The important thing is not only to study the critical motion problem in theory, but also to design a system which can detect if the recovered camera motion is critical so as to avoid the critical and "close to critical" situations.

6.2.2 Uncertainty modeling

The reconstruction methods presented in this thesis are based on linear and bilinear subspace constraints. Singular Value Decomposition (SVD) is used to get the best low-rank approximation of the given measurement matrix. However, SVD is powerful at getting the global minimum only when the feature errors are directional uncorrelated and identically distributed. This is rarely the case in real

data. It is necessary to model the directional uncertainty of features and perform a minimization on a covariance-weighted error measurement.

When the feature uncertainty is isotropic, but not identical, Aguiar and Moura described the rank-1 factorization algorithm to perform a scalar-weighted SVD for motion and shape recovery [Aguiar and Moura, 1999]. Morris and Kanade [Morris and Kanade, 1998] presented a unified orthographic factorization algorithm for points, line segments and planes using directional uncertainty model of features. They solved motion and shape by a quasi-linear algorithm. More interestingly, they can evaluate the reconstructed shape based on the statistical uncertainty model. They discussed their work on perspective cameras in [Morris *et al.*, 2000b]. Irani and Anandan [Irani and Anandan, 2000] described an approach to transform the raw noisy data into a covariance-weighted data space where the noises are directional uncorrelated and identically distributed. In this way they could apply SVD to the transformed data to factor noisy feature correspondences with high degree of directional uncertainty into motion and shape. Not limited to directional uncertainty models, Sun et el. [Sun *et al.*, 1999] discussed error characterization of the factorization methods using results from matrix perturbation theory and covariance propagation for linear models.

There are two reasons why we want to include feature uncertainty models into the reconstruction methods presented in the thesis. One is to improve the accuracy and reliability of the reconstruction results since the directional uncertainty is modeled. Another is to evaluate the reconstruction results quantitatively based on the statistical models.

There is not much work done on uncertainty modeling of uncalibrated reconstruction methods. We are interested in extending Morris and Kanade's approach [Morris and Kanade, 1998] and Irani and Anandan's approach [Irani and Anandan, 2000] to the uncalibrated Euclidean reconstruction method for static scenes. There is no doubt the reconstruction results can be improved given correct directional uncertainty. More importantly, we want to compute the covariance of the recovered camera intrinsic parameters, the camera motion and the scene structure so that we can evaluate the accuracy of the self-calibration results.

It is interesting to analyze the reliability of the reconstruction methods for multiple motion scenes. Assuming that the objects are moving linearly with constant speeds, we propose a unified geometrical representation incorporating the static scene and the moving objects. This representation enables the embedding of the motion constraints into the scene structure, that is, the current shape matrix is composed of two spaces: one is the scene structure space and another is the motion space. The methods make use of the constraints between the camera motion and the current shape matrix to perform the reconstruction. Experiments show that the reconstruction is reliable in the presence of noise. However, theoretical analysis is necessary about the sensitivity to noise of the two spaces (the scene space and the motion space) because each feature point, either static or moving, contributes to the scene space and only the moving points contribute to the motion space. We applied Morris and Kanade's approach

[Morris and Kanade, 1998] to the multiple motion scene orthographic reconstruction method and got some preliminary results.

Suppose G_{ij} is the inverse covariance of the jth feature location at the ith image, $\mathbf{w}_{ij} = [u_{ij} \;\; v_{ij}]^{\mathrm{T}}$ denote the tracked feature location, the error function with uncertainty feature models is,

$$\mathrm{Err} = \sum_{i,j} \frac{1}{2} \left(\mathbf{w}_{ij} - M_i \mathbf{d}_j\right)^{\mathrm{T}} G_{ij} \left(\mathbf{w}_{ij} - M_i \mathbf{d}_j\right) \tag{6.1}$$

where M_i represents the "rotation" matrix of the ith camera for multiple motion scenes composed of the rotation axes $\mathbf{i}_i$, $\mathbf{j}_i$ and the scaled rotation axes $i\mathbf{i}_i$, $i\mathbf{j}_i$, $\mathbf{d}_j$ is the jth "shape" vector composed of the initial position $\mathbf{s}_j$ of the feature and its velocity $\mathbf{v}_j$,

$$M_i = \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} & i\mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} & i\mathbf{j}_i^{\mathrm{T}} \end{bmatrix} \qquad \mathbf{d}_j = \begin{bmatrix} \mathbf{s}_j \\ \mathbf{v}_j \end{bmatrix} \tag{6.2}$$

The maximum likelihood solution for motion and shape is obtained by minimizing Err with respect to the shape and motion parameters. We perform a bilinear minimization process similar to the algorithm described in [Morris and Kanade, 1998] with the difference that we are dealing with a 6 dimensional motion and shape space while Morris and Kanade were working on 3 dimensions.

It is interesting to analyze the shape uncertainty. Since every feature is represented by a 6×1 vector, the inverse covariance of each feature is the Hessian of Err in the shape parameters,

$$H_j = \begin{bmatrix} \frac{\partial^2 \mathrm{Err}}{\partial s_{xj}^2} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj} \partial s_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj} \partial s_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj} \partial s_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj} \partial s_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj} \partial s_{xj}} \\ \frac{\partial^2 \mathrm{Err}}{\partial s_{xj} \partial s_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj}^2} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj} \partial s_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj} \partial s_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj} \partial s_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj} \partial s_{yj}} \\ \frac{\partial^2 \mathrm{Err}}{\partial s_{xj} \partial s_{zi}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj} \partial s_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj}^2} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj} \partial s_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj} \partial s_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj} \partial s_{zj}} \\ \frac{\partial^2 \mathrm{Err}}{\partial s_{xj} \partial v_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj} \partial v_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj} \partial v_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj}^2} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj} \partial v_{xj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj} \partial v_{xj}} \\ \frac{\partial^2 \mathrm{Err}}{\partial s_{xj} \partial v_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj} \partial v_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj} \partial v_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj} \partial v_{yj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj}^2} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj} \partial v_{yj}} \\ \frac{\partial^2 \mathrm{Err}}{\partial s_{xj} \partial v_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{yj} \partial v_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial s_{zj} \partial v_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{xj} \partial v_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{yj} \partial v_{zj}} & \frac{\partial^2 \mathrm{Err}}{\partial v_{zj}^2} \end{bmatrix} = \sum_i M_i^{\mathrm{T}} G_{ij} M_i \tag{6.3}$$

Focusing on the diagonal blocks of H_j, the upper left corner (denoted as a 3×3 matrix H_{js}) approximates the inverse covariance of the initial position of the jth feature and the lower right corner (denoted as a 3×3 matrix H_{jv}) approximates the inverse covariance of its velocity. We have,

$$\begin{aligned} H_{js} &= \sum_i \begin{bmatrix} \mathbf{i}_i & \mathbf{j}_i \end{bmatrix} G_{ij} \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \end{bmatrix} \\ H_{jv} &= \sum_i \begin{bmatrix} i\mathbf{i}_i & i\mathbf{j}_i \end{bmatrix} G_{ij} \begin{bmatrix} i\mathbf{i}_i^{\mathrm{T}} \\ i\mathbf{j}_i^{\mathrm{T}} \end{bmatrix} = \sum_i i^2 \begin{bmatrix} \mathbf{i}_i & \mathbf{j}_i \end{bmatrix} G_{ij} \begin{bmatrix} \mathbf{i}_i^{\mathrm{T}} \\ \mathbf{j}_i^{\mathrm{T}} \end{bmatrix} \end{aligned} \tag{6.4}$$

Therefore, we can approximate the inverse covariance of velocities as $\mathrm{o}(n^2)$ times the inverse covariance of positions. We can prove geometrically that this relationship is correct.

At moment t, the position of a feature point is represented by a 3×1 vector $\mathbf{s}_t$,

$$\mathbf{s}_t = \mathbf{s}_0 + t\mathbf{v} \tag{6.5}$$

where $\mathbf{s}_0$ is its initial position and $\mathbf{v}$ is its velocity. We have,

$$\Delta\mathbf{s}_t = \Delta\mathbf{s}_0 + t\Delta\mathbf{v} = \begin{bmatrix} 1 & t \end{bmatrix} \begin{bmatrix} \Delta\mathbf{s}_0 \\ \Delta\mathbf{v} \end{bmatrix} \tag{6.6}$$

therefore,

$$\mathrm{V}(\Delta\mathbf{s}_t) = \begin{bmatrix} 1 & t \end{bmatrix} \mathrm{V}_{s_0 v} \begin{bmatrix} 1 \\ t \end{bmatrix} \approx \mathrm{V}(\Delta s_0) + t^2\mathrm{V}(\Delta v) \tag{6.7}$$

where Δ represents perturbation and $\mathrm{V}(\Delta)$ is its variance. This equation shows that the scale between the variance of the initial position and that of its velocity is $\mathrm{o}(n^2)$, which demonstrates that the relationship between the inverse covariances (Equation (6.4)) is correct.

6.2.3 Sequences with missing data

The reconstruction methods described in this thesis do not work for the image sequences with missing data, that is, they require that each feature point is visible in each frame. However, practically, there are many image sequences in which the camera views several distinct parts of the scenes due to the camera motion and occlusion. It is desirable to incorporate the information of missing data into the reconstruction framework.

Shum et al. [Shum *et al.*, 1995] proposed an iterative method which minimized the sum of square differences between the fitted low rank matrix and the elements that are not missing in the data matrix. This method can always converge to a locally optimal solution, however, it is not guaranteed to find the global minimum. Urban et al. [Urban *et al.*, 1999] presented a linear projective reconstruction method from image sequences with missing data. It requires the images share a common reference view. Jacobs [Jacobs, 1997] fit a low rank matrix to a matrix with missing elements by combining constraints on the solution derived from small submatrices of the full matrix. He also presented the application of the linear fitting method to structure from motion problem. The basic idea is to regard the missing data estimation and recovery problem as a EM process in order to find maximum likelihood estimates for unknown values. We are interested in exploring the possibilities of incorporating the linear fitting idea into the reconstruction methods presented in this thesis.

6.2.4 Dense shape recovery

The linear and bilinear subspace methods presented in this thesis are feature-based methods. However, it is important to notice that the subspace constraints used throughout the thesis are not limited to the finite space composed of feature points. It is the computation and representation cost which prevent direct application of the methods to dense shape recovery.

We have been working on a system which is a combination of the feature-based uncalibrated reconstruction method and the dense stereo algorithm using level set methods proposed by Faugeras and Keriven [Faugeras and Keriven, 1996, Faugeras and Keriven, 1997, Faugeras and Keriven, 1998a, Faugeras and Keriven, 1998b]. Given calibrated image sequences, the level set stereo algorithm reconstructs the dense shape with the assumptions:

- Camera projections are perspective.
- Object surface is locally smooth.
- Images of the same 3D point share the same intensity.

We are interested in the level set algorithm because its advantages can nicely compensate for the disadvantages of the linear and bilinear subspace methods:

- There is no need to determine the correspondences beforehand for the level set algorithm, while the subspace methods require tracked feature points.
- Visibility and occlusion problems are handled naturally by surface evolution in the level set algorithm, while the subspace methods cannot deal with the sequences with missing data.
- Dense shape is recovered and textureless part is dealt with by surface smoothing, while the subspace methods recover sparse feature positions.

The level set algorithm starts with an initial surface which covers (or stays inside of) the real object. The goal is to move this surface along its normal directions to fit the real object surface by minimizing the intensity errors between the projections of the same 3D point. The Euler-Lagrange equations of the error functional, which are a set of Partial Differential Equations (PDEs), are solved as a time evolution process by level set methods.

Faugeras and Keriven's technical report ([Faugeras and Keriven, 1996]) described details of the derivation and provided nice 2D results. Their conference paper on ECCV'98 ([Faugeras and Keriven, 1998a]) gave 3D results while the journal version ([Faugeras and Keriven, 1998b]) presented several implementation hints. We implemented the level set stereo algorithm and built a combination system based on the subspace reconstruction methods and the level set algorithm, which works on uncalibrated views. We summarize the system as follows:

1. Track feature points (initialized by clicking or automatically selecting feature points on the first frame) by Lucas-Kanade method [Lucas and Kanade, 1981a];
2. Apply the uncalibrated Euclidean reconstruction method to get the camera calibrations and the 3D locations of the feature points;
3. Initialize one 3D surface (currently we are using semi-sphere) which covers the feature points;
4. Repeat the following steps for each point on the surface (parameterized by (v, w)) and for each pair of images where this point is visible;
   (a) Compute the surface point $\mathbf{S}(v, w)$ and the normal vector $\mathbf{N}(v, w)$;
   (b) Compute the mean curvature H and the curvature gradient $d\mathbf{N}$;
   (c) Project $\mathbf{S}(v, w)$ to the pair of images and get the image coordinates $\mathbf{m}_1$ and $\mathbf{m}_2$;
   (d) Compute the homography K from which we can get the affine matrix A;
   (e) Integrate over the image patches (we use 5×5 windows) and compute the change rate β of the normal;
   (f) Move the surface: $\mathbf{S} = \mathbf{S} + \beta\mathbf{N}$.

We would like to explore the following questions based on this system:

- Can this algorithm be regarded as a good way to get dense correspondences? How good is it comparing with Lucas-Kanade method ([Lucas and Kanade, 1981a]) and Irani's rank constrained method ([Irani, 1999])?
- How can the idea of this algorithm be extended to the subspace methods in order to get a dense shape (even with correspondenceless and missing data)?

Appendix A

Homography-Based Scene Analysis from Image Sequences

In this appendix we describe a framework to recover scene depth based on image homography and discuss its application to scene analysis from image sequences. We propose a robust homography algorithm which incorporates contrast/brightness adjustment and robust estimation into image registration. We then present a camera motion solver to obtain the ego-motion and the real/virtual dominant plane position from the image homography, and apply the Levenberg-Marquardt method to generate a dense depth map. We also discuss temporal integration of information over image sequences. Finally we present the results of applying the homography-based method to motion detection problem.

A.1 Introduction

Approaches handling 3D scene analysis from monocular image sequences can be classified into two categories: algorithms which use 2D transformation or model fitting, and algorithms which use 3D geometry analysis. The first category works for the situations where the scene is flat or the camera undergoes pure panning and zooming. The second one deals with the situations where the scene is close to cameras. Image sequences of our interest are taken from a moving airborne platform where the ego-motion is complex and the scene is relatively distant but not necessarily flat, therefore, an integration of 2D and 3D algorithms is more appropriate.

Most approaches of structure from motion were feature-based and could not provide dense depth maps. Xiong and Shafer presented a flow-based method [Xiong and Shafer, 1995] to recover dense shape via the Kalman Filter. They assumed that the feature correspondences were given. Baker et al. [Baker *et al.*, 1998] proposed a method to deal with multi-layer scenes, however, layer segmentation remained a problem. Incorporating 3D geometry into 2D constraints was widely used in motion detec-

tion and segmentation [Shashua and Werman, 1995, Irani and Anandan, 1998]. The plane plus parallax method contributes a great deal to ego-motion computation [Irani *et al.*, 1997], parallax geometry analysis [Kumar *et al.*, 1994, Irani and Anandan, 1996, Irani *et al.*, 1999] and video indexing [Irani *et al.*, 1998].

Temporal information redundancy of image sequences allows us to use efficient, incremental methods which perform temporal integration of information for gradual refinement. We first calculate image homography between consecutive images since the camera-to-scene distance is relatively large and therefore we can use the first-order approximation of the scene as being planar. Section A.2 describes the three components to achieve robust homography including contrast/brightness adjustment, progressive complexity of transformation and robust estimation. Based on the image homography, a camera motion solver is presented in Section A.3 to compute the camera ego-motion and the plane equation, then the Levenberg-Marquardt optimization is used to recover the dense depth map of the scene. Temporal integration is performed over image sequences to refine the scene depth. The results of applying the homography-based method to motion detection are discussed in Section A.4.

A.2 Robust homography

Monocular image sequences taken from a moving airborne platform usually include lighting and environmental changes. Contrast and brightness adjustment is therefore very critical in image registration. Registration by image homography is based on the assumption that either the scene is planar or the camera is only undergoing rotation and/or zooms. However, many image sequences are taken with no restriction of the camera motion and the scenes do not have dominant planes. Therefore, it is necessary to use statistical techniques to obtain robust homography. We incorporate contrast/brightness adjustment and robust estimation into image registration to generate **dominant homography** for complex environments.

A.2.1 Image intensity adjustment

Homography defines the relationship between two images by an eight-parameter perspective transformation,

$$\mathbf{x}' \sim P\mathbf{x} \tag{A.1}$$

where

$$\mathbf{x}' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad P = \begin{bmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{22} & P_{23} \\ P_{31} & P_{32} & P_{33} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{A.2}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the homogeneous representations of the corresponding image coordinates and $\sim$ denotes equality up to a scale. Szeliski and Shum [Szeliski and Shum, 1997] gave a simple solution for the transformation based on which we design the robust homography algorithm.

Due to the difference of viewpoints and change of lighting, image sequences may have different intensity levels from frame to frame. We model the change between images as a linear transformation [Lucas and Kanade, 1981b]:

$$I_0(\mathbf{x}) = \alpha I_1(\mathbf{x}') + \beta \tag{A.3}$$

where α stands for the contrast change, β for the brightness change, I_0 and I_1 are the two images. Combining this with Szeliski and Shum's homography computation [Szeliski and Shum, 1997], we obtain the error function E,

$$E(D,\ \alpha,\ \beta) = \sum_i \left[I_0(\mathbf{x}_i) - \alpha \hat{I}_1(\mathbf{x'}_i) - \beta \right]^2 \tag{A.4}$$

where $\hat{I}_1$ is the warped image of I_1 by the current homography P which is initialized as the 3×3 identity matrix, D is the incremental update for P,

$$(I + D)P \Longrightarrow P \tag{A.5}$$

and each $\mathbf{x'}_i$ is calculated as,

$$\mathbf{x'}_i \sim (I + D)\mathbf{x}_i \tag{A.6}$$

We minimize the error metric using a symmetric positive definite (SPD) solver such as Cholesky decomposition which is time efficient.

A.2.2 Progressive transformation complexity

The image homography is computed hierarchically on Laplacian image pyramids where estimates from coarser levels of the pyramids are used to initialize the registration at finer levels [Anandan, 1989, Bergen *et al.*, 1992]. To decrease the likelihood of the minimization process converging into local minima and to improve the registration speed, we use different transformations with progressive complexity at different pyramid levels, that is, we use translation (2 parameters) at the coarsest level, then scaled rotation plus translation (4 parameters), affine transformation (6 parameters), and finally perspective transformation (8 parameters) at the finest level. The progressive method improves the reliability of the homography computation.

A.2.3 Robust estimation

To deal with scenes without dominant planes, we use robust estimation to compute image homography. The random sample consensus paradigm (RANSAC) [Fischler and Bolles, 1981] is an early example of robust estimation. Similar geometric statistics were also explored in motion analysis approaches [Torr and Murray, 1997, Kanatani, 1997]. We apply the RANSAC scheme to the homography computation by randomly choosing a small subset of the images to obtain an initial homography solution where the subset defines a real/virtual plane, and then identifying the outliers which are the points not lying on the plane. The process is repeated enough times on different subsets and the best solution is the homography which maximizes the number of points lying on the plane. Points which are not identified as outliers are used to obtain the dominant homography as the final step.

The three components (image intensity adjustment, progressive transformation complexity and robust estimation) are used in combination to achieve the robust homography. Figure A.1(a) and (b) show two aerial images of buildings taken under different lighting conditions. The robust estimation randomly chooses 20 subsets, each of which is equal to 5% of the whole image. Each subset generates a homography. The best homography has the largest support area in the image. This area is used to compute the final homography. In this example, the support area for the final homography consists of the tops of several short buildings rather than the real ground because the ground is not actually flat. White dots in Figure A.1(c) are the outliers of the final homography which correspond to the tops of the tall buildings (closer to the camera than the dominant plane) and part of the ground (farther than the dominant plane).

A.3 Recovery of scene depth

A.3.1 Scene depth and homography

Let $\mathbf{x}$ and $\mathbf{x}'$ denote the homogeneous coordinates of the corresponding pixels in two images. The corresponding scene point can be represented by the homogeneous coordinates $[u \;\; v \;\; f \;\; w]^{\mathrm{T}}$ in the 3D coordinate system of the first image, therefore,

$$\mathbf{p} = \left[\frac{u}{w} \;\; \frac{v}{w} \;\; \frac{f}{w}\right]^{\mathrm{T}} \tag{A.7}$$

where w denotes the depth to be recovered, which is called projective depth of point $\mathbf{p}$ in [Szeliski, 1996]. $\mathbf{p}'$ denotes the same scene point with respect to the second image coordinate system,

$$\mathbf{p}' = R\mathbf{p} + T' \tag{A.8}$$

where R represents the rotation between the two image coordinate systems and T' represents the 3D translation between the two views expressed in the second image coordinate system.

Assuming that the cameras are intrinsically calibrated except the focal lengths, we use,

$$V = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad V' = \begin{bmatrix} f' & 0 & 0 \\ 0 & f' & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.9}$$

to represent the perspective projections of the two images and obtain,

$$\begin{aligned} \mathbf{x}' &\sim V'\mathbf{p}' \\ &= V'R\mathbf{p} + V'T' \\ &\sim V'RV^{-1}\mathbf{x} + V'T'\frac{w}{f} \end{aligned} \tag{A.10}$$

Each 3D planar surface can be represented by a 1×3 vector $[a \;\; b \;\; c]$, which is the scaled normal direction whose size denotes the inverse of the distance to the plane from the origin. If $\mathbf{p}$ is on the plane,

$$\begin{bmatrix} a & b & c \end{bmatrix} \mathbf{p} = 1 \tag{A.11}$$

we get,

$$\begin{bmatrix} a & b & c \end{bmatrix} V^{-1}\mathbf{x} = \frac{w}{f} \tag{A.12}$$

According to Equations (A.10) and (A.11), we get,

$$\begin{aligned} \mathbf{x}' &\sim V'RV^{-1}\mathbf{x} + V'T'\frac{w}{f} \\ &= V'RV^{-1}\mathbf{x} + V'T'\begin{bmatrix} a & b & c \end{bmatrix} V^{-1}\mathbf{x} \\ &= V'(R + T'\begin{bmatrix} a & b & c \end{bmatrix})V^{-1}\mathbf{x} \\ &\sim P\mathbf{x} \end{aligned} \tag{A.13}$$

where P is the homography between the two images. Therefore,

$$P \sim V'(R + T'\begin{bmatrix} a & b & c \end{bmatrix})V^{-1} \tag{A.14}$$

A.3.2 Camera motion solver

The robust image registration gives an accurate estimation of the dominant homography between two images. The support region (non-outliers of RANSAC output) corresponds to a real or virtual

planar surface in the scene. Given the camera focal lengths (refer to Section A.4.3 for the recovery of the unknown focal lengths), the camera motion and the plane equation can be solved directly according to Equation (A.14). The camera rotation matrix R is expressed by Euler angles which have 3 variables, the camera translation T' and the plane distance are up to scale, therefore, they have 5 variables in combination. Since the Euler representation of R is non-linear, the Levenberg-Marquardt method is used to solve the above equation. As the number of variables (8 parameters) is small, the optimization process is rapid and stable.

A.3.3 Scene depth solver

The camera motion solver provides the rotation and the translation between the two image coordinate systems. According to Equation (A.10), we have,

$$\mathbf{x}' \sim M\mathbf{x} + w\mathbf{t} \tag{A.15}$$

where $M = V'RV^{-1}$ and $\mathbf{t} = \frac{1}{f}V'T'$ are known. The Levenberg-Marquardt method is used here to minimize:

$$E(w_i) = \sum_i \left[I_0(\mathbf{x}_i) - \alpha I_1(M\mathbf{x}_i + w_i\mathbf{t}) - \beta\right]^2 \tag{A.16}$$

Assuming that the depths of different pixels are independent, we get the diagonal Hessian matrix which makes the optimization process more efficient.

The hierarchical framework used in the homography computation is also applied here. To decrease the possibility of converging to local minima and to improve the efficiency, we use patch-based depth recovery and local search. The image is divided into small patches. Each patch shares the same depth while the patch Jacobian is the sum of the Jacobian of each pixel in the patch. When the patch displacement exceeds a certain scale, even the multilevel depth recovery fails. To overcome this problem, local search is performed at each patch for subpixel displacement. This displacement is used to solve w_i directly and the solution is incorporated into the optimization as initial values.

Figure A.1(d) demonstrates the depth map recovered from the two images in Figure A.1(a) and (b). The darker parts denote the scenes farther from the camera. The image size is 256×240. We use the patch size of 2×2 pixels and the local search area of 7×7 pixels.

A.4 Temporal integration over image sequences

An image sequence stores a large amount of redundant information of scenes as the temporal consistency. We use the information integrated over image sequences to refine the recovered scene depth and take advantage of the depth map to get a better image registration for motion detection.

A.4.1 Depth integration

From each pair of consecutive images, we recover the scene depth represented in the first image coordinate system. It is necessary to propagate this depth representation to the second image coordinate system so that temporal integration can be performed on the recovered depth.

Symmetric to Equation (A.10), we get,

$$\mathbf{x} \sim VR^{-1}V'^{-1}\mathbf{x}' + \frac{w'}{f'}V(-R^{-1}T') \tag{A.17}$$

Take care of the scales in the homogeneous representations of $\mathbf{x}$ and $\mathbf{x}'$,

$$\begin{aligned} k'\mathbf{x}' &= V'RV^{-1}\mathbf{x} + \frac{w}{f}V'T' \\ k\mathbf{x} &= VR^{-1}V'^{-1}\mathbf{x}' + \frac{w'}{f'}V(-R^{-1}T') \end{aligned} \tag{A.18}$$

We obtain,

$$\begin{aligned} k'\mathbf{x}' &= V'RV^{-1}\frac{1}{k}(VR^{-1}V'^{-1}\mathbf{x}' + \frac{w'}{f'}V(-R^{-1}T')) + \frac{w}{f}V'T' \\ &= \frac{1}{k}\mathbf{x}' - \frac{w'}{kf'}V'T' + \frac{w}{f}V'T' \end{aligned} \tag{A.19}$$

that is,

$$(k'k-1)\mathbf{x}' = (\frac{wk}{f} - \frac{w'}{f'})V'T' \tag{A.20}$$

where the 3×1 vector $V'T'$ is the camera motion which is same for all the pixels. Therefore,

$$k'k = 1 \quad \text{and} \quad w' = \frac{f'}{f}kw = \frac{f'w}{fk'} \tag{A.21}$$

In this way we transform the depth w represented in the first image coordinate system to w' represented in the second coordinate system. We can then refine this depth by the next pair of images consisting of the second and the third images. This process is repeated over the entire image sequence.

A.4.2 Plane integration

The first pair of images gives a plane equation from the dominant homography. The plane equation is actually up to scale with the translation parameters. This is the reason why the same scale must be maintained for the same plane in the succeeding pairs in order to refine the current depth. Similar to the depth integration, we need to propagate the plane equation representation from the first image

coordinate system to the second one for temporal integration.

Let $\mathbf{n} = [a \;\; b \;\; c]$ and $\mathbf{n}' = [a' \;\; b' \;\; c']$ denote the equations of the same plane represented in the two image coordinate systems respectively. Since they are the scaled normal directions,

$$\mathbf{n}'^{\mathrm{T}} = \lambda R \mathbf{n}^{\mathrm{T}} \tag{A.22}$$

where R is the rotation between the two coordinate systems and λ is the scale between these two normal directions which is going to be calculated. For point $\mathbf{p} = [x \;\; y \;\; z]^{\mathrm{T}}$ expressed in the first coordinate system, we have,

$$\begin{aligned}
& \mathbf{np} = 1 \quad \text{and} \quad \mathbf{n}'(R\mathbf{p} + T') = 1 \\
\Longrightarrow \quad & \mathbf{n}'R\mathbf{p} - 1 = -\mathbf{n}'T' \\
\Longrightarrow \quad & \lambda\mathbf{np} - 1 = -\lambda\mathbf{n}R^{\mathrm{T}}T' \\
\Longrightarrow \quad & 1 - \frac{1}{\lambda} = -\mathbf{n}R^{\mathrm{T}}T' \\
\Longrightarrow \quad & \lambda = \frac{1}{1 + \mathbf{n}R^{\mathrm{T}}T'}
\end{aligned} \tag{A.23}$$

Therefore, the scale λ and the rotation R propagate the plane position from the first image coordinate system to the second one (Equation (A.22)) so that we can adjust the scale of the camera motion solver for the succeeding pair of images to maintain the plane at the same position.

A.4.3 Focal length recovery

Mohr and Triggs [Mohr and Triggs, 1996] summarized the projective reconstruction approaches and concluded that when the camera intrinsic parameters are constant, three images are enough to recover the Euclidean shape. Pollefeys et al. [Pollefeys *et al.*, 1999] demonstrated that if the skew parameter equals zero, even with varying intrinsic parameters three images are sufficient to recover the Euclidean shape. We assume that all the intrinsic parameters are known except the focal lengths.

Each homography has 8 parameters which include the information of the rotation (3 parameters) and the translation (3 parameters) between the consecutive images. Given the initial values of the first two focal lengths, we can obtain the dominant plane equation from the camera motion solver. The plane equation is propagated to the following images and can then be used to solve the focal lengths from the image homography in the same way as solving the camera motion.

A.4.4 Application to motion detection

In this section we discuss the application of the homography-based method to motion detection.

Detecting moving objects in image sequences taken from moving cameras is an important task in scene analysis. Some algorithms work well in 2D situations when the scene can be approximated by a flat surface and/or when the camera is undergoing only rotations and zooms, and some algorithms can only apply to the scenes when large depth variations are present. Our goal is to perform motion detection in aerial image sequences while the cameras experience complex ego-motion and the scenes can neither be classified as flat surface nor provide significant depth variations.

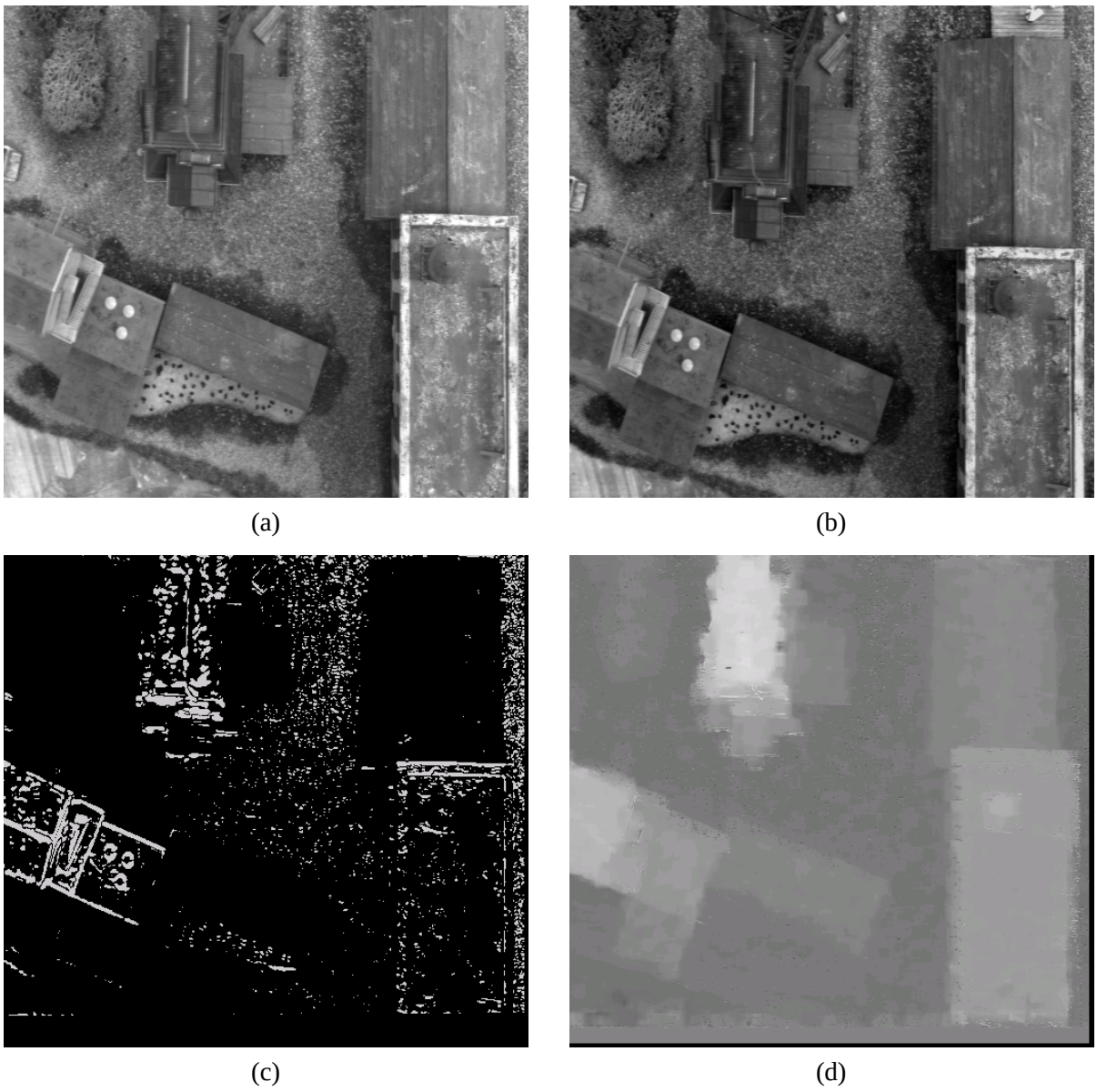
Figure A.2(a) shows three images of the bridge sequence provided by the Video Surveillance and Monitoring (VSAM) project of CMU. The sequence was taken from an airplane flying above a bridge. Two cars were moving on the bridge and one car was moving on the road which was far below the bridge. We first obtained the image homographies to register the consecutive images in the sequence. Figure A.2(b) demonstrates the difference images between the consecutive registered images. White dots indicate the differences which are actually the outliers of the homographies. We can observe that the ground below the bridge was selected as the dominant plane by the robust estimation process. We can also see that both motion (the moving cars) and parallax (the bridge which was closer to the camera than the ground) appear in the difference images. Based on the homographies we recovered the scene depth map by temporal integration over 7 images and used that to register the consecutive images again. Figure A.2(c) shows the recovered depth. It can be seen that the depth map is improved through integration. The recovered depth map of the seventh image shows the scene structure including the bridge in the front and the road along the gully. New difference images (Figure A.2(d)) were generated between the registered images with depth compensation. They show that the differences due to the depth are cleaned up and white dots represent the motion only. Cars on the bridge and on the road below are detected and tracked correctly. However, in the situation where the motion of the object always satisfies the epipolar constraints, the object is classified as a stationary rigid object.

A.5 Discussion

We present a framework for homography-based depth recovery. We first describe a robust homography algorithm which incorporates image contrast/brightness adjustment and robust estimation into image registration. Based on the homography between two images, the camera motion solver gives the solution of the ego-motion and the plane equation, and the solution is refined to generate a dense depth map by the Levenberg-Marquardt method. We also propose the temporal integration of depth recovery and its application to motion detection.

The encouraging temporal integration results motivate us to expand this work to include spatial integration as well. Image homography can be used to generate 2D mosaics [Szeliski and Shum, 1997] and 3D reconstruction from panoramic images always works as the next step [Shum *et al.*, 1998a, Shum *et al.*, 1998b]. The framework described in this appendix presents a way of building 3D mosaics

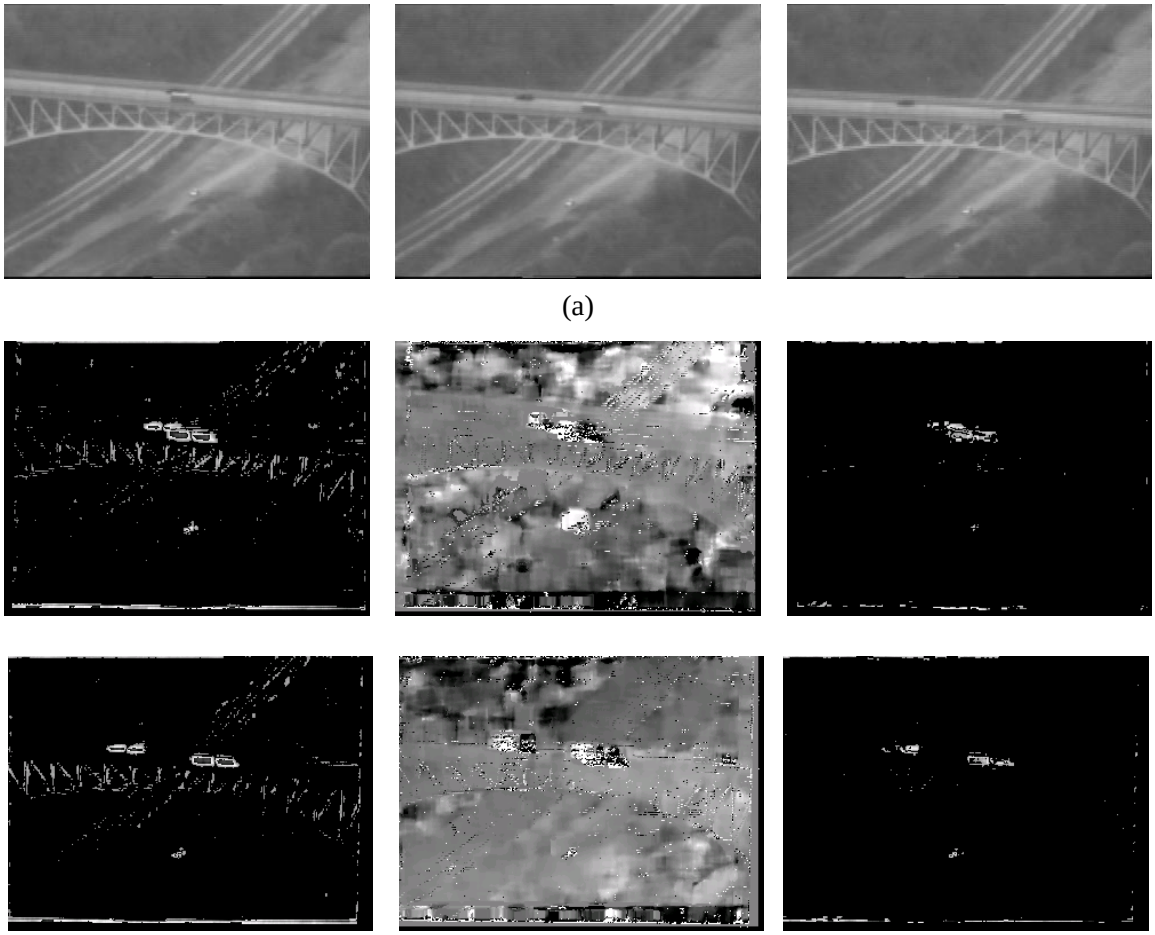
directly from image registration, which makes other application tasks, such as image based rendering and video editing, promising areas to explore. Figure A.3 and Figure A.4 show the 3D mosaics we build for the building sequence and the bridge sequence. The first one is built from the image sequence of 21 images and the second one is from 14 images.



(a) (b)

(c) (d)

Figure A.1: **Robust homography and scene depth.** (a) 1st image, (b) 2nd image of the building sequence. (c) White dots denote the outliers of the robust estimation including the tops of the tall buildings and part of the ground. (d) Recovered depth map (darker denotes farther from the camera).



(a)

(b) (c) (d)

Figure A.2: **Application to motion detection:** (a) 1st, 7th and 11th images of the bridge sequence. (b) 1st and 7th difference images between the registered images. White dots show the differences which are actually the outliers of the homographies. (c) 1st and 7th depth images, darker denotes farther. The depth image is improved through integration. (d) 1st and 7th difference images after the depth compensation. White dots show the differences which correspond to the moving objects while the differences due to the depth are cleaned up.

Figure A.3: **3D mosaic for the building sequence.** This mosaic is built from 21 images.

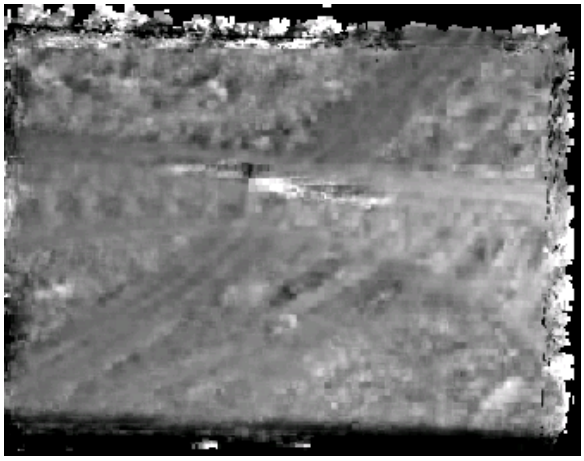


Figure A.4: **3D mosaic for the bridge sequence.** This mosaic is built from 14 images.

Bibliography

[Agapito *et al.*, 1999] L. de Agapito, R.I. Hartley, and E. Hayman. Linear self-calibration of a rotating and zooming camera. In *CVPR99*, pages 15–21, 1999.

[Aguiar and Moura, 1999] P.M.Q. Aguiar and J.M.F. Moura. Factorization as a rank 1 problem. In *CVPR99*, pages I:178–184, 1999.

[Anandan *et al.*, 1994] P. Anandan, K. Hanna, and R. Kumar. Shape recovery from multiple views: A parallax based approach. In *ARPA94*, pages II:947–955, 1994.

[Anandan, 1989] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2(3):283–310, January 1989.

[Avidan and Shashua, 1999] S. Avidan and A. Shashua. Trajectory triangulation of lines: Reconstruction of a 3d point moving along a line from a monocular image sequence. In *CVPR99*, 1999.

[Avidan and Shashua, 2000] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *PAMI*, 22(4):348–357, April 2000.

[Baker *et al.*, 1998] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR98*, pages 434–441, 1998.

[Beardsley *et al.*, 1996] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3d model acquisition from extended image sequences. In *ECCV96*, pages II:683–695, 1996.

[Bergen *et al.*, 1992] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV92*, pages 237–252, 1992.

[Boult and Brown, 1991] T. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Proceedings of the 1991 Visual Motion Workshop*, pages 179–186, 1991.

[Bregler *et al.*, 2000] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *CVPR00*, pages II:690–696, 2000.

[Carlsson and Weinshall, 1998] S. Carlsson and D. Weinshall. Dual computation of projective shape and camera positions from multiple images. *IJCV*, 27(3):227–241, May 1998.

[Christy and Horaud, 1996a] S. Christy and R. Horaud. Euclidean reconstruction: From paraperspective to perspective. In *ECCV96*, pages II:129–140, 1996.

[Christy and Horaud, 1996b] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *PAMI*, 18(11):1098–1104, November 1996.

[Costeira and Kanade, 1998] J.P. Costeira and T. Kanade. A multibody factorization method for independently moving-objects. *IJCV*, 29(3):159–179, 1998.

[Faugeras and Keriven, 1996] O.D. Faugeras and R. Keriven. Variational-principles, surface evolution, pdes, level set methods, and the stereo problem. Technical Report N 3021, INRIA, October 1996.

[Faugeras and Keriven, 1997] O.D. Faugeras and R. Keriven. Level set methods and the stereo problem. In *ScaleSpace97*, 1997.

[Faugeras and Keriven, 1998a] O.D. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *ECCV98*, 1998.

[Faugeras and Keriven, 1998b] O.D. Faugeras and R. Keriven. Variational-principles, surface evolution, pdes, level set methods, and the stereo problem. *IP*, 7(3):336–344, March 1998.

[Faugeras, 1992] O.D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *ECCV92*, pages 563–578, 1992.

[Fischler and Bolles, 1981] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, June 1981.

[Gear, 1998] C.W. Gear. Multibody grouping from motion images. *IJCV*, 29(2):133–150, August 1998.

[Han and Kanade, 1998] M. Han and T. Kanade. Homography-based 3d scene analysis of video sequences. In *DARPA98*, pages 154–160, 1998.

[Han and Kanade, 1999a] M. Han and T. Kanade. The factorization method with linear motions. Technical Report CMU-RI-TR-99-23, Robotics Institute, Carnegie Mellon University, December 1999.

[Han and Kanade, 1999b] M. Han and T. Kanade. Perspective factorization methods for euclidean reconstruction. Technical Report CMU-RI-TR-99-22, Robotics Institute, Carnegie Mellon University, December 1999.

[Han and Kanade, 2000a] M. Han and T. Kanade. Creating 3d models with uncalibrated cameras. In *WACV00*, pages 178–185, 2000.

[Han and Kanade, 2000b] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *CVPR00*, pages II:542–549, 2000.

[Han and Kanade, 2000c] M. Han and T. Kanade. Scene reconstruction from multiple uncalibrated views. Technical Report CMU-RI-TR-00-09, Robotics Institute, Carnegie Mellon University, January 2000.

[Han and Kanade, 2001] M. Han and T. Kanade. Multiple motion scene reconstruction from uncalibrated views. In *ICCV01*, 2001.

[Hartley and Zisserman, 2000] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[Hartley, 1994] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *CVPR94*, pages 908–912, 1994.

[Hartley, 1997] R.I. Hartley. Lines and points in three views and the trifocal tensor. *IJCV*, 22(2):125–140, March 1997.

[Hartley, 1998] R.I. Hartley. Computation of the quadrifocal tensor. In *ECCV98*, pages 20–35, 1998.

[Heyden and Astrom, 1997] A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *CVPR97*, pages 438–443, 1997.

[Heyden, 1997] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *SCIA97*, 1997.

[Heyden, 1998] A. Heyden. Reduced multilinear constraints: Theory and experiments. *IJCV*, 30(1):5–26, October 1998.

[Irani and Anandan, 1996] M. Irani and P. Anandan. Parallax geometry of pairs of points for 3d scene analysis. In *ECCV96*, pages 17–30, 1996.

[Irani and Anandan, 1998] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *PAMI*, 20(6):577–589, June 1998.

[Irani and Anandan, 2000] M. Irani and P. Anandan. Factorization with uncertainty. In *ECCV00*, 2000.

[Irani *et al.*, 1992] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *ECCV92*, pages 282–287, 1992.

[Irani *et al.*, 1997] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *PAMI*, 19(3):268–272, March 1997.

[Irani *et al.*, 1998] M. Irani, P. Anandan, and D. Weinshall. From reference frames to reference planes: Multi-view parallax geometry and applications. In *ECCV98*, pages 829–845, 1998.

[Irani *et al.*, 1999] M. Irani, P. Anandan, and M. Cohen. Direct recovery of planar-parallax from multiple frames. In *Vision Algorithms Theory and Practice*, pages 85–98, 1999.

[Irani, 1999] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *ICCV99*, pages 626–633, 1999.

[Jacobs, 1997] D.W. Jacobs. Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *CVPR97*, pages 206–212, 1997.

[Kahl and Triggs, 1999] F. Kahl and B. Triggs. Critical motions in euclidean structure from motion. In *CVPR99*, pages II:366–372, 1999.

[Kahl *et al.*, 2000] F. Kahl, B. Triggs, and K. Astrom. Critical motions for autocalibration when some intrinsic parameters can vary. *Journal of Math. Imaging and Vision*, 13(2), 2000.

[Kanatani and Morris, 2000] K. Kanatani and D.D. Morris. Gauges and gauge transformations in 3-d reconstruction from a sequence of images. In *ACCV00*, pages 1046–1051, 2000.

[Kanatani, 1997] K. Kanatani. Introduction to statistical optimization for geometric computation. In *Lecture notes at CMU*, 1997.

[Kumar *et al.*, 1994] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR94*, pages 685–688, 1994.

[Lucas and Kanade, 1981a] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[Lucas and Kanade, 1981b] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *DARPA81*, pages 121–130, 1981.

[Ma and Ahuja, 1998] J. Ma and N. Ahuja. Dense shape and motion from region coorespondences by factorization. In *CVPR98*, pages 219–224, 1998.

[Mahamud and Hebert, 2000] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *CVPR00*, pages II:430–437, 2000.

[McLauchlan, 1999] P.F. McLauchlan. Gauge invariance in projective 3d reconstruction. In *MVIEW99*, 1999.

[McLauchlan, 2000] P.F. McLauchlan. Gauge independence in optimization algorithms for 3d vision. In *Vision Algorithms: Theory and Practice*, pages 183–197, 2000.

[Mohr and Triggs, 1996] R. Mohr and B. Triggs. Projective geometry for image analysis. In *Tutorial given at ISPRS*, 1996.

[Mohr *et al.*, 1995] R. Mohr, L. Quan, and F. Veillon. Relative 3d reconstruction using multiple uncalibrated images. *IJRR*, 14(6):619–632, December 1995.

[Morris and Kanade, 1998] D.D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *ICCV98*, pages 696–702, 1998.

[Morris *et al.*, 1999] D.D. Morris, K. Kanatani, and T. Kanade. Uncertainty modeling for optimal structure from motion. In *IEEE Workshop on Vision Algorithms: Theory and Practice*, pages 33–40, 1999.

[Morris *et al.*, 2000a] D.D. Morris, K. Kanatani, and T. Kanade. 3d model accuracy and gauge fixing. Technical Report CMU-RI-TR-00-32, Robotics Institute, Carnegie Mellon University, December 2000.

[Morris *et al.*, 2000b] D.D. Morris, K. Kanatani, and T. Kanade. Uncertainty modeling for optimal structure from motion. In *Vision Algorithms: Theory and Practice*, pages 200–215, 2000.

[Poelman and Kanade, 1997] C. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *PAMI*, 19(3):206–218, 1997.

[Pollefeys *et al.*, 1999] M. Pollefeys, R. Koch, and L. VanGool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *IJCV*, 32(1):7–25, August 1999.

[Quan and Kanade, 1997] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. *PAMI*, 19(8):834–845, August 1997.

[Quan, 1995] L. Quan. Invariants of 6 points and projective reconstruction from 3 uncalibrated images. *PAMI*, 17(1):34–46, 1995.

[Quan, 1996] L. Quan. Self-calibration of an affine camera from multiple views. *IJCV*, 19(1):93–105, July 1996.

[Sawhney *et al.*, 1999] H. Sawhney, Y. Guo, J. Asmuth, and R. Kumar. Independent motion detection in 3d scenes. In *ICCV99*, pages 612–619, 1999.

[Shashua and Avidan, 1996] A. Shashua and S. Avidan. The rank 4 constraint in multiple (≥ 3) view geometry. In *ECCV96*, pages 196–206, 1996.

[Shashua and Werman, 1995] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *ICCV95*, pages 920–925, 1995.

[Shashua and Wolf, 2000] A. Shashua and L.B. Wolf. Homography tensors: On algebraic entities that represent three views of static or moving planar points. In *ECCV00*, 2000.

[Shashua *et al.*, 1999] A. Shashua, S. Avidan, and M. Werman. Trajectory triangulation over conic sections. In *ICCV99*, pages 330–336, 1999.

[Shum *et al.*, 1995] H.Y. Shum, K. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *PAMI*, 17(9):854–867, September 1995.

[Shum *et al.*, 1998a] H.Y. Shum, M. Han, and R. Szeliski. Interactive construction of 3d models from panoramic mosaics. In *CVPR98*, pages 427–433, 1998.

[Shum *et al.*, 1998b] H.Y. Shum, R. Szeliski, S. Baker, M. Han, and P. Anandan. Interactive 3d modeling from multiple images using scene regularities. In *SMILE98*, 1998.

[Sturm and Triggs, 1996] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *ECCV96*, pages II:709–720, 1996.

[Sturm, 1997a] P. Sturm. Critical motion sequences and conjugacy of ambiguous euclidean reconstructions. In *SCIA97*, 1997.

[Sturm, 1997b] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *CVPR97*, pages 1100–1105, 1997.

[Sturm, 1999] P.F. Sturm. Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. In *BMVC99*, pages Multi–View Techniques, 1999.

[Sturm, 2000] P. Sturm. Algorithms for plane-based pose estimation. In *CVPR00*, pages I:706–711, 2000.

[Sun *et al.*, 1999] Z. Sun, V. Ramesh, and A. Tekalp. Error characterization of the factorization approach to shape and motion recovery. In *Vision Algorithms Theory and Practice*, pages 219–233, 1999.

[Szeliski and Shum, 1997] R. Szeliski and H. Shum. Creating full view panoramic image mosaics and texture-mapped models. In *SIGGRAPH97*, pages 251–258, 1997.

[Szeliski, 1996] R. Szeliski. Video mosaics for virtual environments. *IEEE CGA*, 16(2):22–30, March 1996.

[Tomasi and Kanade, 1992] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2):137–154, 1992.

[Torr and Murray, 1993] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. *SPIE*, 2059:432–443, 1993.

[Torr and Murray, 1997] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *IJCV*, 24(3):271–300, September 1997.

[Triggs *et al.*, 2000] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzibbon. Bundle adjustment – a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–375, 2000.

[Triggs, 1995] B. Triggs. Matching constraints and the joint image. In *ICCV95*, pages 338–343, 1995.

[Triggs, 1996] B. Triggs. Factorization methods for projective structure and motion. In *CVPR96*, pages 845–851, 1996.

[Triggs, 1997] B. Triggs. Autocalibration and the absolute quadric. In *CVPR97*, pages 609–614, 1997.

[Tsai, 1987] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *RA*, 3(4):323–344, 1987.

[Urban *et al.*, 1999] M. Urban, T. Pajdla, and V. Hlavac. Projective reconstruction from n views having one view in common. In *Vision Algorithms Theory and Practice*, pages 116–130, 1999.

[Wexler and Shashua, 2000] Y. Wexler and A. Shashua. On the synthesis of dynamic scenes from reference views. In *CVPR00*, pages II:576–581, 2000.

[Xiong and Shafer, 1995] Y. Xiong and S.A. Shafer. Dense structure from a dense optical flow sequence. In *SCV95*, pages 1–6, 1995.

[Yu *et al.*, 1996] H. Yu, Q. Chen, G. Xu, and M. Yachida. 3d shape and motion by svd under higher-order approximation of perspective projection. In *ICPR96*, page A80.22, 1996.

[Zelnik-Manor and Irani, 1999] L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *ICCV99*, pages 710–715, 1999.