# Robust and Accurate Structure from Motion of Rigid and Nonrigid Objects

by

Guanghui Wang

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Guanghui Wang

# Abstract

As a central theme in computer vision, the problem of 3D structure and motion recovery from image sequences has been widely studied during the past three decades, and considerable progress has been made in theory, as well as in practice. However, there are still several challenges remaining, including algorithm robustness and accuracy, especially for nonrigid modeling. The thesis focuses on solving these challenges and several new robust and accurate algorithms have been proposed.

The first part of the thesis reviews the state-of-the-art techniques of structure and motion factorization. First, an introduction of structure from motion and some mathematical background of the technique is presented. Then, the general idea and different formulations of structure from motion for rigid and nonrigid objects are discussed.

The second part covers the proposed quasi-perspective projection model and its application to structure and motion factorization. Previous algorithms are based on either a simplified affine assumption or a complicated full perspective projection model. The affine model is widely adopted due to its simplicity, whereas the extension to full perspective suffers from recovering projective depths. A quasi-perspective model is proposed to fill the gap between the two models. It is more accurate than the affine model from both theoretical analysis and experimental studies. More geometric properties of the model are investigated in the context of one- and two-view geometry. Finally, the model was applied to structure from motion and a framework of rigid and nonrigid factorization under quasi-perspective assumption is established.

The last part of the thesis is focused on the robustness and three new algorithms are proposed. First, a spatial-and-temporal-weighted factorization algorithm is proposed to handle significant image noise, where the uncertainty of image measurement is estimated from a new perspective by virtue of reprojection residuals. Second, a rank-4 affine factorization algorithm is proposed to avoid the difficulty of image alignment with erroneous data, followed by a robust factorization scheme that can work with missing and outlying data. Third,

the robust algorithm is extended to nonrigid scenarios and a new augmented nonrigid factorization algorithm is proposed to handle imperfect tracking data.

The main contributions of the thesis are as follows: The proposed quasi-perspective projection model fills the gap between the simplicity of the affine model and the accuracy of the perspective model. Its application to structure and motion factorization greatly increases the efficiency and accuracy of the algorithm. The proposed robust algorithms do not require prior information of image measurement and greatly improve the overall accuracy and robustness of previous approaches. Moreover, the algorithms can also be applied directly to structure from motion of nonrigid objects.

# Acknowledgements

I would like to thank my supervisor Dr. John Zelek for his extensive support, encouragement, and motivating discussions during my study. I am deeply grateful to Dr. Jonathan Wu for his invaluable guidance and support throughout my entire academic experience in Canada. I would also like to thank Dr. Ruzena Bajcsy for her insightful advice and kind help during my visit at the University of California, Berkeley.

Many thanks to all members of my committee, Dr. David Clausi, Dr. Alexander Wong, Dr. Richard Mann, Dr. Zhou Wang, and Dr. John Barron for their helpful and insightful comments and suggestions to the thesis.

I would like to extend my gratitude to all faculty and staff members at the Department of Systems Design Engineering for their readiness to help at all times. I am especially grateful to Vicky Lawrence for her helpful support.

Furthermore, I owe thanks to the staff members at the Centre for Teaching Excellence, Centre for Career Action, and the Student Success Office of the University of Waterloo for their support to my study and career development. I would also like to thank my friends, including Adel Fakih, Dong Li, Jinfu Yang, Dibyendu Mukherjee, Xiangru Li, and many more, who made my experience at Waterloo more fruitful and enjoyable.

In addition, I have been very privileged to get several scholarships to support my study. I convey special acknowledgement to NSERC CGS Scholarship, NSERC CGS-MSFSS, President's Graduate Scholarship of the University of Waterloo, and the Ontario Graduate Scholarship in Science and Technology.

Last but not least, I owe my thanks to my family for their infinite love and support throughout everything. Thank you with all my heart.

# Dedication

*To my wife Junfeng and daughter Nina.*

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Making computers see and understand the world are the main tasks of computer vision. As a central theme in computer vision, the problem of 3D structure and motion recovery from images or video sequences has been widely studied during the past three decades [72].

The classical method for 3D reconstruction is stereo vision using two or three images. Once the correspondences between these images have been established, the 3D structure can be calculated via triangulations. For a sequence of many images, the typical approach is the structure and motion factorization algorithm, which was first proposed by Tomasi and Kanade [74]. The factorization method is based on a bilinear formulation that decomposes image measurements directly into the structure and motion components. The algorithm assumes that the tracking matrix of an image sequence is available and deals uniformly with the data from all images. Thus, it is more robust and more accurate than the methods that use only two images [17][42][57][65][77].

In recent years, considerable progress has been made in theory and practice, resulting in many successful applications in robot navigation and map building, industrial inspection, medical image analysis, reverse engineering, autonomous vehicles, and digital entertainment. However, the problem still remains far from being solved.

In this chapter, the definition and classification of structure from motion (SfM) is presented, followed by the challenges of structure and motion recovery. Then, the contributions of this thesis are summarized.

## 1.1 Problem Definition of SfM

In projective geometry, a point is normally denoted in homogeneous form. Suppose $\bar{\mathbf{X}}_j = [x_j, y_j, z_j]^T$ is a 3D space point, which is projected to a 2D image point

$\bar{\mathbf{x}}_{ij} = [u_{ij}, v_{ij}]^T$ in the $i$-th frame. Their corresponding homogeneous forms are denoted as $\mathbf{X}_j = [\bar{\mathbf{X}}_j, 1]^T$ and $\mathbf{x}_{ij} = [\bar{\mathbf{x}}_{ij}, 1]^T$, respectively.

Under a full perspective projection model, a 3D point is projected to an image point by the following equation.

$$\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j = \mathbf{K}_i[\mathbf{R}_i, \mathbf{t}_i]\mathbf{X}_j \tag{1.1}$$

where $\lambda_{ij}$ is a nonzero depth scale; $\mathbf{P}_i$ is a $3 \times 4$ projection matrix of the $i$-th camera; $\mathbf{R}_i$ and $\mathbf{t}_i$ are the corresponding rotation matrix and translation vector of the camera with respect to the world system; and $\mathbf{K}_i$ is the camera calibration matrix. When the object is far away from the camera with relatively small depth variation, one may safely assume a simplified affine camera model as below to approximate the perspective projection.

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i\bar{\mathbf{X}}_j + \bar{\mathbf{t}}_i \tag{1.2}$$

where $\mathbf{A}_i$ is a $2 \times 3$ affine projection matrix; the image point $\bar{\mathbf{x}}_{ij}$ and the space point $\bar{\mathbf{X}}_j$ are denoted in nonhomogeneous form. Under affine projection model, the mapping from space to the image becomes linear since the unknown depth scalar $\lambda_{ij}$ in (1.1) is eliminated. If all image points of each frame are registered to the centroid of that image and a relative image coordinate system is adopted, the translation term $\bar{\mathbf{t}}_i$ will vanish. Thus, the affine projection Equation (1.2) is further simplified to

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i\bar{\mathbf{X}}_j. \tag{1.3}$$

Suppose we have an image sequence of $m$ frames and a set of $n$ feature points tracked across the sequence. The coordinates of the tracked features are denoted as $\{\bar{\mathbf{x}}_{ij} = \begin{bmatrix} u_{ij} \\ v_{ij} \end{bmatrix} | i = 1, ..., m, j = 1, ..., n\}$, we can arrange these tracking data into a compact matrix as

$$\mathbf{W} = \begin{matrix} \texttt{frames} \\ \downarrow \end{matrix} \overbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}^{\texttt{points}}_{2m \times n} = \begin{bmatrix} \begin{bmatrix} u_{11} \\ v_{11} \end{bmatrix} & \cdots & \begin{bmatrix} u_{1n} \\ v_{1n} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} u_{m1} \\ v_{m1} \end{bmatrix} & \cdots & \begin{bmatrix} u_{mn} \\ v_{mn} \end{bmatrix} \end{bmatrix} \tag{1.4}$$

where $\mathbf{W}$ is called the tracking or measurement matrix, which is a $2m \times n$ matrix composed of all tracked features across the sequence.

Under the perspective projection (1.1), the tracking data can be denoted in

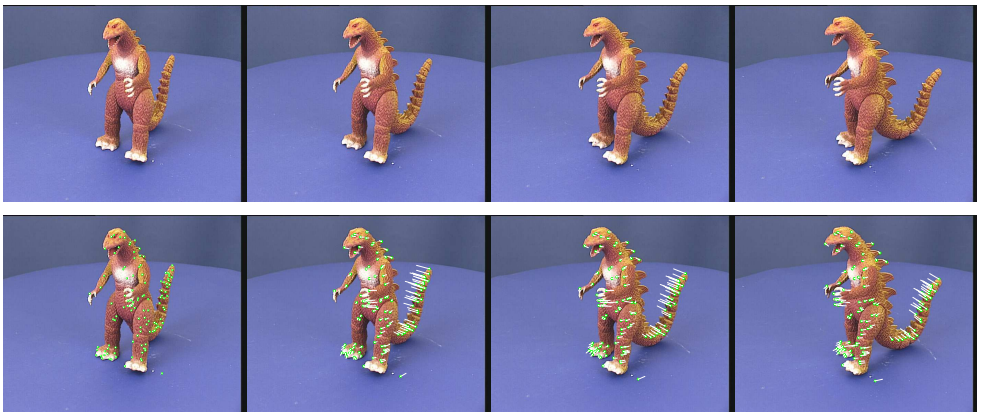


Figure 1.1: Four consecutive frames from the dinosaur sequence, where 115 tracked features are overlaid on the images in the second row, and the white lines denote the relative disparities between consecutive frames. Courtesy of Andrew Fitzgibbon and Andrew Zisserman.

the following homogeneous form by including the depth scales.

$$\dot{\mathbf{W}} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix}_{3m\times n} = \begin{bmatrix} \lambda_{11}\begin{bmatrix} u_{11} \\ v_{11} \\ 1 \end{bmatrix} & \cdots & \lambda_{1n}\begin{bmatrix} u_{1n} \\ v_{1n} \\ 1 \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\begin{bmatrix} u_{m1} \\ v_{m1} \\ 1 \end{bmatrix} & \cdots & \lambda_{mn}\begin{bmatrix} u_{mn} \\ v_{mn} \\ 1 \end{bmatrix} \end{bmatrix}. \quad (1.5)$$

We call $\dot{\mathbf{W}}$ the projective-depth-scaled or weighted tracking matrix. The depth scale $\lambda_{ij}$ is usually unknown.

Fig. 1.1 shows an example of the feature tracking result for a dinosaur toy sequence, which contains 36 frames that are taken evenly around a turn table. The images and tracking data were downloaded from the Visual Geometry Group of Oxford University. Feature tracking has been an active research topic since the beginning of computer vision. Many effective methods have been proposed, such as the Kanade-Lucas-Tomasi (KLT) Feature Tracker [46][69], stereo vision matching via epipolar geometry [106], scale-invariant feature transform (SIFT) [45], and many more. Most approaches in SfM usually assume all features are tracked across the entire sequence.

The problem of structure from motion is defined as: Given $n$ tracked features across a sequence of $m$ frames, we want to recover the 3D Euclidean structure

Table 1.1: Classification of structure and motion factorization of rigid and nonrigid objects, where 'Affine' and 'Persp.' stand for affine and perspective projection models, respectively.

| Classification | | Tracking | Motion matrix | Shape matrix | Upgrading matrix |
|---|---|---|---|---|---|
| Rigid | Affine | $\mathbf{W} \in \mathbb{R}^{2m\times n}$ | $\mathbf{M} \in \mathbb{R}^{2m\times 3}$ | $\bar{\mathbf{S}} \in \mathbb{R}^{3\times n}$ | $\mathbf{H} \in \mathbb{R}^{3\times 3}$ |
| | Persp. | $\dot{\mathbf{W}} \in \mathbb{R}^{3m\times n}$ | $\mathbf{M} \in \mathbb{R}^{3m\times 4}$ | $\mathbf{S} \in \mathbb{R}^{4\times n}$ | $\mathbf{H} \in \mathbb{R}^{4\times 4}$ |
| Nonrigid | Affine | $\mathbf{W} \in \mathbb{R}^{2m\times n}$ | $\mathbf{M} \in \mathbb{R}^{2m\times 3k}$ | $\bar{\mathbf{B}} \in \mathbb{R}^{3k\times n}$ | $\mathbf{H} \in \mathbb{R}^{3k\times 3k}$ |
| | Persp. | $\dot{\mathbf{W}} \in \mathbb{R}^{3m\times n}$ | $\mathbf{M} \in \mathbb{R}^{3m\times(3k+1)}$ | $\mathbf{B} \in \mathbb{R}^{(3k+1)\times n}$ | $\mathbf{H} \in \mathbb{R}^{(3k+1)\times(3k+1)}$ |

$\mathbf{S}_{ij} = \{\mathbf{X}_{ij}\}$ for $i = 1, \cdots, m, j = 1, \cdots, n$, and the motion parameters $\{\mathbf{R}_i, \mathbf{t}_i\}$ of the camera corresponding to each frame. The factorization based algorithm has been proved to be an effective method to deal with this problem. According to the property of the object and the camera model employed, the algorithm can be classified generally into the following categories.

1. Rigid objects under affine assumption;
2. Rigid objects under perspective projection model;
3. Nonrigid objects under affine projection model;
4. Nonrigid objects under perspective projection model.

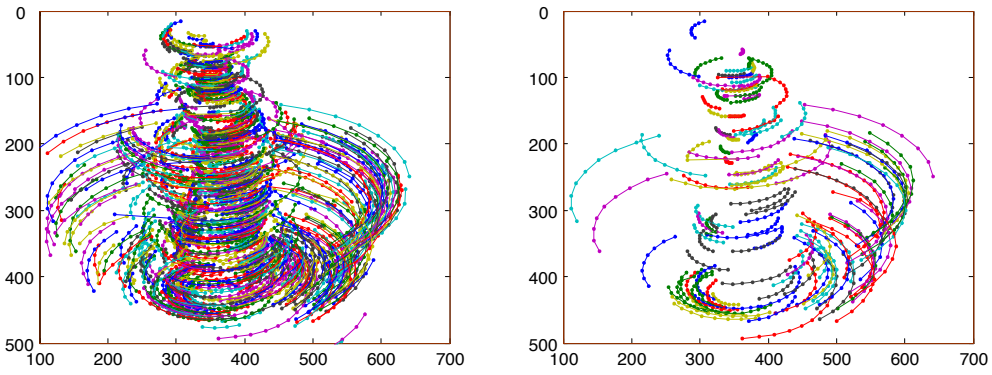
Table 1.1 shows the differences between different factorization algorithms, where the meaning of some symbols, such as $\mathbf{M}, \mathbf{S}, \mathbf{B}$, and $\mathbf{H}$ will be defined and discussed in the following chapters.

## 1.2 Challenges in SfM

Although great progress has been made in structure and motion recovery from image sequences, many problems have not been effectively solved by the available approaches, especially when the data are corrupted by outliers, missing features, and significant noise. The major challenges in SfM are as follows.

1. Trade-off between accuracy and efficiency

   Most factorization algorithms are based on the affine assumption due to its simplicity, although it is just a zero-order approximation to the real imaging process. The extension to a full perspective model suffers from recovery of the unknown projective depths, which is computationally intensive and no convergence is guaranteed.




(a) (b)

Figure 1.2: The tracked features of the dinosaur sequence, where the consecutive tracked features are connected by colorful lines. (a) The features being tracked in more than four consecutive frames; and (b) the features being tracked in more than eight consecutive frames.

2. Accurate solution in presence of significant noise

   General factorization methods usually assume error-free tracking data, however, the performance will degenerate in the presence of significant noise or errors. Researchers proposed to use weighted factorization to handle noisy and erroneous data, however, the weight matrix is hard to retrieve and usually unavailable in many situations.

3. Robust algorithm in presence of outlying and missing data

   Outliers and missing data are inevitable during the process of feature tracking. Figure 1.2 shows an example of the missing features in tracking. Existing approaches usually adopt RANSAC or other hypothesis-and-test algorithms to detect outliers. However, these methods are computationally intensive and can only be applied to rigid factorization. Few reports in the literature can handle outliers for nonrigid factorization.

## 1.3 Thesis Contributions

The thesis is endeavored to solve the challenges of accuracy and robustness for SfM. It is not only academically significant, but also significant to many applications in 3D modeling, such as robot navigation, motion analysis, and visual surveillance. Below are the major contributions of the thesis.

First, we propose a quasi-perspective projection model and a structure and motion factorization framework based on that model. Previous algorithms are either based on a simplified affine assumption or a complicated full perspective projection model. The affine model is widely adopted due to its simplicity, whereas its extension to full perspective suffers from recovering projective depths. The proposed quasi-perspective model fills the gap between the simplicity of the affine model and the accuracy of the perspective model. It is more accurate than the affine model from both theoretical analyses and experimental studies. The geometric properties of the model are investigated in the context of one- and two-view geometry. Finally, the model is applied to the SfM framework of rigid and nonrigid objects.

Second, three robust SfM algorithms are proposed to handle the data that are corrupted by outliers, missing features, and significant noise: (i) a spatial-and-temporal-weighted factorization algorithm is designed to handle significant noise contained in the tracking data, where the uncertainty of image measurement is estimated from a new perspective by virtue of reprojection residuals; (ii) an augmented affine factorization algorithm is proposed to handle outlying and missing data for rigid scenarios. As a new addition to previous affine factorization family based on rank-3 constraint, the algorithm avoids the difficulty of image alignment; and (iii) the augmented affine factorization algorithm is successfully extended to nonrigid scenarios in the presence of imperfect tracking data.

## 1.4 Thesis Organization

The rest of this thesis is organized as follows.

Chapter 2 presents the state-of-the-art techniques for the problem of structure from motion, including a brief introduction of different formulations of structure and motion factorization under different projection models.

Chapter 3 describes the affine approximation of a full perspective projection model, and proposes a quasi-perspective projection model under the assumption that the camera is far away from the object with small lateral rotations.

In Chapter 4, the geometrical properties of the quasi-perspective projection model are investigated in the context of one- and two-view geometry, including quasi-perspective projection matrix, fundamental matrix, plan induced homograph, and quasi-perspective reconstruction.

Chapter 5 discusses a quasi-perspective-model-based algorithm for structure and motion recovery of both rigid and nonrigid objects, and establishes a framework of rigid and nonrigid factorization under quasi-perspective assumption. Furthermore, the computation details on how to upgrade the solutions to the Euclidean space are presented in this chapter.

In Chapter 6, a spatial-and-temporal-weighted factorization algorithm is proposed to handle significant noise contained in the tracking data. In the algorithm, the error distribution is estimated from a new perspective by virtue of reprojection residuals, and the image errors are modeled both spatially and temporally to cope with different kinds of uncertainties.

Chapter 7 describes a robust structure from motion algorithm for rigid objects. First, a rank-4 augmented affine factorization algorithm is proposed to overcome the difficulty in image alignment for imperfect data. Then, a robust structure and motion factorization scheme is proposed to handle outlying and missing data.

The robust algorithm is further extended to nonrigid scenarios in Chapter 8, where the outliers can be detected directly from image reprojection residuals of nonrigid factorization. A new augmented nonrigid factorization algorithm is proposed in this chapter, followed by a robust scheme to handle imperfect tracking data of nonrigid image sequence.

Chapter 9 summarizes the thesis conclusion and presents a discussion of potential research work for future study.

# Chapter 2

# Background

In this chapter, the state-of-the-art techniques for the problem of structure from motion are presented, followed by a brief introduction of different formulations of structure and motion factorization, including the factorization algorithms of rigid and nonrigid objects based on different camera models.

## 2.1 Literature Review

In this section, we will present a review of the structure and motion factorization algorithms, including the factorization of rigid objects and static scenes, nonrigid objects and dynamic scenes, and robust algorithms.

### 2.1.1 Factorization for rigid objects and static scenes

The original factorization algorithm assumes an orthographic projection model and works only for rigid objects and static scenes [74]. The main idea of the algorithm is to decompose the tracking matrix into the motion and structure components simultaneously by Singular Value Decomposition (SVD) with low-rank approximation. It was extended to weak perspective and paraperspective projection by Poelman and Kanade [62]. The orthographic, weak perspective, and paraperspective projections can be generalized with the affine camera model, which is a zero order (for weak-perspective) or first order (for paraperspective) approximation of a full perspective projection [34][62][65].

More generally, Christy and Horaud [15] extended the algorithm to a perspective camera model by incrementally performing the factorization under affine assumption. The method is an affine approximation to full perspective projection. Triggs and Sturm [71][77] proposed a full projective reconstruction method via rank-4 factorization of a scaled tracking matrix, where the unknown depth scales

were recovered from pairwise epipolar geometry. The method was further studied in Heyden *et al.* [37] and Mahamud and Hebert [47], and different iterative schemes were proposed to recover the projective depths by minimizing reprojection errors. Recently, Oliensis and Hartley [56] provided a complete theoretical convergence analysis for the iterative extensions. Unfortunately, no iteration has been shown to converge sensibly, and they proposed a simple extension, called CIESTA, to give a reliable initialization to other algorithms.

Almost all factorization algorithms are limited in handling the tracking data of point features. Alternatively, Quan and Kanade [64] proposed an analogous factorization algorithm for line features under affine assumption. The algorithm decomposes the whole structure and motion into three substructures which can be solved linearly via factorization of appropriate measurement matrix. The line-based factorization requires at least seven lines in three views. Whereas the point-based algorithm only needs a minimum of four points in three frames.

### 2.1.2 Factorization for nonrigid objects and dynamic scenes

In real world, many scenarios are nonrigid or dynamic, such as articulated motion [58], human faces carrying different expressions, lip movements, hand gesture, and moving vehicles, etc. In order to deal with these situations, many extensions stemming from the factorization algorithm were proposed to relax the rigidity constraint.

Costeira and Kanade [16] first discussed how to recover the motion and shape of several independent moving objects via factorization using orthographic projection. Bascle and Blake [8] proposed a method for factorizing facial expressions and poses based on a set of preselected basis images. Recently, Li *et al.* [44] proposed to segment multiple rigid-body motions from point correspondences via subspace separation. Yan and Pollefeys [97][98] proposed a factorization-based approach to recover the structure and kinematic chain of articulated objects. Zelnik-Manor and Irani [101][102] analyzed the problem of multi-sequence factorization of multiple objects by both temporal synchronization of sequences and spacial matching across sequences. Del Bue and Agapito [20] proposed a scheme for nonrigid stereo factorization. Taylor *et al.* [73] proposed a framework to solve unknown body deformation under orthography by introducing a constraint of locally-rigid motion.

In the seminal work by Bregler *et al.* [11], it is demonstrated that the 3D shape of a nonrigid object can be expressed as a weighted linear combination of a set of shape bases. Then, the shape bases and camera motions are factorized simultaneously for all time instants under the rank constraint of the tracking matrix. Following this idea, the method was extensively investigated and developed

by many researchers, such as Brand [9][10], Del Bue *et al.* [21][18], Torresani *et al.* [75][76], and Xiao *et al.* [95][96]. Recently, Rabaud and Belongie [66] relaxed the Bregler's assumption [11] by assuming that only small neighborhoods of shapes are well-modeled by a linear subspace; and proposed a novel approach to solve the problem by adopting a manifold-learning framework. He *et al.* [36] proposed an prior-free approach for nonrigid structure and motion recovery, where no extra prior knowledge about the nonrigid scene or the camera motions were assumed.

Most nonrigid factorization methods are based on affine camera model due to its simplicity. It was extended to perspective projection in [96] by iteratively recovering the projective depths. The perspective factorization is more complicated and does not guarantee its convergence to the correct depths, especially for nonrigid scenarios [32]. Vidal and Abretske [78] proposed that the constraints among multiple views of a nonrigid shape consisting of $k$ shape bases can be reduced to multilinear constraints. They presented a closed form solution to the reconstruction of a nonrigid shape consisting of two shape bases. Hartley and Vidal [34] proposed a closed form solution to the nonrigid shape and motion with calibrated cameras or fixed intrinsic parameters.

Since the factorization is only defined up to a nonsingular transformation matrix, many researchers adopt metric constraints to recover the matrix and upgrade the factorization results to the Euclidean space [9][11][18][76]. However, the rotation constraint may cause ambiguity in the combination of shape bases. Xiao *et al.* [95] proposed a basis constraint to solve the ambiguity and provide a closed-form solution to the problem.

### 2.1.3 Robust structure and motion factorization

Most factorization methods usually assume all features are reliably tracked across the sequence. In the presence of missing data, SVD factorization can not be used directly, and some researchers have proposed to solve the motion and shape matrices alternatively while maintaining the other matrix fixed, such as the alternative factorization [49], power factorization [33][79][82], and factor analysis [29]. In practice, outlying data are inevitable during the process of feature tracking; as a consequence, the performance of the algorithm will degenerate. The most popular strategy in the computer vision field for solving this type of problem is RANSAC (RANdom SAmple Consensus) [26], Least Median of Squares (LMedS) [32], and similar hypothesise-and-test frameworks [14][52][67]. However, these methods are preliminary designed for two or three views, they are not suitable for long sequence factorization due to its highly intensive computational cost.

In recent years, the problem of robust factorization has received a lot of attention, and some practical methods have been proposed to handle noisy and erroneous data [100]. Aguitar and Moura [3] proposed a scalar-weighted SVD algorithm by minimizing the weighted square errors. Anandan and Irani [7] proposed a covariance-weighted factorization to factorize noisy correspondences with a high degree of directional uncertainty. Gruber and Weiss [29] formulated the problem as a factor analysis and derived an Expectation Maximization (EM) algorithm to incorporate prior knowledge and enhance the robustness to missing data and uncertainties. Zelnik-Manor *et al.* [103] defined a new type of motion consistency based on temporal consistency; and applied it to multi-body factorization with directional uncertainty.

Zaharescu and Horaud [100] introduced a Gaussian mixture model and incorporated it with the EM algorithm, an approach that is resilient to outliers. Huynh *et al.* proposed an iterative approach to correct the outliers with 'pseudo' observations. Ke and kanade [41] proposed a robust algorithm to handle outliers by minimizing the $L1$ norm of the reprojection errors. Eriksson and Hengel [23] introduced the $L1$ norm minimization to the Wiberg algorithm to handle missing data and outliers. Buchanan and Fitzgibbon [12] presented a comprehensive comparison on a number of factorization algorithms. Their study strongly supports second order nonlinear optimization strategy.

Okatani *et al.* [55] proposed to incorporate a damping factor into the Wiberg method to solve the problem. Yu *et al.* [99] presented a quadratic program (QP) formulation for robust multi-model fitting of geometric structures in vision data. Wang *et al.* [92] proposed an adaptive kernel-scale weighted hypotheses (AKSWH) to segment multiple-structure data even in the presence of a large number of outliers. Paladini *et al.* [60] proposed an alternating bilinear approach to solve nonrigid SfM by introducing a globally optimal projection step of the motion matrices into the manifold of metric constraints. Additional studies are referenced in Aanæs *et al.* [1] and Okatani and Deguchi [54].

## 2.2 Structure and Motion Recovery of Rigid Objects

In this section, we will introduce the structure and motion factorization algorithm for rigid objects under affine and perspective camera models.

### 2.2.1 Rigid factorization under affine projection

Suppose the image points in each frame are registered to the corresponding centroid. Under affine projection model, the imaged points in the $i$-th frame are

formulated as

$$[\bar{\mathbf{x}}_{i1}, \bar{\mathbf{x}}_{i2}, \cdots, \bar{\mathbf{x}}_{in}] = \mathbf{A}_i[\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \cdots, \bar{\mathbf{X}}_n],\ \forall\, i = 1, \cdots, m. \tag{2.1}$$

Stacking the equations (2.1) for all frames together, we can obtain

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_m \end{bmatrix}}_{\mathbf{M}_{2m\times 3}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1, & \cdots, & \bar{\mathbf{X}}_n \end{bmatrix}}_{\bar{\mathbf{S}}_{3\times n}}. \tag{2.2}$$

The above equation can be written concisely as $\mathbf{W} = \mathbf{M}\bar{\mathbf{S}}$, which is called the general factorization expression under affine projection assumption. Suppose the tracking data across the sequence of $m$ frames are available, i.e. the tracking matrix $\mathbf{W}$ is given, our purpose is to recover the motion matrix $\mathbf{M}$ and the rigid shape matrix $\bar{\mathbf{S}}$.

#### SVD decomposition with rank constraint

The tracking matrix is a $2m \times n$ matrix with highly rank-deficiency. From the right side of (2.2), we can easily find that the rank of the tracking matrix is at most 3 for noise-free data since both $\mathbf{M}$ and $\bar{\mathbf{S}}$ are at most of rank 3. However, when the data is corrupted by image noise, the rank of $\mathbf{W}$ will be greater than 3. Here we use the concept of SVD decomposition to obtain the rank-3 approximation and factorize the tracking matrix into the motion and shape matrices.

Without loss of generosity, we assume $2m \geq n$ and perform SVD decomposition on the tracking matrix.

$$\mathbf{W} = \mathbf{U}_{2m\times n}\Sigma_{n\times n}\mathbf{V}^T_{n\times n} \tag{2.3}$$

where $\Sigma = diag(\sigma_1, \sigma_2, \cdots, \sigma_n)$ is a diagonal matrix with all diagonal entries composed by the singular values of $\mathbf{W}$ and arranged in descending order as $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$; and $\mathbf{U}$ and $\mathbf{V}$ are $2m \times n$ and $n \times n$ orthogonal matrices respectively. Thus, $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_n$ with $\mathbf{I}_n$ an $n \times n$ identical matrix. It is noted that the assumption $2m \geq n$ is not crucial, we can obtain a similar decomposition when $2m < n$ by simply taking a transpose of the tracking matrix.

In ideal case, $\mathbf{W}$ is of rank 3, which is equivalent to $\sigma_4 = \sigma_5 = \cdots = \sigma_n = 0$. However, when the data is contaminated by noise, the rank of $\mathbf{W}$ is definitely greater than 3. Actually, the rank may also be greater than 3 even for noise free data since the affine camera model is just an approximation of the real imaging process. We will now seek a rank-3 matrix $\mathbf{W}'$ that can best approximate the tracking matrix. Let us partition the matrices $\mathbf{U}, \Sigma$, and $\mathbf{V}$ as follows:

$$\mathbf{U} \quad = \quad [\mathbf{U}'_{2m\times 3} | \mathbf{U}''_{2m\times(n-3)}],$$

$$\begin{aligned}\Sigma &= \begin{bmatrix}\Sigma'_{3\times 3} & \mathbf{0}\\ \mathbf{0} & \Sigma''_{(n-3)\times(n-3)}\end{bmatrix}, \\ \mathbf{V} &= [\mathbf{V}'_{n\times 3}|\mathbf{V}''_{n\times(n-3)}].\end{aligned} \tag{2.4}$$

Then, the SVD decomposition (2.3) can be written as

$$\mathbf{W} = \underbrace{\mathbf{U}'\Sigma'\mathbf{V}'^T}_{\mathbf{W}'} + \underbrace{\mathbf{U}''\Sigma''\mathbf{V}''^T}_{\mathbf{W}''} \tag{2.5}$$

where $\Sigma' = diag(\sigma_1, \sigma_2, \sigma_3)$ contains the first three greatest singular values of the tracking matrix, $\mathbf{U}'$ is the first three columns of $\mathbf{U}$, and $\mathbf{V}'^T$ is the first three rows of $\mathbf{V}^T$. It is easy to prove that $\mathbf{W}' = \mathbf{U}'\Sigma'\mathbf{V}'^T$ is the best rank-3 approximation of $\mathbf{W}$ in the Frobenius norm. Now let us define

$$\tilde{\mathbf{M}} = \mathbf{U}'\Sigma'^{\frac{1}{2}} \tag{2.6}$$

$$\tilde{\mathbf{S}} = \Sigma'^{\frac{1}{2}}\mathbf{V}'^T \tag{2.7}$$

where $\tilde{\mathbf{M}}$ is a $2m \times 3$ matrix and $\tilde{\mathbf{S}}$ is a $3 \times n$ matrix. Then, we have $\mathbf{W}' = \tilde{\mathbf{M}}\tilde{\mathbf{S}}$, a similar form of the factorization expression (2.2). In fact, $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ are one set of the maximum likelihood affine reconstruction of the tracking matrix $\mathbf{W}$. Obviously, the decomposition is not unique since it is defined up to a nonsingular linear transformation matrix $\mathbf{H} \in \mathbb{R}^{3\times 3}$ as $\tilde{\mathbf{M}}\tilde{\mathbf{S}} = (\tilde{\mathbf{M}}\mathbf{H})(\mathbf{H}^{-1}\tilde{\mathbf{S}})$. If we can find a transformation matrix $\mathbf{H}$ that can make

$$\mathbf{M} = \tilde{\mathbf{M}}\mathbf{H} \tag{2.8}$$

exactly corresponds to a metric motion matrix as in (2.2), then the structure $\bar{\mathbf{S}} = \mathbf{H}^{-1}\tilde{\mathbf{S}}$ will be upgraded from affine to the Euclidean space. We call the transformation $\mathbf{H}$ an upgrading matrix, which can be recovered through a metric constraint by enforcing orthogonality on the rotation matrix.

#### Euclidean stratification and reconstruction

Let us assume a simplified camera model with only one parameter, i.e. the focal length $f$. Suppose the upgrading matrix is $\mathbf{H}$, which upgrades the matrix $\tilde{\mathbf{M}}$ in (2.6) to the Euclidean motion matrix as in (2.2). Then, the motion matrix corresponding to frame $i$ can be written as

$$\begin{bmatrix} f\mathbf{r}_{1i}^T \\ f\mathbf{r}_{2i}^T \end{bmatrix} = \mathbf{M}_i = \tilde{\mathbf{M}}_i\mathbf{H} = \begin{bmatrix} \mathbf{m}_{1i}^T \\ \mathbf{m}_{2i}^T \end{bmatrix}\mathbf{H} \tag{2.9}$$

which leads to

$$\begin{bmatrix} \mathbf{m}_{1i}^T \\ \mathbf{m}_{2i}^T \end{bmatrix}\mathbf{H}\mathbf{H}^T\begin{bmatrix}\mathbf{m}_{1i}|\mathbf{m}_{2i}\end{bmatrix} = f^2\begin{bmatrix} \mathbf{r}_{1i}^T \\ \mathbf{r}_{2i}^T \end{bmatrix}\begin{bmatrix}\mathbf{r}_{1i}|\mathbf{r}_{2i}\end{bmatrix} = f^2\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}. \tag{2.10}$$

Let us define $\mathbf{Q} = \mathbf{H}\mathbf{H}^T$, which is a $3 \times 3$ symmetric matrix with 6 unknowns. The following constraints can be obtained from (2.10).

$$\begin{cases} \mathbf{m}_{1i}^T \mathbf{Q} \mathbf{m}_{1i} = \mathbf{m}_{2i}^T \mathbf{Q} \mathbf{m}_{2i} \\ \mathbf{m}_{1i}^T \mathbf{Q} \mathbf{m}_{2i} = \mathbf{m}_{2i}^T \mathbf{Q} \mathbf{m}_{1i} = 0 \end{cases} . \tag{2.11}$$

The constraints (2.11) are called metric or rotation constraints, which yield a set of over-constrained equations for all frames $i = 1, \cdots, m$. Thus, $\mathbf{Q}$ can be calculated linearly via least squares and the upgrading matrix $\mathbf{H}$ is then extracted from $\mathbf{Q}$ using Cholesky decomposition [32]. Finally, the correct metric motion and structure matrices are obtained by applying the upgrading matrix as $\mathbf{M} = \tilde{\mathbf{M}}\mathbf{H}$, $\bar{\mathbf{S}} = \mathbf{H}^{-1}\tilde{\mathbf{S}}$, and the rotation matrices corresponding to each frame are then extracted from $\mathbf{M}$.

It is noted that the above solution is defined only up to an arbitrary rotation matrix since the choice of the world coordinate system is free. In practice, we can simply choose the first frame as a reference, i.e. setting $\mathbf{R}_1 = \mathbf{I}_3$, and register all other frames to it.

The implementation details of the above factorization algorithm are summarized as follows.

1. Register all image points in each frame to their centroid and construct the tracking matrix;
2. Perform rank-3 SVD decompozation on the tracking matrix to obtain a solution of $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ from (2.6) and (2.7);
3. Compute the upgrading matrix $\mathbf{H}$ from (2.11);
4. Recover the metric motion matrix $\mathbf{M} = \hat{\mathbf{M}}\mathbf{H}$ and shape matrix $\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}}$;
5. Retrieve the rotation matrix of each frame from $\mathbf{M}$.

### 2.2.2 Rigid factorization under perspective projection

Many previous studies on rigid factorization adopt an affine camera model due to its simplicity. However, the assumption is valid only when the objects have small depth variation and are far away from the cameras. Otherwise, the algorithm may fail or yield poor results.

Christy and Horaud [15] extended the above methods to a perspective camera model by incrementally performing the factorization under affine assumption. The method is an affine approximation to full perspective projection. Sturm [71] and Triggs and Sturm[77] proposed a full projective reconstruction method via rank-4 factorization of a scaled tracking matrix with projective depths recovered

from pairwise epipolar geometry. The method was further studied in [37][47], where different iterative schemes were proposed to recover the projective depths by minimizing image reprojection errors.

Under the perspective projection (1.1), all imaged points in the $i$-th frame are formulated as

$$[\lambda_{i1}\mathbf{x}_{i1}, \lambda_{i2}\mathbf{x}_{i2}, \cdots, \lambda_{in}\mathbf{x}_{in}] = \mathbf{P}_i[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n],\ \forall\, i = 1, \cdots, m. \tag{2.12}$$

Thus, we can obtain the general perspective factorization expression by gathering the equation (2.12) for all frames as follows.

$$\underbrace{\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix}}_{\dot{\mathbf{W}}_{3m\times n}} = \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{bmatrix}}_{\mathbf{M}_{3m\times 4}} \underbrace{\begin{bmatrix} \mathbf{X}_1, & \cdots, & \mathbf{X}_n \end{bmatrix}}_{\mathbf{S}_{4\times n}}. \tag{2.13}$$

Compared with the affine factorization (2.2), the main differences lie in the dimension and entries of the tracking matrix, as well as the dimension of the motion and shape matrices. Given a set of consistent projective scales $\lambda_{ij}$, the rank of the weighted tracking matrix is at most 4, since the rank of either the motion matrix $\mathbf{M}$ or the shape matrix $\mathbf{S}$ is not greater than 4. For noise contaminated data, $rank(\dot{\mathbf{W}}) > 4$. We can adopt a similar SVD decomposition process as in (2.4) to obtain a best rank-4 approximation of the scale-weighted tracking matrix and factorize it into a $3m \times 4$ motion matrix $\tilde{\mathbf{M}}$ and a $4 \times n$ shape matrix $\tilde{\mathbf{S}}$.

Obviously, this factorization corresponds to a projective reconstruction, which is defined up to a $4 \times 4$ transformation matrix $\mathbf{H}$. Therefore, we need to upgrade the solution from the projective to the Euclidean space. Through the above analysis, we can see that there are essentially two indispensable steps in perspective factorization. One is the computation of the projective depths, the other is the recovery of the upgrading matrix.

## 2.3 Structure and Motion Recovery of Nonrigid Objects

We assumed rigid objects and static scenes in the last section. In the real world, however, many objects do not have fixed structures, such as human faces with different expressions, torsos, and animals bodies, etc. In this section, we will extend the factorization algorithm to the nonrigid and deformable objects. Fig. 2.1 shows the deformable structure of a jellyfish at different time instance.

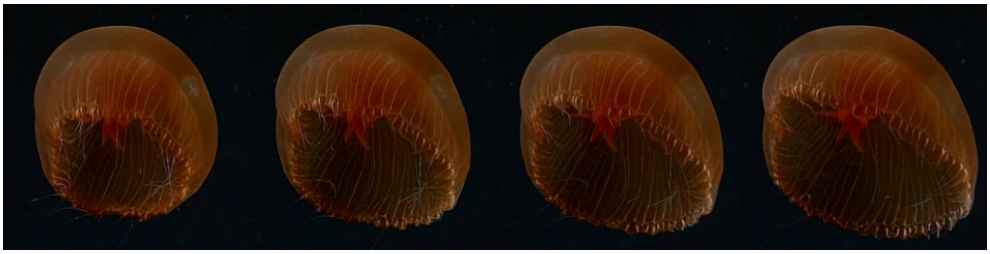


Figure 2.1: The structure of a jellyfish with different deformations. Courtesy of BBC Planet Earth TV series.

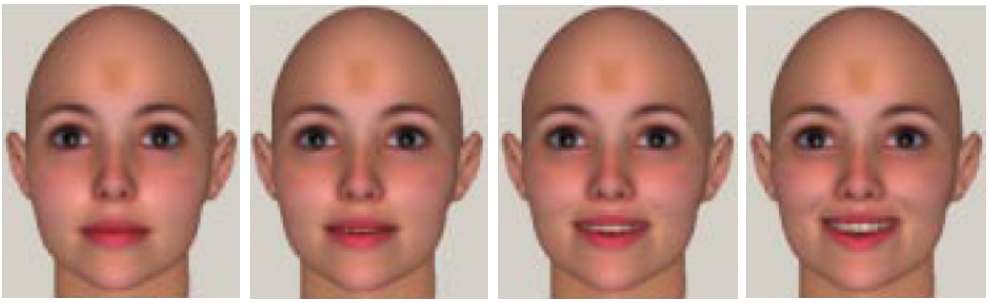


Figure 2.2: Four female face models carrying expressions from neutral to smiling, where any two structure may be taken as shape bases, and the other models can be derived through a weighted linear combinations of the two bases. Courtesy of Jing Xiao.

### 2.3.1 Bregler's deformation model

For nonrigid structure, if all surface features deform randomly at any time instance, there is currently no suitable method to recover its structure from images. Here we restrict our study to a specific kind of deformable objects following the idea first proposed by Bregler *et al.* [11], where the 3D structure of the nonrigid structure is approximated by a weighted combination of a set of shape bases.

Fig. 2.2 shows a very simple example of face models from neutral to smiling with only mouth movements. The deformation can be modeled from two shape bases. If more face expressions, such as joy, sadness, surprise, fear, etc., are involved, more shape bases are needed to model the deformation.

Suppose the deformable structure $\bar{\mathbf{S}}_i \in \mathbb{R}^{3\times n}$ is expressed as a weighted combination of $k$ principal modes of deformation $\mathbf{B}_l \in \mathbb{R}^{3\times n}, l = 1, \cdots, k$. The 3D

model can be expressed as

$$\bar{\mathbf{S}}_i = \sum_{l=1}^{k} \omega_{il} \mathbf{B}_l \tag{2.14}$$

where $\omega_{il} \in \mathbb{R}$ is the deformation weight for base $l$ at frame $i$. A perfect rigid object corresponds to the situation of $k = 1$ and $\omega_{il} = 1$. Suppose all image features are registered to their centroid in each frame, then, we have the following formulation under orthographic projection.

$$\begin{aligned} \mathbf{W}_i &= [\bar{\mathbf{x}}_{i1}, \bar{\mathbf{x}}_{i2}, \cdots, \bar{\mathbf{x}}_{in}] = \mathbf{R}_{Ai} \Big( \sum_{l=1}^{k} \omega_{il} \mathbf{B}_l \Big) \\ &= \big[ \omega_{i1} \mathbf{R}_{Ai}, \cdots, \omega_{ik} \mathbf{R}_{Ai} \big] \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \end{bmatrix}, \forall\, i = 1, \cdots, m \end{aligned} \tag{2.15}$$

where $\mathbf{R}_{Ai}$ stands for the first two rows of the rotation matrix corresponding to the $i$-th frame. Then, we can obtain the factorization equation of the tracking matrix by stacking all instances of equations (2.15) frame by frame as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m \times n}} = \underbrace{\begin{bmatrix} \omega_{11} \mathbf{R}_{A1} & \cdots & \omega_{1k} \mathbf{R}_{A1} \\ \vdots & \ddots & \vdots \\ \omega_{m1} \mathbf{R}_{Am} & \cdots & \omega_{mk} \mathbf{R}_{Am} \end{bmatrix}}_{\mathbf{M}_{2m \times 3k}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \end{bmatrix}}_{\bar{\mathbf{B}}_{3k \times n}}. \tag{2.16}$$

The above equation can be written in short as $\mathbf{W} = \mathbf{M}\bar{\mathbf{B}}$, which is similar as the rigid factorization (2.2). The only difference lies in the form and dimension of the motion and shape matrices. From the right side of (2.16), it is easy to find that the rank of the nonrigid tracking matrix is at most $3k$ (usually $2m$ and $n$ are both larger than $3k$). The goal of nonrigid factorization is to recover the motion and the deformable structure corresponding to each frame.

### 2.3.2 Nonrigid factorization under affine assumption

Following the same idea of rigid factorization, we perform SVD decomposition on the nonrigid tracking matrix and impose the rank-$3k$ constraint, $\mathbf{W}$ can be factorized into a $2m \times 3k$ matrix $\tilde{\mathbf{M}}$ and a $3k \times n$ matrix $\tilde{\mathbf{B}}$. However, the decomposition is not unique as any nonsingular linear transformation matrix $\mathbf{H} \in \mathbb{R}^{3k \times 3k}$ can be inserted into the factorization which leads to an alternative result $\mathbf{W} = (\tilde{\mathbf{M}}\mathbf{H})(\mathbf{H}^{-1}\tilde{\bar{\mathbf{B}}})$. If we have a transformation matrix $\mathbf{H}$ that can resolve the affine ambiguity and upgrade the solution to the Euclidean space, the shape bases are then easily recovered from $\mathbf{B} = \mathbf{H}^{-1}\tilde{\mathbf{B}}$, while the rotation matrix $\mathbf{R}_{Ai}$ and

the weighting coefficient $\omega_{ij}$ can be decomposed from $\mathbf{M} = \tilde{\mathbf{M}}\mathbf{H}$ by Procrustes analysis [9][11][76].

Similar to the rigid situation, the upgrading matrix can be recovered by imposing metric constraint to the motion matrix. In the following section, we will briefly review some typical methods to deal with this problem.

### Metric constraints

To recover the upgrading matrix, many researchers apply metric constraints to the rotation matrix. Bregler *et al.* [11] first introduced the nonrigid factorization framework and proposed a sub-block factorization algorithm to recover the rotation matrix $\mathbf{R}_{Ai}$ and deformation weights $\omega_{il}$ by decomposing every two-row sub-blocks of the motion matrix $\tilde{\mathbf{M}}$.

In (2.16), each two-row sub-block is given by

$$\mathbf{M}_i = \left[\omega_{i1}\mathbf{R}_{Ai}, \cdots, \omega_{ik}\mathbf{R}_{Ai}\right] \tag{2.17}$$

which can be rearranged as a $k \times 6$ matrix $\mathbf{M}'_i$ as follows.

$$\mathbf{M}'_i = \begin{bmatrix} \omega_{i1}\mathbf{r}_{1i}^T & | & \omega_{i1}\mathbf{r}_{2i}^T \\ \vdots & & \vdots \\ \omega_{ik}\mathbf{r}_{1i}^T & | & \omega_{ik}\mathbf{r}_{2i}^T \end{bmatrix}_{k\times 6} = \begin{bmatrix} \omega_{i1} \\ \vdots \\ \omega_{ik} \end{bmatrix} \left[\mathbf{r}_{1i}^T | \mathbf{r}_{2i}^T\right] \tag{2.18}$$

where $\mathbf{r}_{1i}^T$ and $\mathbf{r}_{2i}^T$ are the first and second rows of the rotation matrix, respectively. Clearly, the deformation weight $\omega_{il}$ and the rotation matrix can be easily derived via the SVD factorization of $\mathbf{M}'_i$ with rank-1 constraint. However, such recovered rotation matrix

$$\tilde{\mathbf{R}}_{Ai} = \begin{bmatrix} \mathbf{r}_{1i}^T \\ \mathbf{r}_{2i}^T \end{bmatrix}$$

is usually not an orthonormal matrix, thus, an orthonormality process is required to find the rotation matrix $\mathbf{R}_{Ai}$ [74].

Concerning the recovery of the upgrading matrix, Xiao *et al.* [95] presented a block computation method via metric constraints. Suppose the $l$-th column triples of $\mathbf{H}$ is $\mathbf{H}_l, l = 1, \cdots, k$, which is independent of each other since $\mathbf{H}$ is nonsingular. Then, we have

$$\tilde{\mathbf{M}}\mathbf{H}_l = \begin{bmatrix} \omega_{1l}\mathbf{R}_{A1} \\ \vdots \\ \omega_{ml}\mathbf{R}_{Am} \end{bmatrix} \tag{2.19}$$

and

$$\tilde{\mathbf{M}}\mathbf{H}_l\mathbf{H}_l^T\tilde{\mathbf{M}}^T = \begin{bmatrix} \omega_{11}^2\mathbf{R}_{A1}\mathbf{R}_{A1}^T & * & \cdots & * \\ * & \omega_{2l}^2\mathbf{R}_{A2}\mathbf{R}_{A2}^T & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & \omega_{ml}^2\mathbf{R}_{Am}\mathbf{R}_{Am}^T \end{bmatrix} \tag{2.20}$$

where $'*'$ stands for nonzero entries. Let $\mathbf{Q}_l = \mathbf{H}_l\mathbf{H}_l^T$, which is a $3k \times 3k$ symmetric matrix with $\frac{1}{2}k(9k+1)$ unknowns. Since $\mathbf{R}_{Ai}$ is an orthonormal rotation matrix, from each diagonal block in (2.20), we have

$$\tilde{\mathbf{M}}_{2i-1:2i}\mathbf{Q}_l\tilde{\mathbf{M}}_{2i-1:2i}^T = \omega_{il}^2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.21}$$

where $\tilde{\mathbf{M}}_{2i-1:2i}$ stands for the $i$-th two-row of $\tilde{\mathbf{M}}$. Thus, we have the following linear constraints on $\mathbf{Q}_l$.

$$\begin{cases} \tilde{\mathbf{M}}_{2i-1}\mathbf{Q}_l\tilde{\mathbf{M}}_{2i}^T = 0 \\ \tilde{\mathbf{M}}_{2i-1}\mathbf{Q}_l\tilde{\mathbf{M}}_{2i-1}^T - \tilde{\mathbf{M}}_{2i}\mathbf{Q}_l\tilde{\mathbf{M}}_{2i}^T = 0 \end{cases}, \forall\, i = 1, \cdots, m. \tag{2.22}$$

Therefore, $\mathbf{Q}_l$ may be computed linearly via least squares from (2.22) given sufficient frames, then, $\mathbf{H}_l$ is recovered from $\mathbf{Q}_l$ via Cholesky decomposition.

In sub-block factorization, it is assumed that all configurations concerning the camera motion and deformation weights are contained in the initially factorized matrix $\tilde{\mathbf{M}}$. Nevertheless, the initial decomposition may yield random solution for $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{B}}$, and result in a bad estimation of the rotation matrix [9][19]. To overcome the limitation of sub-block factorization, Torresani *et al.* [76] proposed a tri-linear approach to solving $\mathbf{B}_l, \omega_{il}$, and $\mathbf{R}_{Ai}$ alternatively by minimizing the following cost function.

$$f(\mathbf{B}_l, \omega_{il}, \mathbf{R}_{Ai}) = \left\| \mathbf{W}_i - \mathbf{R}_{Ai}\Big(\sum_{l=1}^{k}\omega_{il}\mathbf{B}_l\Big) \right\|_F^2, \forall\, i = 1, \cdots, m. \tag{2.23}$$

The iterative algorithm that solves the cost function is initialized by the rigid assumption solution, and a rigid factorization [74] on the nonrigid tracking matrix is performed to obtain an average shape (mean shape) matrix $\tilde{\mathbf{S}}_{rig}$ and a rigid rotation matrix $\tilde{\mathbf{R}}_{Ai}, i = 1, \cdots, m$, for each frame; while the deformation weights are initialized randomly. The algorithm is performed iteratively through the following three steps:

1. Estimate the shape bases $\bar{\mathbf{B}}$ from $\tilde{\mathbf{R}}_{Ai}$ and $\omega_{il}$;
2. Update the deformation weight $\omega_{il}$ from $\bar{\mathbf{B}}$ and $\tilde{\mathbf{R}}_{Ai}$;

3. Update the rotation matrix $\tilde{\mathbf{R}}_{Ai}$ from $\bar{\mathbf{B}}$ and $\omega_{il}$.

The procedure is simpler than the general nonlinear method and usually converges to a proper solution. However, the algorithm does not preserve the replicated block structure of the motion matrix (2.16) during iterations [19].

Similar to the tri-linear technique, Wang *et al.* [82] proposed a rotation constrained power factorization technique by combining the orthonormality of the rotation matrix into a power factorization algorithm. In Addition, Brand [9] proposed a flexible factorization technique to compute the upgrading matrix and recover the motion parameters and deformation weights using an alternative orthonormal decomposition algorithm.

#### Basis constraints

One main problem of using a metric constraint to recover the upgrading matrix lies in its ambiguity. Given the same tracking data, different motion and deformable shapes may be found, since any nonsingular linear transformation matrix can be inserted into the factorization process and, as a result, lead to different sets of eligible shape bases.

On the other hand, when we use the constraints (2.22) to recover the matrix $\mathbf{Q}_l$, it appears that if we have enough features and frames, the upgrading matrix can be solved linearly by exploring all the constraints in (2.22). Unfortunately, using only the rotation constraints may be insufficient when the object deforms at varying speed, since most of the constraints are redundant. Xiao *et al.* [96] proposed a basis constraint to solve this ambiguity.

The main idea is based on the assumption that there exists $k$ frames which include independent shapes that can be treated as a set of bases. Suppose the first $k$ frames are independent of each other. Then, their corresponding weighting coefficients can be set as

$$\omega_{il} = \begin{cases} 1 & \text{if } i,l = 1,...,k \text{ and } i = l \\ 0 & \text{if } i,l = 1,...,k \text{ and } i \neq l \end{cases} . \tag{2.24}$$

Let us define $\Omega = \{(i,j) | i = 1,\cdots,k, j = 1,\cdots,m, i \neq l\}$, then, from (2.20) we

can obtain the following basis constraint.

$$\begin{cases} \tilde{\mathbf{M}}_{2i-1}\mathbf{Q}_l\tilde{\mathbf{M}}_{2j-1}^T &= \begin{cases} 1, i=j=l \\ 0, (i,j)\in\Omega \end{cases} \\ \tilde{\mathbf{M}}_{2i}\mathbf{Q}_l\tilde{\mathbf{M}}_{2j}^T &= \begin{cases} 1, i=j=l \\ 0, (i,j)\in\Omega \end{cases} \\ \tilde{\mathbf{M}}_{2i-1}\mathbf{Q}_l\tilde{\mathbf{M}}_{2j}^T &= \begin{cases} 0, i=j=l \\ 0, (i,j)\in\Omega \end{cases} \\ \tilde{\mathbf{M}}_{2i}\mathbf{Q}_l\tilde{\mathbf{M}}_{2j-1}^T &= \begin{cases} 0, i=j=l \\ 0, (i,j)\in\Omega \end{cases} \end{cases} . \tag{2.25}$$

Altogether we have $4m(k-1)$ linear basis constraints. Using both the metric constraints (2.22) and the basis constraints (2.25), Xiao *et al.* [96] derived a linear close form solution to the nonrigid factorization by dividing the problem into $k$ linear systems. However, the method deals with each column-triples $\mathbf{M}_i$ separately. Thus, the repetitive block structure of the entire motion matrix is not observed during computation. In addition, the solution is dependant on the selection of shape bases, which are treated as prior information of the deformation, but such a selection may be difficult in some situations. Following this idea, Brand [10] proposed a modified approach based on the deviation of the solution from metric constraints. The approach explores a weak constraint to the independent shape bases.

### 2.3.3 Nonrigid factorization under perspective projection

In this section, we will discuss the perspective reconstruction method proposed in [96]. From perspective projection (1.1), we can obtain the following equation system by stacking the projection of each frame.

$$\dot{\mathbf{W}} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1\mathbf{S}_1 \\ \vdots \\ \mathbf{P}_m\mathbf{S}_m \end{bmatrix} \tag{2.26}$$

where $\mathbf{S}_i$ is a $4\times n$ matrix which denotes the 3D structure corresponding to the $i$-th frame in homogeneous form. For rigid objects, the shape does not change with time, thus, $\mathbf{S}_1 = \cdots = \mathbf{S}_m$. Following Bregler's deformation model (2.14), we have

$$\mathbf{P}_i\mathbf{S}_i = \sum_{l=1}^{k}\left(\omega_{il}\mathbf{P}_i^{(1:3)}\mathbf{B}_l\right) + \mathbf{P}_i^{(4)}\mathbf{1}^T \tag{2.27}$$

where $\mathbf{P}_i^{(1:3)}$ and $\mathbf{P}_i^{(4)}$ denote the first three columns and the last column of $\mathbf{P}_i$, respectively; and $\mathbf{1}^T$ is a $n$-vector with unit entries. From equations (2.26) and

(2.27), we obtain the following expression for perspective nonrigid factorization.

$$\dot{\mathbf{W}} = \underbrace{\begin{bmatrix} \omega_{11}\mathbf{P}_1^{(1:3)} & \cdots & \omega_{1k}\mathbf{P}_1^{(1:3)} & \mathbf{P}_1^{(4)} \\ \vdots & \ddots & \vdots & \vdots \\ \omega_{m1}\mathbf{P}_m^{(1:3)} & \cdots & \omega_{mk}\mathbf{P}_m^{(1:3)} & \mathbf{P}_m^{(4)} \end{bmatrix}}_{\mathbf{M}_{3m\times(3k+1)}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \\ \mathbf{1}^T \end{bmatrix}}_{\mathbf{B}_{(3k+1)\times n}} \tag{2.28}$$

where $\mathbf{M} \in \mathbb{R}^{3m\times(3k+1)}$ and $\mathbf{B} \in \mathbb{R}^{(3k+1)\times n}$ are the motion matrix and shape bases, respectively. All nonrigid structures of one object share the same set of shape bases, and both $\mathbf{M}$ and $\mathbf{B}$ are of full rank. Therefore, the rank of the scale-weighted tracking matrix is no more than $\min((3k+1), 3m, n)$. In practice, the point and frame numbers are usually larger than the shape bases number, so the rank of $\dot{\mathbf{W}}$ is at most $3k+1$. This is consistent with that of rigid factorization, where $k = 1$ and the rank is no more than 4.

Suppose the projective depth scales in $\dot{\mathbf{W}}$ are available, then, a projective solution of $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{B}}$ can be obtained through SVD decomposition of the weighted tracking matrix with rank-$(3k+1)$ constraint. Obviously, the solution is defined up to a nonsingular transformation matrix $\mathbf{H} \in \mathbb{R}^{(3k+1)\times(3k+1)}$. Similar to the rigid case, the upgrading matrix can be recovered using both the metric constraints and the basis constraints. Then, the Euclidean motion parameters and deformation structures are recovered from the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{B}}$. Please refer to Xiao and Kanade [96] for computation details.

As for the recovery of the projective depths, a similar iteration method as in [31] and [47], based on the rank constraint on $\dot{\mathbf{W}}$, is adopted. The algorithm starts with weak perspective assumption by setting $\lambda_{ij} = 1$, then, the depth scales are optimized iteratively by minimizing the following cost function

$$J(\lambda_{ij}) = \min \|\dot{\mathbf{W}} - \tilde{\mathbf{M}}\tilde{\mathbf{B}}\|_F^2. \tag{2.29}$$

The minimizing process is achieved iteratively by first factorizing $\dot{\mathbf{W}}$ into $\tilde{\mathbf{M}}\tilde{\mathbf{B}}$ with the given depth scales and then updating the depth scales through back projection. In deformation case, the rank of the tracking matrix is $3k+1$ and the dimension of $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{B}}$ follows the Equation (2.28). In order to avoid trivial solutions of $\lambda_{ij} = 0$, the following constraints are enforced alternatively during the computation to ensure that the depth scales of all points in any frame or a single point in all images have unit norms. As a result, the minimization of (2.29) is converted to a simple eigenvalue problem [96].

### 2.3.4 Nonrigid factorization in trajectory space

Bregler's assumption [11] represents the nonrigid structure in shape space as a linear combination of shape bases. Most recently, Akhter *et al.* [4] [5] proposed a dual representation of nonrigid objects in trajectory space, where the evolving 3D deformable structure is described in the trajectory space by a linear combination of some basis trajectories.

Suppose the structure of a nonrigid object is represented as follows by an $m \times 3n$ matrix.

$$\mathbf{S}^* = \begin{bmatrix} X_{11} & \cdots & X_{1n} & Y_{11} & \cdots & Y_{1n} & Z_{11} & \cdots & Z_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ X_{m1} & \cdots & X_{mn} & Y_{m1} & \cdots & Y_{mn} & Z_{m1} & \cdots & Z_{mn} \end{bmatrix}_{m \times 3n} . \tag{2.30}$$

The row space of $\mathbf{S}^*$ spans the shape space, while the column space of $\mathbf{S}^*$ corresponds to the trajectories of the points, all of which form the trajectory space. Since both the row and column spaces have the same dimension, the trajectory of the points can also be denoted by $k$ trajectory bases. Let us write the trajectory as

$$\mathbf{T}(j) = [\mathbf{T}_x^T(j), \mathbf{T}_y^T(j), \mathbf{T}_z^T(j)]^T \tag{2.31}$$

where

$$\mathbf{T}_x^T(j) = [X_{ij}, \cdots, X_{mj}],\ \mathbf{T}_y^T(j) = [Y_{ij}, \cdots, Y_{mj}],\ \mathbf{T}_z^T(j) = [Z_{ij}, \cdots, Z_{mj}] \tag{2.32}$$

are the $j$-th trajectory in the three coordinates directions. Each trajectory can be denoted by $k$ trajectory bases as follows.

$$\mathbf{T}_x^T(j) = \sum_{l=1}^{k} \omega_{xl}(j)\theta^l,\ \mathbf{T}_y^T(j) = \sum_{l=1}^{k} \omega_{yl}(j)\theta^l,\ \mathbf{T}_z^T(j) = \sum_{l=1}^{k} \omega_{zl}(j)\theta^l \tag{2.33}$$

where $\omega_{xl}(j)$, $\omega_{yl}(j)$, and $\omega_{zl}(j)$ are the coefficients; and $\theta^l = [\theta_1^l, \cdots, \theta_m^l]^T \in \mathbb{R}^m$ is a trajectory basis vector. Expending (2.33) yields

$$\underbrace{\begin{bmatrix} X_{11} & \cdots & X_{1n} \\ Y_{11} & \cdots & Y_{1n} \\ Z_{11} & \cdots & Z_{1n} \\ X_{21} & \cdots & X_{2n} \\ Y_{21} & \cdots & Y_{2n} \\ Z_{21} & \cdots & Z_{2n} \\ \vdots & \cdots & \vdots \\ X_{m1} & \cdots & X_{mn} \\ Y_{m1} & \cdots & Y_{mn} \\ Z_{m1} & \cdots & Z_{mn} \end{bmatrix}}_{\mathbf{S}_{3m \times n}} = \underbrace{\begin{bmatrix} \theta_1^T & & \\ & \theta_1^T & \\ & & \theta_1^T \\ \vdots & & \\ & \vdots & \\ & & \vdots \\ \theta_m^T & & \\ & \theta_m^T & \\ & & \theta_m^T \end{bmatrix}}_{\mathbf{\Theta}_{3m \times 3k}} \underbrace{\begin{bmatrix} \begin{bmatrix} \omega_{x1}^1 & \omega_{x2}^1 & \cdots & \omega_{xn}^1 \\ \vdots & \vdots & \vdots & \vdots \\ \omega_{x1}^k & \omega_{x2}^k & \cdots & \omega_{xn}^k \end{bmatrix} \\ \begin{bmatrix} \omega_{y1}^1 & \omega_{y2}^1 & \cdots & \omega_{yn}^1 \\ \vdots & \vdots & \vdots & \vdots \\ \omega_{y1}^k & \omega_{y2}^k & \cdots & \omega_{yn}^k \end{bmatrix} \\ \begin{bmatrix} \omega_{z1}^1 & \omega_{z2}^1 & \cdots & \omega_{zn}^1 \\ \vdots & \vdots & \vdots & \vdots \\ \omega_{z1}^k & \omega_{z2}^k & \cdots & \omega_{zn}^k \end{bmatrix} \end{bmatrix}}_{\mathbf{\Omega}_{3k \times n}} \tag{2.34}$$

where $\theta_j^T = [\theta_j^1, \cdots, \theta_j^k]^T$. Thus, the nonrigid trajectory can be written concisely as follows.

$$\mathbf{S}_{3m\times n} = \mathbf{\Theta}_{3m\times 3k}\mathbf{\Omega}_{3k\times n}. \tag{2.35}$$

Under orthographic projection, the mapping from 3D space to 2D image can be written as

$$\mathbf{W} = \mathbf{MS} = (\mathbf{M\Theta})\mathbf{\Omega} = \mathbf{\Lambda\Omega} \tag{2.36}$$

where

$$\mathbf{M} = \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_m \end{bmatrix} \tag{2.37}$$

with $\mathbf{R}_j$ the $j$-th $2\times 3$ orthographic projection matrix; and $\mathbf{\Lambda} = \mathbf{M\Theta}$ is a $3m\times 3k$ matrix. It is clear that the factorization in trajectory space (2.36) is a dual expression of the nonrigid factorization in shape space (2.16).

Like in shape space, the rank of (2.36) is at most $3k$, the factorization can be easily obtained via SVD decomposition, however, the solution is not unique and a similar metric constraint can be used to upgrade the solution from perspective to the Euclidean space by an invertible upgrading matrix $\mathbf{H} \in \mathbb{R}^{3k\times 3k}$.

An advantage of the trajectory based algorithm lies in the fact that the trajectory bases can be predefined independently of the tracking data. The key point here is how to select the trajectory bases. Some available candidates include Discrete Sine/Cosine transformation, discrete Wavelet transformation, Hadamard transform basis, etc. In [5], it was demonstrated that DCT basis was suitable to represent human motions. The method was used to reconstruct the 3D trajectory of a moving point from 2D projections [61]. Gotardo and Martinez further studied the approach [27] and applied a kernel trick in the standard nonrigid SfM [28].

## 2.4 Discussion

In the literature, most algorithms assume affine camera model due to its simplicity, while the extension to full perspective model is computational intensive and no convergence is guaranteed. To bridge the gap between the two model, a quasi-perspective projection model is proposed in Chapter 3. The model is showed to be more accurate than the affine assumption and more computational efficient than the full perspective model. The geometric properties of the quasi-perspective model are investigated in Chapter 4 in the context of one- and two-view geometry. Finally, a structure from motion framework for both rigid and nonrigid objects is established in Chapter 5. Theoretical analyses and

experimental studies demonstrate the advantages of the proposed model over the previous affine and perspective projection models.

Robustness is another important issue of SfM because the tracking data are usually corrupted by outliers, missing features, and significant noise. The main difficulty of the robust algorithm lies in the estimation of feature uncertainties, which has not been effectively solved in the literature. In this thesis, the uncertainties are modeled from a new perspective by virtue of the fact that the uncertainties associated with image features are largely in proportion to the reprojection residuals.

A spatial-and-temporal-weighted factorization algorithm is proposed in Chapter 6, where the image uncertainties are modeled both spatially and temporally to address different kinds of errors. An augmented affine factorization algorithm is proposed in Chapter 7 to circumvent the problem of image registration in affine factorization in the presence of outlying and missing data. Based on the new formulation, a robust factorization scheme is presented in the chapter to handle outliers in rigid factorization. In Chapter 8, the idea is successfully extended to nonrigid scenarios with imperfect tracking data. Unlike other existing robust algorithms, the proposed technique requires no prior information of the error distributions of the tracked features. Extensive experiments validate its advantages and effectiveness.

# Chapter 3

# Quasi-Perspective Projection Model

The chapter focuses on the approximation of a full perspective projection model. We first present a review on the affine camera model, including orthographic projection, weak-perspective projection, and paraperspective projection. Then, under the assumption that the camera is far away from the object with small lateral rotations, we show that the imaging process can be modeled by quasi-perspective projection. Geometrical analysis and experimental study show that the proposed model is more accurate than the affine model.

## 3.1 Introduction

The modeling of imaging formation is an important issue for many computer vision applications, such as structure from motion, object recognition, pose estimation, etc. Geometrically, a camera maps data from 3D space to 2D image space. The general camera model used in computer vision is modeled by perspective projection. This is an ideal and accurate model for a wide range of existing cameras. However, the resulting equations from perspective projection are complicated and nonlinear due to the unknown scaling factor [56]. To simplify computation, researchers have proposed many approximations to the full perspective projection.

The most common approximation includes weak-perspective projection, orthographic projection, and paraperspective projection [6]. These approximations are generalized as affine camera model [51] [68]. Faugeras [24] studied the properties of projective cameras. Hartley and Zisserman [32] presented a comprehensive survey and in-depth analysis of different camera models. Affine camera is a zero-order (for weak-perspective) or a first-order (for paraperspective) approximation of full perspective projection [15]. It is valid only when the depth variation of the object is small compared to the distance from camera to the

object. Kanatani *et al.* [40] analyzed a general form of symmetric affine camera model to mimic perspective projection and provided the minimal requirements for orthographic, weak perspective, and para-perspective simplification. The model contains two free variables that can be determined through self-calibration.

In structure from motion, the affine assumption is widely adopted due to its simplicity. In this chapter, we try to make a trade-off between the simplicity of the affine model and the accuracy of the full perspective projection model. By assuming that the camera is far away from the object with small lateral rotations, which is similar to affine assumption and is easily satisfied in practice, we propose a quasi-perspective projection model and present an error analysis of different projection models [83]. The proposed model is shown to be more accurate than the affine approximation. In Chapters 4 and 5 of this thesis, we will further analyze the geometrical properties of the model [85] and its application to structure and motion factorization. Part of this chapter was published in International Journal of Computer Vision [86].

The remaining part of the chapter is organized as follows. The affine projection model is reviewed in Section 3.2. The proposed quasi-perspective model and error analysis are elaborated in Section 3.3. Some experimental evaluations on synthetic data are given in Section 3.4.

## 3.2 Affine Projection Model

Under perspective projection, a 3D point $\mathbf{X}_j$ is projected onto an image point $\mathbf{x}_{ij}$ in frame $i$ according to equation

$$\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j = \mathbf{K}_i[\mathbf{R}_i, \mathbf{t}_i]\mathbf{X}_j \tag{3.1}$$

where $\lambda_{ij}$ is an unknown depth scale; $\mathbf{P}_i$, $\mathbf{R}_i$, and $\mathbf{t}_i$ are the projection matrix, rotation matrix, and translation vector, respectively; and $\mathbf{K}_i$ is the camera calibration matrix with the following general form.

$$\mathbf{K}_i = \begin{bmatrix} f_i & \varsigma_i & u_{0i} \\ 0 & \kappa_i f_i & v_{0i} \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.2}$$

For most precise industrial CCD cameras, it is safe to assume zero skew $\varsigma_i = 0$, known principal point $u_{0i} = v_{0i} = 0$, and unit aspect ratio $\kappa_i = 1$. Then, the camera is simplified to have only one intrinsic parameter $f_i$.

When the distance of an object from a camera is much greater than the depth variation of the object, we may assume an affine camera model. Under affine assumption, the last row of the projection matrix is of the form $\mathbf{P}_{3i}^T \simeq [0, 0, 0, 1]$,

where '$\simeq$' denotes equality up to scale. Thus, a general affine projection matrix for the $i$-th view can be written as

$$\mathbf{P}_{Ai} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \bar{\mathbf{t}}_i \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{3.3}$$

where $\mathbf{A}_i \in \mathbb{R}^{2\times 3}$ is composed by the upper-left $2 \times 3$ submatrix of $\mathbf{P}_i$, $\bar{\mathbf{t}}_i$ is a translation vector. Under affine assumption, the imaging process (3.1) can be simplified by removing the unknown scale factor $\lambda_{ij}$.

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i \bar{\mathbf{X}}_j + \bar{\mathbf{t}}_i. \tag{3.4}$$

Under the affine projection (3.4), the mapping from space to the image is linear. One attractive attribute of the affine camera model is that the mapping is independent of the translation term if relative coordinates are employed in both space and image coordinate frames.

Suppose $\bar{\mathbf{X}}_r$ is a reference point in space and $\bar{\mathbf{x}}_{ir}$ is its image in the $i$-th frame. Then, we have $\bar{\mathbf{x}}_{ir} = \mathbf{A}_i \bar{\mathbf{X}}_r + \bar{\mathbf{t}}_i$. Let us denote

$$\bar{\mathbf{x}}'_{ij} = \bar{\mathbf{x}}_{ij} - \bar{\mathbf{x}}_{ir},\ \bar{\mathbf{X}}'_j = \bar{\mathbf{X}}_j - \bar{\mathbf{X}}_r$$

as the relative image and space coordinates. We can immediately obtain a simplified affine projection equation in terms of relative coordinates.

$$\bar{\mathbf{x}}'_{ij} = \mathbf{A}_i \bar{\mathbf{X}}'_j. \tag{3.5}$$

The translation term $\bar{\mathbf{t}}_i$ is actually the image of the world origin. It is easy to verify that the centroid of a set of space points is projected to the centroid of their images. In practice, we can simply choose the centroid as the reference point, then the translation term vanishes if all the image points in each frame are registered to the corresponding centroid. The affine matrix $\mathbf{A}_i$ has six independent variables which encapsulate both intrinsic and extrinsic parameters of the affine camera. According to RQ decomposition [32], matrix $\mathbf{A}_i$ can be uniquely decomposed into the following form.

$$\mathbf{A}_i = \mathbf{K}_{Ai}\mathbf{R}_{Ai} = \begin{bmatrix} \alpha_{1i} & \zeta_i \\ & \alpha_{2i} \end{bmatrix} \begin{bmatrix} \mathbf{r}_{1i}^T \\ \mathbf{r}_{2i}^T \end{bmatrix} \tag{3.6}$$

where $\mathbf{K}_{Ai}$ is the intrinsic calibration matrix. In accordance with the camera matrix of perspective projection, $\alpha_{1i}$ and $\alpha_{2i}$ are the scaling factors of the two image axes and $\alpha_{1i}/\alpha_{2i}$ is defined as the aspect ratio, $\zeta_i$ is the skew factor of the affine camera. For most CCD cameras, we usually assume unit aspect ratio

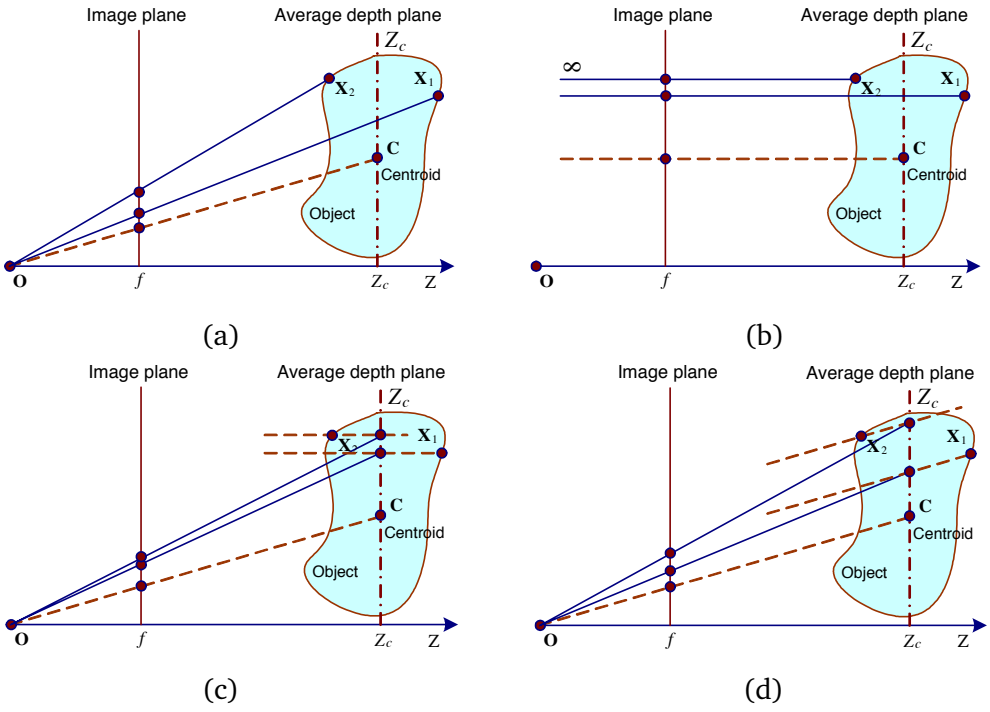



Figure 3.1: The imaging process of different projection models, where **O** is the optical center; $Z = f$ is the image plane; **C** is the centroid of the object; $Z = Z_c$ is the average depth plane; and $\mathbf{X}_1$ and $\mathbf{X}_2$ are two space points on the object. (a) Perspective projection; (b) orthographic projection; (c) weak-perspective projection; (d) para-perspective projection.

$\alpha_{1i} = \alpha_{2i} = \alpha_i$, and zero skew $\zeta_i = 0$. $\mathbf{R}_{Ai}$ is the rotation matrix, and $\mathbf{r}_{1i}^T$ and $\mathbf{r}_{2i}^T$ are the first two rows of the rotation matrix with the constraint

$$\mathbf{r}_{1i}^T \mathbf{r}_{2i} = 0, \ \|\mathbf{r}_{1i}\|^2 = \|\mathbf{r}_{2i}\|^2 = 1 \tag{3.7}$$

while the third row of the rotation matrix can always be recovered from $\mathbf{r}_{3i} = \mathbf{r}_{1i} \times \mathbf{r}_{2i}$. From the above analysis, we can easily see that the affine matrix $\mathbf{A}_i$ has six degrees of freedom. Under affine assumption, the camera projection is usually modeled by three special cases, i.e. orthographic projection, weak perspective projection, and para-perspective projection, as shown in Figure 3.1.

Orthographic projection is the simplest approximation. In this case, it is assumed $\alpha_{1i} = \alpha_{1i} = 1$ and $\zeta_i = 0$. Thus, the projection can be modelled as

$$\mathbf{K}_{ortho} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.8}$$

where the subscript index $i$ is omitted for simplicity. Under weak-perspective projection, the space point is first projected to the average depth plane via orthographic projection, then projected to the image by perspective projection. Thus, the scaling factor is included as $\alpha_{1i} = \alpha_{2i} = \alpha_i$ and $\zeta_i = 0$, which is equivalent to a scaled orthography.

$$\mathbf{K}_{weak} = \alpha \mathbf{K}_{ortho} = \alpha \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{3.9}$$

The weak-perspective projection can model the scaling effect caused by depth changes between images. It is suitable for objects with small depth variations. Para-perspective is a more generalized affine model which is a step closer to perspective projection. As shown in Figure 3.1, the main difference between para-perspective and weak-perspective projection is that the space point is first projected to the average depth plane along the line passing through optical center and the centroid of the object. Thus, it models not only the scaling of weak perspective, but also the apparent result of an object moving towards the edge of the image. Please refer to papers [53] and [62] for more details on para-perspective projection. It can be verified that weak-perspective is a zero-order approximation of full perspective projection, while paraperspective is a first-order approximation.

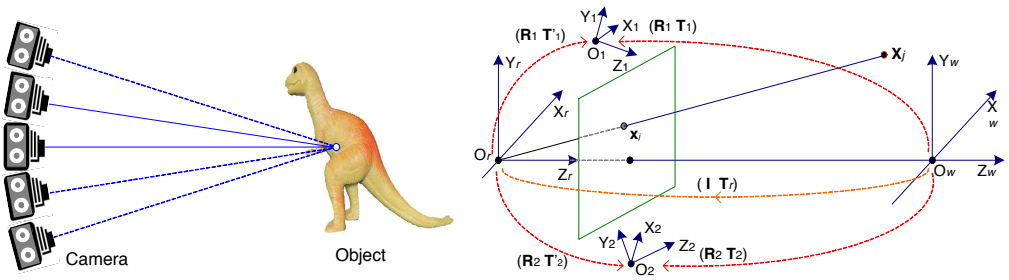
## 3.3 Quasi-Perspective Projection Model

In this section, we will propose a new quasi-perspective projection model to fill the gap between the simplicity of the affine camera model and the accuracy of the full perspective projection model.

### 3.3.1 Quasi-perspective projection

Under perspective projection, the image formation process is shown in Figure 3.2. In 3D reconstruction, in order to ensure that large overlapping parts of the object are observed, the camera usually undergoes really small movements across adjacent views, especially for images of a video sequence.

Suppose $O_w - X_w Y_w Z_w$ is a world coordinate system selected on the object to be reconstructed. $O_i - X_i Y_i Z_i$ is the camera coordinate system with $O_i$ being the optical center of the camera. Without loss of generality, we assume that there is a reference camera system $O_r - X_r Y_r Z_r$. Since the world system can be set freely, we align it with the reference frame as illustrated in Figure 3.2. Therefore, the rotation $\mathbf{R}_i$ of frame $i$ with respect to the reference frame is the same as the rotation of the camera to the world system.




(a) (b)

Figure 3.2: The imaging process of a camera and the coordinates systems. (a) Camera setup with respect to the object; and (b) the relationship of world coordinate system and camera systems at different viewpoint.

**Definition 3.1 (Axial and lateral rotation)**
The orientation of a camera is usually described by roll-pitch-yaw angles. For the $i$-th frame, we define the pitch, yaw, and roll as the rotations $\alpha_i, \beta_i$, and $\gamma_i$ of the camera with respect to the $X_w, Y_w$, and $Z_w$ axes of the world system. As shown in Figure 3.2, the optical axis of the cameras usually point towards the object. For convenience of discussion, we define $\gamma_i$ as the axial rotation angle, and define $\alpha_i$ and $\beta_i$ as lateral rotation angles.

**Proposition 3.1**
Suppose the camera undergoes small lateral rotation with respect to the reference frame, then, the variation of projective depth $\lambda_{ij}$ is proportional to the depth of the space point. The projective depths of a point at different views have same trend of variation.

**Proof** Suppose the rotation matrix and translation vector of the $i$-th frame with respect to the world system are

$$\mathbf{R}_i = \begin{bmatrix} \mathbf{r}_{1i}^T \\ \mathbf{r}_{2i}^T \\ \mathbf{r}_{3i}^T \end{bmatrix}, \ \mathbf{T}_i = \begin{bmatrix} t_{xi} \\ t_{yi} \\ t_{zi} \end{bmatrix}. \tag{3.10}$$

Then, the projection matrix can be written as

$$\begin{aligned} \mathbf{P}_i &= \mathbf{K}_i[\mathbf{R}_i, \mathbf{t}_i] \\ &= \begin{bmatrix} f_i \mathbf{r}_{1i}^T + \varsigma_i \mathbf{r}_{2i}^T + u_{0i} \mathbf{r}_{3i}^T & f_i t_{xi} + \varsigma_i t_{yi} + u_{0i} t_{zi} \\ \kappa_i f_i \mathbf{r}_{2i}^T + v_{0i} \mathbf{r}_{3i}^T & \kappa_i f_i t_{yi} + v_{0i} t_{zi} \\ \mathbf{r}_{3i}^T & t_{zi} \end{bmatrix}. \end{aligned} \tag{3.11}$$

The rotation matrix can be decomposed into rotations around the three axes of the world frame.

$$\begin{aligned}\mathbf{R}_i=&\mathbf{R}(\gamma_i)\mathbf{R}(\beta_i)\mathbf{R}(\alpha_i) \qquad (3.12)\\ =&\begin{bmatrix}\cos\gamma_i,-\sin\gamma_i,0\\ \sin\gamma_i,\ \cos\gamma_i,\ 0\\ 0,\quad 0,\quad 1\end{bmatrix}\begin{bmatrix}\cos\beta_i,\ 0,\sin\beta_i\\ 0,\quad 1,\quad 0\\ -\sin\beta_i,0,\cos\beta_i\end{bmatrix}\begin{bmatrix}1,\quad 0,\quad 0\\ 0,\cos\alpha_i,-\sin\alpha_i\\ 0,\sin\alpha_i,\ \cos\alpha_i\end{bmatrix}\\ =&\begin{bmatrix}\cos\gamma_i\cos\beta_i,\cos\gamma_i\sin\beta_i\sin\alpha_i-\sin\gamma_i\cos\alpha_i,\cos\gamma_i\sin\beta_i\cos\alpha_i+\sin\gamma_i\sin\alpha_i\\ \sin\gamma_i\cos\beta_i,\sin\gamma_i\sin\beta_i\sin\alpha_i+\cos\gamma_i\cos\alpha_i,\sin\gamma_i\sin\beta_i\cos\alpha_i-\cos\gamma_i\sin\alpha_i\\ -\sin\beta_i,\qquad \cos\beta_i\sin\alpha_i,\qquad \cos\beta_i\cos\alpha_i\end{bmatrix}.\end{aligned}$$

Inserting (3.11) and (3.12) into (3.1), we have

$$\lambda_{ij}=[\mathbf{r}_{3i}^T,t_{zi}]\mathbf{X}_j=-(\sin\beta_i)x_j+(\cos\beta_i\sin\alpha_i)y_j+(\cos\beta_i\cos\alpha_i)z_j+t_{zi}. \qquad (3.13)$$

From Figure 3.2, the rotation angles $\alpha_i,\beta_i$, and $\gamma_i$ of the camera to the world system are the same as those to the reference frame. Under small lateral rotations, i.e., small angles of $\alpha_i$ and $\beta_i$, we have

$$\sin\beta_i\ll\cos\beta_i\cos\alpha_i,\ \cos\beta_i\sin\alpha_i\ll\cos\beta_i\cos\alpha_i. \qquad (3.14)$$

Thus, Equation (3.13) can be approximated by

$$\lambda_{ij}\approx(\cos\beta_i\cos\alpha_i)z_j+t_{zi}. \qquad (3.15)$$

All features $\{x_{ij}|j=1,...,n\}$ in the $i$-th frame correspond to the same rotation angles $\alpha_i,\beta_i,\gamma_i$, and translation $t_{zi}$. It is evident from (3.15) that the projective depths of a point in all frames have same trend of variation, which are proportional to the value of $z_j$. Actually, the projective depths have no relation with the axial rotation $\gamma_i$. ■

**Proposition 3.2**
Under small lateral rotations and a further assumption that the distance from the camera to an object is significantly greater than the object depth, i.e., $t_{zi}>>z_j$, the ratio of $\{\lambda_{ij}|i=1,...,m\}$ corresponding to any two different frames can be approximated by a constant.

**Proof** Let us take the reference frame as an example, the ratio of the projective depths of any frame $i$ to those of the reference frame can be written as

$$\mu_i=\frac{\lambda_{rj}}{\lambda_{ij}}\approx\frac{(\cos\beta_r\cos\alpha_r)z_j+t_{zr}}{(\cos\beta_i\cos\alpha_i)z_j+t_{zi}} \qquad (3.16)$$

$$= \frac{\cos\beta_r \cos\alpha_r (z_j / t_{zi}) + t_{zr}/t_{zi}}{\cos\beta_i \cos\alpha_i (z_j / t_{zi}) + 1}$$

where $\cos\beta_i \cos\alpha_i \leq 1$. Under the assumption that $t_{zi} >> z_j$, the ratio can be approximated by

$$\mu_i = \frac{\lambda_{rj}}{\lambda_{ij}} \approx \frac{t_{zr}}{t_{zi}}. \tag{3.17}$$

All features in a frame have the same translation term. Therefore, we can see from (3.17) that the projective depth ratios of two frames for all features have the same approximation $\mu_i$. ■

According to Proposition 3.2, the projective depth scale is decoupled into the product of two parts as $\lambda_{ij} = \frac{1}{\mu_i}\lambda_{rj}$. Thus, the perspective projection Equation (3.1) can be approximated by

$$\frac{1}{\mu_i}\lambda_{rj}\mathbf{x}_{ij} = \mathbf{P}_i \mathbf{X}_j. \tag{3.18}$$

Let us denote $\lambda_{rj}$ as $\frac{1}{\ell_j}$, and reformulate (3.18) as

$$\mathbf{x}_{ij} = \mathbf{P}_{qi}\mathbf{X}_{qj} \tag{3.19}$$

where

$$\mathbf{P}_{qi} = \mu_i \mathbf{P}_i, \;\; \mathbf{X}_{qj} = \ell_j \mathbf{X}_j. \tag{3.20}$$

We call Equation (3.19) quasi-perspective projection model. Compared with general perspective projection, the quasi-perspective model assumes that projective depths between different frames are defined up to a constant $\mu_i$. Thus, the projective depths can be decoupled and implicitly embedded in the scalars of the homogeneous structure $\mathbf{X}_{qj}$ and the projection matrix $\mathbf{P}_{qi}$. As a result, the difficult problem on estimating the unknown depths is avoided. The model is more general than affine projection model (3.4), where all projective depths are simply assumed to be equal to $\lambda_{ij} = 1$.

### 3.3.2 Error analysis of different models

In this section, we will present a heuristic analysis on imaging errors of quasi-perspective and affine camera models with respect to the full perspective projection. For simplicity, the subscript in '$i$' of the frame number is omitted throughout the analysis.

Suppose the intrinsic parameters of the cameras are known, and all images are normalized by the cameras as $\mathbf{K}_i^{-1}\mathbf{x}_{ij} \rightarrow \mathbf{x}_{ij}$. Then, the projection matrices under different projection models can be written as

$$\mathbf{P} = \begin{bmatrix} \mathbf{r}_1^T & t_x \\ \mathbf{r}_2^T & t_y \\ \mathbf{r}_3^T & t_z \end{bmatrix}, \quad \mathbf{r}_3^T = [-\sin\beta, \cos\beta\sin\alpha, \cos\beta\cos\alpha] \tag{3.21}$$

$$\mathbf{P}_q = \begin{bmatrix} \mathbf{r}_1^T & t_x \\ \mathbf{r}_2^T & t_y \\ \mathbf{r}_{3q}^T & t_z \end{bmatrix}, \quad \mathbf{r}_{3q}^T = [0, 0, \cos\beta\cos\alpha] \tag{3.22}$$

$$\mathbf{P}_a = \begin{bmatrix} \mathbf{r}_1^T & t_x \\ \mathbf{r}_2^T & t_y \\ \mathbf{0}^T & t_z \end{bmatrix}, \quad \mathbf{0}^T = [0, 0, 0] \tag{3.23}$$

where $\mathbf{P}$ is the projection matrix of full perspective projection, $\mathbf{P}_q$ is that of quasi-perspective assumption, and $\mathbf{P}_a$ is that of affine projection. Obviously, the main difference between the above projection matrices lies only in the last row. For a space point $\bar{\mathbf{X}} = [x, y, z]^T$, its projected images under different camera models are given by

$$\mathbf{m} = \mathbf{P}\begin{bmatrix} \bar{\mathbf{X}} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ \mathbf{r}_3^T\bar{\mathbf{X}} + t_z \end{bmatrix} \tag{3.24}$$

$$\mathbf{m}_q = \mathbf{P}_q\begin{bmatrix} \bar{\mathbf{X}} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ \mathbf{r}_{3q}^T\bar{\mathbf{X}} + t_z \end{bmatrix} \tag{3.25}$$

$$\mathbf{m}_a = \mathbf{P}_a\begin{bmatrix} \bar{\mathbf{X}} \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ t_z \end{bmatrix} \tag{3.26}$$

where

$$u = \mathbf{r}_1^T\bar{\mathbf{X}} + t_x, v = \mathbf{r}_2^T\bar{\mathbf{X}} + t_y \tag{3.27}$$

$$\mathbf{r}_3^T\bar{\mathbf{X}} = -(\sin\beta)x + (\cos\beta\sin\alpha)y + (\cos\beta\cos\alpha)z \tag{3.28}$$

$$\mathbf{r}_{3q}^T\bar{\mathbf{X}} = (\cos\beta\cos\alpha)z \tag{3.29}$$

and the nonhomogeneous image points can be denoted as

$$\bar{\mathbf{m}} = \frac{1}{\mathbf{r}_3^T\bar{\mathbf{X}} + t_z}\begin{bmatrix} u \\ v \end{bmatrix} \tag{3.30}$$

$$\bar{\mathbf{m}}_q = \frac{1}{\mathbf{r}_{3q}^T \bar{\mathbf{X}} + t_z} \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.31}$$

$$\bar{\mathbf{m}}_a = \frac{1}{t_z} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{3.32}$$

The point $\bar{\mathbf{m}}$ is an ideal image by perspective projection. Let us define $\mathbf{e}_q = \|\bar{\mathbf{m}}_q - \bar{\mathbf{m}}\|$ as the error of quasi-perspective, and $\mathbf{e}_a = \|\bar{\mathbf{m}}_a - \bar{\mathbf{m}}\|$ as the error of affine, where $\|\cdot\|$ stands for the Euclidean norm of a vector. Then, we have

$$\begin{aligned}
\mathbf{e}_q &= \|\bar{\mathbf{m}}_q - \bar{\mathbf{m}}\| \\
&= \left\| \frac{\mathbf{r}_3^T \bar{\mathbf{X}} + t_z}{\mathbf{r}_{3q}^T \bar{\mathbf{X}} + t_z} \bar{\mathbf{m}} - \bar{\mathbf{m}} \right\| = \det\left( \frac{(\mathbf{r}_3^T - \mathbf{r}_{3q}^T)\bar{\mathbf{X}}}{\mathbf{r}_{3q}^T \bar{\mathbf{X}} + t_z} \right) \|\bar{\mathbf{m}}\| \\
&= \det\left( \frac{-(\sin\beta)x + (\cos\beta \sin\alpha)y}{(\cos\beta\cos\alpha)z + t_z} \right) \|\bar{\mathbf{m}}\| \qquad (3.33) \\
\mathbf{e}_a &= \|\bar{\mathbf{m}}_a - \bar{\mathbf{m}}\| \\
&= \left\| \frac{\mathbf{r}_3^T \bar{\mathbf{X}} + t_z}{t_z} \bar{\mathbf{m}} - \bar{\mathbf{m}} \right\| = \det\left( \frac{\mathbf{r}_3^T \bar{\mathbf{X}}}{t_z} \right) \|\bar{\mathbf{m}}\| \\
&= \det\left( \frac{-(\sin\beta)x + (\cos\beta \sin\alpha)y + (\cos\beta\cos\alpha)z}{t_z} \right) \|\bar{\mathbf{m}}\|. \qquad (3.34)
\end{aligned}$$

Based on the above equations, it is rational to state the following results for different projection models.

1. The axial rotation angle $\gamma$ around $Z$-axis has no influence on the images of $\bar{\mathbf{m}}$, $\bar{\mathbf{m}}_q$, and $\bar{\mathbf{m}}_a$.

2. When the distance of a camera to an object is much larger than the object depth, both $\bar{\mathbf{m}}_q$ and $\bar{\mathbf{m}}_a$ are close to $\bar{\mathbf{m}}$.

3. When the camera system is aligned with the world system, i.e., $\alpha = \beta = 0$, we have $\mathbf{r}_{3q}^T = \mathbf{r}_3^T = [0, 0, 1]$ and $\mathbf{e}_q = 0$. Thus, $\bar{\mathbf{m}}_q = \bar{\mathbf{m}}$, and the quasi-perspective assumption is equivalent to perspective projection.

4. When the rotation angles $\alpha$ and $\beta$ are small, we have $\mathbf{e}_q < \mathbf{e}_a$, i.e., the quasi-perspective assumption is more accurate than the affine assumption.

5. When the space point lies on the plane through the world origin and perpendicular to the principal axis, i.e., the direction of $\mathbf{r}_3^T$, we have $\alpha = \beta = 0$ and $z = 0$. It is easy to verify that $\bar{\mathbf{m}} = \bar{\mathbf{m}}_q = \bar{\mathbf{m}}_a$.

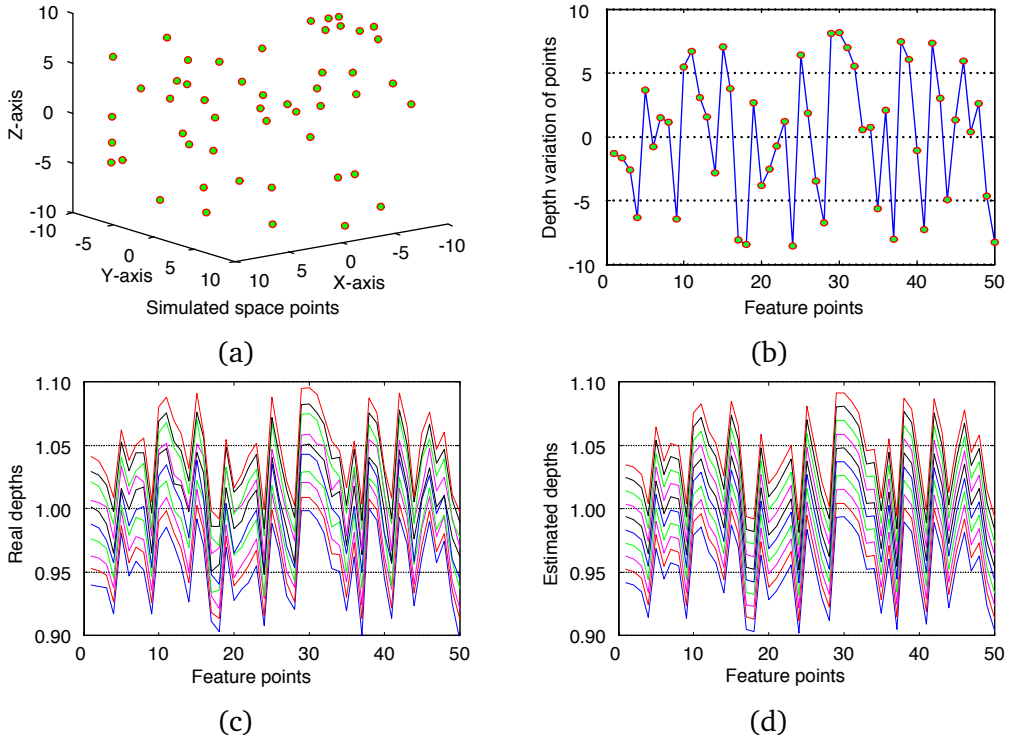
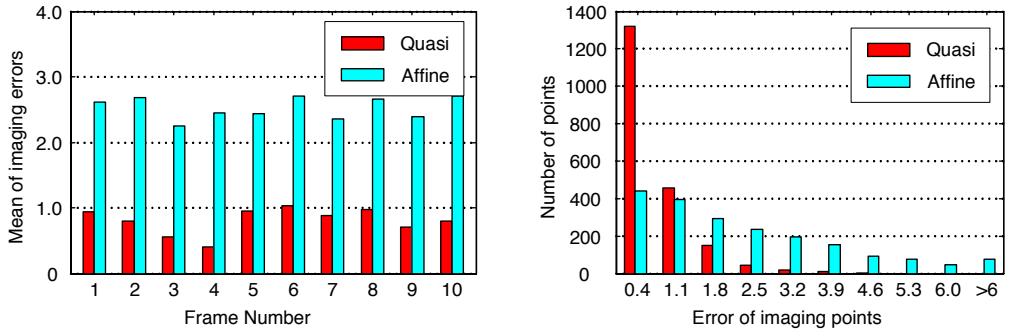



Figure 3.3: Evaluation on projective depth approximation of the first 50 points. (a) Coordinates and distribution of the synthetic space points; (b) the depth variation of the space points; (c) the real projective depths of the imaged points after normalization; and (d) the approximated projective depths under quasi-perspective assumption.

## 3.4 Experimental Evaluations

During simulation, we randomly generated 200 points within a cube of $20 \times 20 \times 20$ in space as shown in Figure 3.3(a), only the first 50 points are displayed for simplicity. The depth variation in Z-direction of the space points is shown in Figure 3.3(b). We simulated 10 images from these points by perspective projection. The image size is set at $800 \times 800$. The camera parameters are set as follows: focal lengths are set randomly between 900 and 1100, the principal point is set to the image center, and the skew is zero. The rotation angles are set randomly between $\pm 5°$. The X and Y positions of the cameras are set randomly between $\pm 15$, while the Z position is spaced evenly from 200 to 220.

The true projective depths $\lambda_{ij}$ associated with these points across the 10 different views are shown in Figure 3.3(c), where the values are given after




(a) (b)

Figure 3.4: Evaluation of the imaging errors by different camera models. (a) The mean error of the generated images by the quasi-perspective and affine projection models; and (b) the histogram distribution of the errors by different projection models.

normalization so that they have unit mean.

### 3.4.1 Imaging errors

Using the simulated data, we estimated the values of $\lambda_{1j}$ and $\mu_i$ from (3.15) and (3.16), and constructed the estimated projective depths from $\hat{\lambda}_{ij} = \frac{\lambda_{1j}}{\mu_i}$. The normalized result is shown in Figure 3.3(d). We can see from the experiment that the recovered projective depths are very close to the ground truth. They are generally in proportion to the depth variation of the space points in the Z-direction. If we adopt the affine camera model, it is equivalent to setting all projective depths to $\lambda_{ij} = 1$. The error is obviously much bigger than that recovered from the quasi-perspective assumption.

According to the projection equations (3.30) to (3.34), different images will be obtained if we adopt different camera models. We generated three sets of images using the simulated space points via full perspective projection model, affine camera model, and the quasi-perspective projection model. We compared the errors given by the quasi-perspective projection model (3.33) and affine assumption (3.34). The mean error of different models in each frame is shown in Figure 3.4(a), the histogram distribution of the errors for all 200 points across 10 frames is shown in Figure 3.4(b). The results indicate that the error of quasi-perspective assumption is much smaller than that under affine assumption.

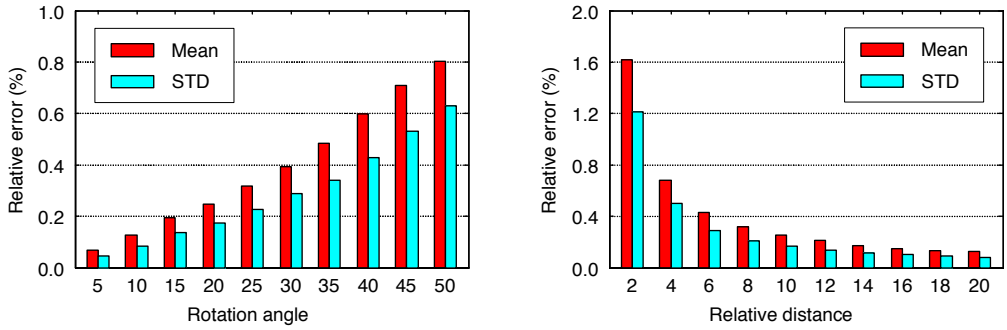



Figure 3.5: Evaluation on quasi-perspective projection under different imaging conditions. (a) The relative error of the estimated depths under different rotation angles; and (b) the relative error with respect to different relative distances.

### 3.4.2 Influence of imaging conditions

The proposed quasi-perspective model is based on the assumption of small camera movement. We investigated the influence of different imaging conditions to the model. First, we fix the camera position as given in the first test and vary the amplitude of rotation angles from $\pm 5^\circ$ to $\pm 50^\circ$ in steps of $5^\circ$. At each step, we calculate the following defined relative errors of the recovered projective depths.

$$e_{ij} = \frac{|\lambda_{ij} - \hat{\lambda}_{ij}|}{\lambda_{ij}} \times 100\,(\%) \tag{3.35}$$

where $\hat{\lambda}_{ij}$ is the estimated projective depth. At each step,100 independent tests were carried out so as to obtain a statistically meaningful result. The mean and standard deviation of $e_{ij}$ are shown in Figure 3.5(a).

Then, we fix the rotation angles at $\pm 5^\circ$ and vary the relative distance of the camera to the object (i.e. the ratio between the distance of the camera to the object center and the object depth) from 2 to 20 in steps of 2. The mean and standard deviation of $e_{ij}$ at each step for 100 tests are shown in Figure 3.5(b). It is evident that the quasi-perspective projection is a good approximation ($e_{ij} < 0.5\%$) when the rotation angles are less than $\pm 35^\circ$ and the relative distance is larger than 6. Please note that the result is obtained from noise free data.

## 3.5 Conclusion

In this chapter, we have proposed a quasi-perspective projection model and analyzed the projection errors of different projection models. The proposed

model is a trade-off between the affine assumption and the perspective projection. It is computationally efficient with better accuracy than the affine approximation.

The proposed model is suitable for structure and motion factorization of a short sequence with small camera motions. It should be noted that the small rotation assumption of the proposed model is not a limiting factor and is usually satisfied in many real-world applications. During image acquisition of an object to be reconstructed, we tend to control the camera movement so as to guarantee large overlapping part, which also facilitates the feature tracking process. Some geometrical properties of the model in one-view and two-view [85] will be presented in Chapter 4. The application details to structure and motion factorization [86] will be discussed in Chapter 5.

# Chapter 4

# Properties of Quasi-Perspective Model

This chapter investigates the geometrical properties of the quasi-perspective projection model in one- and two-view geometry. The main results are as follows: (i) Quasi-perspective projection matrix has nine degrees of freedom, and the parallelism along the $X$ and $Y$ directions in world system are preserved in images; (ii) quasi-fundamental matrix can be simplified to a special form with only six degrees of freedom. The fundamental matrix is invariant to any non-singular projective transformation; (iii) plane induced homography under quasi-perspective model can be simplified to a special form defined by six degrees of freedom. The quasi-homography may be recovered from two pairs of corresponding points with known fundamental matrix; and (iv) any two reconstructions in quasi-perspective space are defined up to a non-singular quasi-perspective transformation.

## 4.1 Introduction

Stereo vision and structure from motion algorithms are based on certain assumption of camera models, including the perspective projection model, affine projection model, and the quasi-perspective projection model proposed in Chapter 3. Different models have different geometrical properties.

The fundamental matrix is an important concept in stereo vision as it encapsulates the underlying epipolar geometry between images. Classical linear techniques for fundamental matrix estimation are sensitive to noise. Hartley [35] analyzed the problem and proposed a normalized eight-point algorithm to improve the stability and accuracy of computation. Zhang and Kanade[107] provided a good review on fundamental matrix estimation and uncertainty analysis. Hu *et al.* [38] proposed to use evolutionary agents for epipolar geometry estimation. The RANdom SAmple Consensus (RANSAC) paradigm [26] was originated for robust parameter estimation in presence of outliers that severely

affect least-squares based techniques. Torr *et al.* [59] proposed to adopt RANSAC to estimate fundamental matrix. Cheng and Lai [13] proposed a consensus sampling technique to increase the probability of sampling inliers. Dellaert *et al.* [22] also proposed a robust method to reject outliers and reconstruct 3D scene geometry.

The concept of affine camera and affine fundamental matrix were well established in [51] [68] as a generalization of orthographic, weak perspective, and paraperspective projections. Zhang and Xu [108] presented a general expression of fundamental matrix for both projective and affine cameras. Wolf and Shashua [94] investigated the recovery of affine fundamental matrix and structure of multiple planes moving relatively to each other under pure translation between two cameras.

Mendonca and Cipolla [48] investigated the trifocal tensor for an affine trinocular rig. Guilbert *et al.* [30] presented a batch algorithm to recover the Euclidean camera motion from sparse data for affine cameras. Lehmann *et al.* [43] proposed an integral projection approach to determine affine fundamental matrix directly from two sets of features without any correspondence or explicit constraint on the data. Shimshoni *et al.* [70] presented a geometric interpretation for weak-perspective motion from two and three images. Zhang *et al.* [105] investigated the problem of structure reconstruction by the combination of perspective and weak-perspective images.

The quasi-perspective projection model was originally proposed for factorization based structure recovery from image sequences [86]. In this chapter, we will further investigate some geometrical properties of the model in one- and two-view geometry. Part of this chapter was published in Pattern Recognition Journal [85].

The remainder of this chapter is organized as follows. Properties of the quasi-perspective projection matrix are given in Section 4.2. The two-view geometry of the model is elaborated in Section 4.3. Some properties on quasi-perspective reconstruction are presented in Section 4.4. Extensive experimental evaluations on synthetic and real images are reported in Section 4.5 and 4.6 respectively.

## 4.2 One-View Geometrical Property

In Chapter 3, under the assumption that the camera is far away from the object with small lateral rotations and translation, we proved that the variation of projective depth $\lambda_{ij}$ is mainly proportional to the depth of the corresponding space point, and the projective depths between different frames can be defined up to a constant. Thus, the projective depths may be decoupled and implicitly embedded in the scalars of the homogeneous structure and the projection matrix.

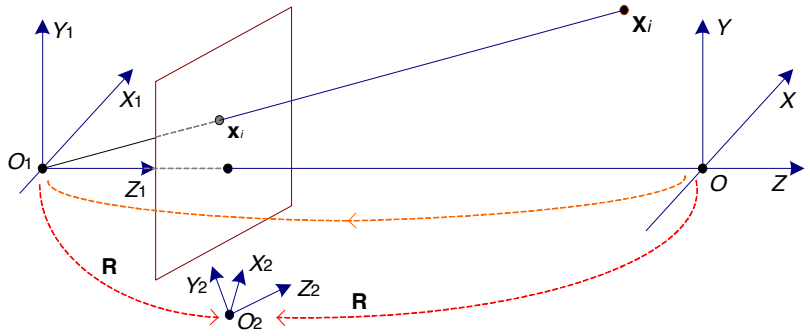



Figure 4.1: The imaging process and relationship of the world coordinate system with respect to the camera system at different positions.

Consequently, the imaging process is simplified to the following quasi-perspective projection model.

$$\mathbf{x}_i = \mathbf{P}_q \mathbf{X}_{qi} \tag{4.1}$$

where $\mathbf{P}_q$ is the quasi-perspective projection matrix and $\mathbf{X}_{qi}$ is scale weighted space point in homogeneous form. In this section, we will present some geometrical properties of the model.

**Proposition 4.1**
The quasi-perspective projection matrix has nine degrees of freedom, and its last row is of the form $\mathbf{P}_3^T \simeq [0,0,*,*]$, where $'*'$ stands for a nonzero entry.

**Proof** As shown in Figure 4.1, let us take the coordinate system $O_1 - X_1Y_1Z_1$ of first camera as a reference frame. Without loss of generality, we assume that the world coordinate system $O-XYZ$ is aligned with the reference system. Then, the camera rotation $\mathbf{R}$ with respect to the reference frame is the same as the rotation with respect to the world system.

Suppose the rotation and translation of the second view are $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3]^T = \mathbf{R}(\gamma)\mathbf{R}(\beta)\mathbf{R}(\alpha)$ and $\mathbf{T} = [t_1, t_2, t_3]^T$, where $\alpha, \beta$, and $\gamma$ are the rotation angles along the three axes $X, Y$, and $Z$, respectively. Then, the last row of the projection matrix can be written as

$$\mathbf{P}_3^T = [\mathbf{r}_3^T, t_3] = [-\sin\beta, \cos\beta\sin\alpha, \cos\beta\cos\alpha, t_3]. \tag{4.2}$$

Under quasi-perspective assumption of small rotations, we have

$$\{\sin\alpha, \sin\beta, \sin\gamma\} \to 0, \ \{\cos\alpha, \cos\beta, \cos\gamma\} \to 1 \tag{4.3}$$

which results to $\sin\beta \ll \cos\beta\cos\alpha \leq 1$, and $\cos\beta\sin\alpha \ll \cos\beta\cos\alpha \leq 1$. Thus, the last row (4.2) can be approximated by

$$\mathbf{P}_3^T \simeq [0, 0, \cos\beta\cos\alpha, t_z] = [0, 0, *, *]. \tag{4.4}$$

As a consequence, the quasi-perspective projection matrix $\mathbf{P}_q$ has only ten nonzero entries and nine degrees of freedom since it is defined up to a global scale. The matrix $\mathbf{P}_q$ can be linearly solved from a minimum of $4\frac{1}{2}$ space to image correspondences. In comparison, at least $5\frac{1}{2}$ correspondences are required for the recovery of a full perspective projection matrix, and four pairs of matches for the affine model. ■

**Proposition 4.2**
Under quasi-perspective projection, parallel lines along $X$ and $Y$ directions of the world system are mapped to parallel lines in the image.

**Proof** In projective geometry, parallel lines in 3D space intersect at a point at infinity, which is also called ideal point in $\mathbb{P}^3$. The intersection of the parallel lines along $X$ and $Y$ directions can be written in canonical form as $\mathbf{V}_x = [1, 0, 0, 0]^T$, $\mathbf{V}_y = [0, 1, 0, 0]^T$. Then, their images are obtained as

$$\mathbf{v}_x = \mathbf{P}_q\mathbf{V}_x = \begin{bmatrix} * \\ * \\ 0 \end{bmatrix}, \ \mathbf{v}_y = \mathbf{P}_q\mathbf{V}_y = \begin{bmatrix} * \\ * \\ 0 \end{bmatrix}. \tag{4.5}$$

Obviously, both $\mathbf{v}_x$ and $\mathbf{v}_y$ locate at infinity, and the parallelism is preserved along the $X$ and $Y$ directions. ■

In a similar way, we can verify that parallel lines on the $O-XY$ coordinate plane or any other parallel planes also project to parallel lines in the image. However, the parallel relationship is not preserved in the $Z$-direction. This is different with respect to affine camera model where the parallelism is invariant.

When the space points are coplanar, we assume that the plane is $Z = 0$ without loss of generality, then, the quasi-perspective projection (4.1) is simplified to

$$\mathbf{x}_i \simeq \mathbf{H}_\pi \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & h_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix} \tag{4.6}$$

where $\mathbf{H}_\pi \in \mathbb{R}^{3\times3}$ is called homography or perspectivity. There are six degrees of freedom in the perspectivity, so $\mathbf{H}_\pi$ can be recovered from three non-collinear space points with known positions. The form of the perspectivity in (4.6) is the same as that under affine model. In contrast, a general homography under perspective model has eight degrees of freedom and at least four points are required for computation.

## 4.3 Two-View Geometrical Property

In this section, More geometrical properties of quasi-perspective projection are introduced in the context of two views.

### 4.3.1 Fundamental matrix

Epipolar geometry is the intrinsic projective geometry between a pair of stereo images, and the intrinsic geometry between two images can be encapsulated as

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0 \tag{4.7}$$

where $\mathbf{F} \in \mathbb{R}^{3\times3}$ is called fundamental matrix; and $\mathbf{x}$ and $\mathbf{x}'$ are a pair of corresponding points between the images. If the camera parameters are calibrated and the images are normalized as $\mathbf{y} = \mathbf{K}^{-1}\mathbf{x}$, $\mathbf{y}' = \mathbf{K}'^{-1}\mathbf{x}'$, we have an analogous relation as $\mathbf{y}'^T\mathbf{E}\mathbf{y} = 0$, where $\mathbf{E}$ is named as essential matrix. The fundamental and essential matrices are related by

$$\mathbf{E}_q = \mathbf{K}'^T\mathbf{F}_q\mathbf{K}. \tag{4.8}$$

Both the fundamental matrix and the essential matrix are rank-2 homogeneous matrices defined up to scales. Thus, they have only seven degrees of freedom.

**Proposition 4.3**
Both the fundamental matrix and the essential matrix under quasi-perspective projection can be simplified to the form of

$$\begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix},$$

which is defined by five degrees of freedom.

**Proof** Given the rotation $\mathbf{R} = \mathbf{R}(\gamma)\mathbf{R}(\beta)\mathbf{R}(\alpha)$ and the translation $\mathbf{t} = [t_1, t_2, t_3]^T$ between two views, the essential matrix can be computed from

$$\mathbf{E}_q = [\mathbf{t}]_\times \mathbf{R} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} \tag{4.9}$$

where

$$\begin{aligned} [\mathbf{t}]_\times &= \begin{bmatrix} 0 & t_3 & -t_2 \\ -t_3 & 0 & t_1 \\ t_2 & -t_1 & 0 \end{bmatrix}, \\ e_{11} &= t_3 \sin\gamma \cos\beta + t_2 \sin\beta, \\ e_{21} &= -t_3 \cos\gamma \cos\beta - t_1 \sin\beta, \\ e_{12} &= t_3(\cos\gamma \cos\alpha + \sin\gamma \sin\beta \sin\alpha) - t_2 \cos\beta \sin\alpha, \\ e_{22} &= t_3(\sin\gamma \cos\alpha - \cos\gamma \sin\beta \sin\alpha) + t_1 \cos\beta \sin\alpha. \end{aligned}$$

Using Equation (4.3), we can see that $\{e_{11}, e_{22}\} \to 0$. Therefore, the essential matrix is simplified to

$$\mathbf{E}_q = \begin{bmatrix} 0 & e_{12} & e_{13} \\ e_{21} & 0 & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix}. \tag{4.10}$$

Suppose the camera parameters are fixed as

$$\mathbf{K} = \mathbf{K}' = \begin{bmatrix} f_1 & 0 & u_0 \\ 0 & f_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

then, the quasi-fundamental matrix can be obtained from

$$\begin{aligned} \mathbf{F}_q = \mathbf{K}'^{-T} \mathbf{E}_q \mathbf{K}^{-1} &= \begin{bmatrix} e_{11}/f_1^2 & e_{12}/(f_1 f_2) & * \\ e_{21}/(f_1 f_2) & e_{22}/f_2^2 & * \\ * & * & * \end{bmatrix} \\ &= \begin{bmatrix} 0 & f_{12} & f_{13} \\ f_{21} & 0 & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} = \begin{bmatrix} 0 & * & * \\ * & 0 & * \\ * & * & * \end{bmatrix}. \end{aligned} \tag{4.12}$$

Both $\mathbf{E}_q$ and $\mathbf{F}_q$ have the same form with seven entries. The two matrices are defined up to scales, subjected to an additional rank-2 constraint. Therefore, $\mathbf{E}_q$ and $\mathbf{F}_q$ have only five degrees of freedom. ■

To address the rank-2 constraint, the quasi-fundamental matrix may be parameterized as follows.

$$\mathbf{F}_q = \begin{bmatrix} 0 & f_{12} & k_2 f_{12} \\ f_{21} & 0 & k_1 f_{21} \\ f_{31} & f_{32} & k_1 f_{31} + k_2 f_{32} \end{bmatrix}. \tag{4.13}$$

The quasi-essential matrix can also be parameterized in a similar way. There are only six entries in (4.13), and the fundamental matrix can be estimated via nonlinear iterations. Under the above parametrization, the epipole in the second image is given by $\mathbf{e}' = [-k_1, -k_2, 1]^T$. It should be noted that the parametrization (4.13) degenerates when $\mathbf{e}'$ lies at infinity or the camera undergoes pure translation. In these cases, it is easy to verify that $f_{12} = f_{21} = 0$. Thus, the first two columns of $\mathbf{F}_q$ are linearly dependent.

**Proposition 4.4**
Given two quasi-perspective camera matrices $\mathbf{P}_q$ and $\mathbf{P}'_q$, the fundamental matrix between the two views can be recovered from $\mathbf{F}_q = [\mathbf{e}']_\times \mathbf{P}'_q \mathbf{P}_q^+$, where $\mathbf{P}_q^+$ denotes the pseudo-inverse of $\mathbf{P}_q$. The fundamental matrix is invariant to any non-singular projective transformation $\mathbf{H} \in \mathbb{R}^{4\times 4}$. i.e. $\mathbf{F}_q$ remains the same if we set $\mathbf{P}_q \leftarrow \mathbf{P}_q\mathbf{H}$ and $\mathbf{P}'_q \leftarrow \mathbf{P}'_q\mathbf{H}$.

**Proof** Similar to the case of perspective projection as in [107], it is easy to obtain the following relationship.

$$\mathbf{F}_q = [\mathbf{e}']_\times \mathbf{P}'_q \mathbf{P}_q^+. \tag{4.14}$$

Suppose a space point $\mathbf{X}_i$ is projected to $\mathbf{x}_i$ and $\mathbf{x}'_i$ via the projection matrices $\mathbf{P}_q$ and $\mathbf{P}'_q$ respectively. If we apply a non-singular projective transformation $\mathbf{H}$ to the world system, i.e. $\mathbf{P}_q \leftarrow \mathbf{P}_q\mathbf{H}$, $\mathbf{P}'_q \leftarrow \mathbf{P}'_q\mathbf{H}$, and $\mathbf{X}_i \leftarrow \mathbf{H}^{-1}\mathbf{X}_i$. It is easy to verify that the transformation does not change the images $\mathbf{x}_i$ and $\mathbf{x}'_i$. Thus, the camera pairs $\{\mathbf{P}_q, \mathbf{P}'_q\}$ and $\{\mathbf{P}_q\mathbf{H}, \mathbf{P}'_q\mathbf{H}\}$ correspond to the same fundamental matrix as:

$$\mathbf{F}_q = [\mathbf{e}']_\times \mathbf{P}'_q \mathbf{P}_q^+ = [\mathbf{e}']_\times (\mathbf{P}'_q\mathbf{H})(\mathbf{P}_q\mathbf{H})^+ \tag{4.15}$$

which indicates that the quasi-fundamental matrix is invariant to the transformation $\mathbf{H}$. Specifically, we can choose a certain transformation matrix to register the first camera to the world system and obtain the following projection matrices:

$$\mathbf{P}_q = \mathbf{K}[\,\mathbf{I}\,|\,0\,],\ \mathbf{P}'_q = \mathbf{K}'[\,\mathbf{R}\,|\,\mathbf{t}\,]. \tag{4.16}$$

Then, the epipole in the second image equals to $\mathbf{e}' = \mathbf{K}'\mathbf{t}$, and the fundamental matrix can be expressed by substituting (4.16) into (4.14).

$$\begin{aligned}\mathbf{F}_q =& [\mathbf{K}'\mathbf{t}]_\times \mathbf{K}'[\mathbf{R}\,|\,\mathbf{t}]\Big(\mathbf{K}[\mathbf{I}\,|\,0]\Big)^+ = \mathbf{K}'^{-T}\mathbf{t}_\times \mathbf{R}\mathbf{K}^{-1} \\ =& \mathbf{K}'^{-T}\mathbf{E}_q\mathbf{K}^{-1}. \end{aligned} \tag{4.17}$$

Equation (4.17) derives (4.12) from a different viewpoint. ■

**Remark 4.1**

For the computation of a general fundamental matrix under perspective projection, we may adopt a normalized 8-point linear algorithm [35], iterative minimization algorithm of Sampson distance [93], 7-point nonlinear algorithm with rank-2 constraint [107], or the Gold Standard algorithm [32]. Please refer to [32][107] for more details and a comparison of the above algorithms. Similarly, we have normalized 6-point linear algorithm and 5-point nonlinear algorithm for the estimation of quasi-perspective fundamental matrix. Usually, we adopt the linear algorithm for initial estimation, and utilize the Gold Standard algorithm to further optimize the fundamental matrix.

**Remark 4.2**

Under affine assumption, the optical center of an affine camera locates at infinity, it follows that all epipolar lines are parallel and both epipoles are at infinity. Thus, the affine fundamental matrix is simplified to the form

$$\mathbf{F}_a = \begin{bmatrix} 0 & 0 & * \\ 0 & 0 & * \\ * & * & * \end{bmatrix}$$

which is already a rank-2 matrix with four degrees of freedom.

### 4.3.2 Plane induced homography

For coplanar space points, their images in two views are related with a planar homography, which is named as plane induced homography.

**Proposition 4.5**

Under quasi-perspective projection, the plane induced homography can be simplified to the form of

$$\mathbf{H}_q = \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{bmatrix}$$

Table 4.1: The entry number and degrees of freedom (DOF) of different geometric matrices.

| Model | | Projection matrix | Perspectivity matrix | Fundamental matrix | Essential matrix | Homography matrix |
|---|---|---|---|---|---|---|
| Entry (DOF) | Persp | 12 (11) | 9 (8) | 9 (7) | 9 (7) | 9 (7) |
| | Quasi | 10 (9) | 7 (6) | 7 (5) | 7 (5) | 7 (6) |
| | Affine | 9 (8) | 7 (6) | 5 (4) | 5 (4) | 7 (6) |

with six degrees of freedom.

**Proof** Suppose $\mathbf{x}$ and $\mathbf{x}'$ are a pair of images of a coplanar space point $\mathbf{X}$ in the two views, and $\mathbf{H}_\pi$ and $\mathbf{H}'_\pi$ are the perspective homographies of the two views. Then, from (4.6) we have

$$\mathbf{x} \simeq \mathbf{H}_\pi \mathbf{X},\ \mathbf{x}' \simeq \mathbf{H}'_\pi \mathbf{X}. \tag{4.18}$$

By eliminating $\mathbf{X}$ from (4.18), we get

$$\mathbf{x}' \simeq \mathbf{H}'_\pi \mathbf{H}_\pi^{-1} \mathbf{x} \tag{4.19}$$

where $\mathbf{H}_q = \mathbf{H}'_\pi \mathbf{H}_\pi^{-1}$ is called plane induced homography which can be expanded as

$$\mathbf{H}_q = \mathbf{H}'_\pi \mathbf{H}_\pi^{-1} = \begin{bmatrix} h'_{11} & h'_{12} & h'_{13} \\ h'_{21} & h'_{22} & h'_{23} \\ 0 & 0 & h'_{33} \end{bmatrix} \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & h_{33} \end{bmatrix}^{-1} = \begin{bmatrix} * & * & * \\ * & * & * \\ 0 & 0 & * \end{bmatrix}. \tag{4.20}$$

The homography $\mathbf{H}_q$ is a full rank matrix with six degrees of freedom, and at least three non-collinear corresponding points can give a unique solution. ■

It is easy to verify that the homography under an affine camera model has the same form as (4.20). In contrast, a general homography under a perspective model has eight degrees of freedom and at least four points are required for computation. A comparison of the entry numbers and degrees of freedom under different camera models of above discussed geometric matrices are tabulated in Table 4.1.

**Proposition 4.6**
Given a fundamental matrix $\mathbf{F}_q$, the plane induced homography $\mathbf{H}_q$ may be recovered from two pairs of correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i, i = 1, 2$.

The result is obvious, since an additional correspondence of the epipoles $\mathbf{e} \longleftrightarrow \mathbf{e}'$ can be obtained from the fundamental matrix as

$$\mathbf{F}_q\mathbf{e} = 0,\ \mathbf{F}_q^T\mathbf{e}' = 0. \tag{4.21}$$

Therefore, if the two image points $\mathbf{x}_i, i = 1, 2$ are not collinear with the epipole $\mathbf{e}$, $\mathbf{H}_q$ can be uniquely determined from the three correspondences.

The homography $\mathbf{H}_\infty$ induced by the plane at infinity is called infinite homography. $\mathbf{H}_\infty$ is of great importance in stereo vision since it is closely related with camera calibration and affine reconstruction. According to the Proposition 4.6, the infinite homography may be computed from the correspondences of two vanishing points if the fundamental matrix is known. This is an interesting result for quasi-perspective projection model.

### 4.3.3 RANSAC computation

In the above analysis, we assume all correspondences are inliers without mismatches. However, mismatches are inevitable in real applications, and the result may be severely disturbed in presence of outliers. In this case, we usually adopt the RANSAC algorithm [26] to eliminate outliers and obtain a robust estimation. RANSAC algorithm is an iterative method to estimate parameters of a mathematical model and is computationally intensive. We will present a comparison on the number of trials required for different projection models.

Suppose the outlier-to-inlier ratio is $k = N_{outlier}/N_{inlier}$, and the number of the minimum subset required to estimate the model is $n$. We want to ensure that at least one of the random samples is free from outliers with a probability of $p$. Then, the trial number $N$ must satisfy

$$1 - p = \left(1 - (\frac{1}{k+1})^n\right)^N \tag{4.22}$$

which leads to

$$N = \frac{ln(1-p)}{ln\left(1 - (\frac{1}{k+1})^n\right)}. \tag{4.23}$$

Under the given probability $p$, the number of trials depend on the outlier-to-inlier ratio $k$ and the subset number $n$. In practice, we usually select a

Table 4.2: The number of trials required for different models to ensure probability $p = 99\%$ with respect to different minimal subsets and outlier-to-inlier ratios .

| Model | | Minimal subset | Outlier-to-inlier ratio 10% | 20% | 40% | 60% | 80% | 100% |
|---|---|---|---|---|---|---|---|---|
| Fundamental | Persp | 8/7 | 8/7 | 18/15 | 66/47 | 196/122 | 506/280 | 1177/588 |
| | Quasi | 6/5 | 6/5 | 12/9 | 33/23 | 75/46 | 155/85 | 293/146 |
| | Affine | 4 | 5 | 7 | 16 | 28 | 47 | 72 |
| Homography | Persp | 4 | 5 | 7 | 16 | 28 | 47 | 72 |
| | Quasi | 3 | 4 | 6 | 11 | 17 | 25 | 35 |
| | Affine | 3 | 4 | 6 | 11 | 17 | 25 | 35 |

conservative probability as $p = 0.99$. Table 4.2 shows the required number of trials under different conditions.

We can conclude from Table 4.2 that the required number of trials increases sharply with the increase in subset number $n$ and outlier-to-inlier ratio $k$. The quasi-perspective algorithm is computationally less intensive than the perspective projection algorithm, especially for large proportion of outliers. As noted in Remark 4.1, we may adopt a normalized 8-point linear algorithm or a 7-point nonlinear algorithm for fundamental estimation under perspective projection. Accordingly, we have a 6-point linear algorithm and a 5-point nonlinear algorithm for the computation of quasi-fundamental matrix. We can use the simple linear algorithm when the ratio $k$ is small. However, it is wise to adopt a nonlinear algorithm for large outlier ratios so as to speed up the computation.

## 4.4 3D Structure Reconstruction

Quasi-perspective projection is a special case of the full perspective projection. Thus, most theories on 3D reconstruction under the perspective model may be applied directly to the quasi-perspective model. Some important properties of quasi-perspective reconstruction are summarized as follows.

**Proposition 4.7**
Under quasi-perspective assumption, a pair of canonical cameras can be defined as

$$\mathbf{P}_q = [\,\mathbf{I}\,|\,\mathbf{0}\,],\ \mathbf{P}'_q = [\,\mathbf{M}_q\,|\,\mathbf{t}\,] \tag{4.24}$$

where $\mathbf{M}_q$ is a $3 \times 3$ matrix with its last row of the form $[\,0, 0, *\,]$.

**Proposition 4.8**
Suppose $(\mathbf{P}_{q1}, \mathbf{P}'_{q1}, \{\mathbf{X}_{1i}\})$ and $(\mathbf{P}_{q2}, \mathbf{P}'_{q2}, \{\mathbf{X}_{2i}\})$ are two quasi-perspective reconstructions of a set of correspondences $\mathbf{x}_i \longleftrightarrow \mathbf{x}'_i$ between two images. Then, the two reconstructions are defined up to a quasi-perspective transformation as below.

$$\mathbf{P}_{q2} = \mathbf{P}_{q1}\mathbf{H}_q,\ \mathbf{P}'_{q2} = \mathbf{P}'_{q1}\mathbf{H}_q,\ \mathbf{X}_{2i} = \mathbf{H}_q^{-1}\mathbf{X}_{1i}$$

where the transformation $\mathbf{H}_q$ is a $4 \times 4$ non-singular matrix of the form

$$\mathbf{H}_q = \begin{bmatrix} \mathbf{A}_{2\times 2} & \mathbf{B}_{2\times 2} \\ \mathbf{0}_{2\times 2} & \mathbf{C}_{2\times 2} \end{bmatrix}.$$

Under quasi-perspective transformation $\mathbf{H}_q$, we have

$$\mathbf{P}_{q2}\mathbf{X}_{2i} = (\mathbf{P}_{q1}\mathbf{H}_q)(\mathbf{H}_q^{-1}\mathbf{X}_{1i}) = \mathbf{P}_{q1}\mathbf{X}_{1i} = \mathbf{x}_i \tag{4.25}$$

$$\mathbf{P}'_{q2}\mathbf{X}_{2i} = (\mathbf{P}'_{q1}\mathbf{H}_q)(\mathbf{H}_q^{-1}\mathbf{X}_{1i}) = \mathbf{P}'_{q1}\mathbf{X}_{1i} = \mathbf{x}'_i. \tag{4.26}$$

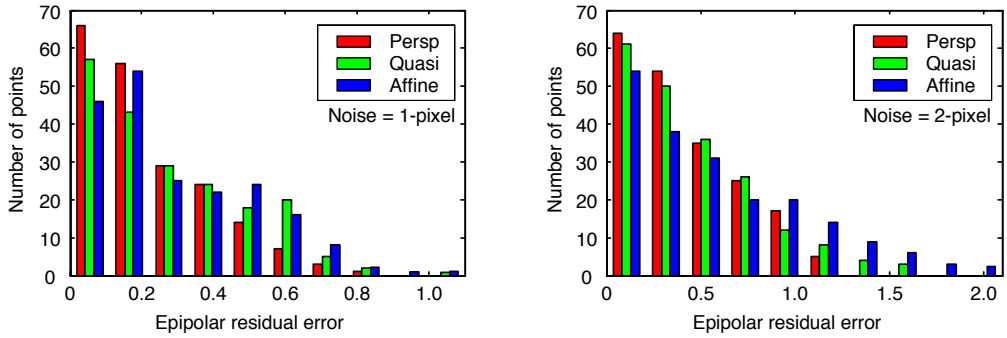
It is easy to verify that the transformed camera matrices $\mathbf{P}_{q2} = \mathbf{P}_{q1}\mathbf{H}_q$ and $\mathbf{P}'_{q2} = \mathbf{P}'_{q1}\mathbf{H}_q$ have the same forms as the quasi-perspective projection matrices as given in Proposition 4.1. The transformed space points can be written as

$$\mathbf{X}_{2i} = \mathbf{H}_q^{-1}\mathbf{X}_{1i} = \begin{bmatrix} \mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{C}^{-1} \\ \mathbf{0} & \mathbf{C}^{-1} \end{bmatrix} \mathbf{X}_{1i}. \tag{4.27}$$

It is obvious that the parallelism along $X$ and $Y$ axes are preserved under the transformation $\mathbf{H}_q$, since ideal points $\mathbf{X}_{1i} = [1,0,0,0]^T$ or $\mathbf{X}_{1i} = [0,1,0,0]^T$ are mapped to ideal points according to (4.27).

For the recovery of 3D structure and camera motions, we may adopt a stratified reconstruction algorithm [25] to refine the structure from perspective to affine, and finally to the Euclidean space. In this chapter, we assume calibrated cameras. Therefore, the metric structure can be recovered directly from singular value decomposition (SVD) of the essential matrix [32]. The implementation of the reconstruction algorithm is summarized as follows.

1. Establish initial correspondences between the two images according to the method in [80];




(a) (b)

Figure 4.2: Histogram distribution of the epipolar residual errors under different camera models. (a) The result obtained with 1-pixel Gaussian noise; and (b) the result obtained with 2-pixel Gaussian noise.

2. Estimate the quasi-fundamental matrix via RANSAC algorithm and eliminate outliers;

3. Optimize the fundamental matrix via the Gold Standard algorithm as stated in Remark 4.1 and recover the essential matrix from (4.8);

4. Perform SVD decomposition on the essential matrix and extract the camera projection matrices according to the method in [32], which will give four pairs of solutions;

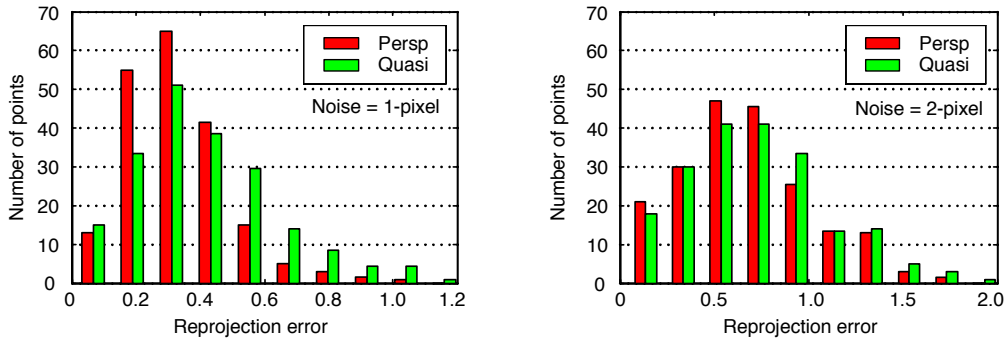
5. Resolve ambiguity in the solutions via trial and error. Take one pair of matching points as reference and reconstruct it from the above four solutions, only the true solution can make the reconstructed point lie in front of both cameras;

6. Compute the 3D structure of all correspondences via triangulation from the recovered camera matrices;

7. Optimize the solution via bundle adjustment [32].

## 4.5 Evaluations on Synthetic Data

During simulation, we randomly generated 200 points within a cube of $20 \times 20 \times 20$ in space, and simulated two images from these points by perspective projection. The image size is set at $800 \times 800$. The camera parameters are set as follows: focal lengths are set randomly between 1000 and 1100. Three rotation




(a) (b)

Figure 4.3: Histogram distribution of reprojection errors by plane induced homography. (a) The result obtained with 1-pixel Gaussian noise; and (b) the result obtained with 2-pixel Gaussian noise.

angles are set randomly between $\pm 5°$. The $X$ and $Y$ positions of the cameras are set randomly between $\pm 15$, while the $Z$ positions are set randomly between 210 to 220. The synthetic imaging conditions are very close to affine and the quasi-perspective assumptions.

### 4.5.1 Fundamental matrix and homography

We recovered the quasi-fundamental matrix $\mathbf{F}_q$ by normalized 6-point algorithm and calculated the epipolar residual error $\varepsilon_{1i}$ which is defined as the distance of a point to the associated epipolar line.

$$\varepsilon_{1i} = \frac{1}{2}\Big(dis(\mathbf{x}_i, \mathbf{F}_q^T \mathbf{x}_i')^2 + dis(\mathbf{x}_i', \mathbf{F}_q \mathbf{x}_i)^2\Big) \tag{4.28}$$

where $'dis(*,*)'$ denotes the Euclidean distance from a point to a line. The histogram distribution of the errors across all 200 correspondences is outlined in Fig. 4.2. Gaussian image noise was added to each image point during the test. As a comparison, we also recovered the general fundamental matrix $\mathbf{F}$ by normalized 8-point algorithm and the affine fundamental matrix $\mathbf{F}_a$ by normalized 4-point algorithm. We see that the error of the quasi-perspective projection lies in between those of the perspective projection and affine projection models. Thus, the quasi-perspective fundamental matrix is a better approximation than the affine fundamental matrix.

To evaluate the accuracy of homography estimation, we set all space points on the plane $Z = 10$ and regenerated two images with the same camera parameters. Then, we recovered the plane induced homography $\mathbf{H}_q$ and $\mathbf{H}$ under

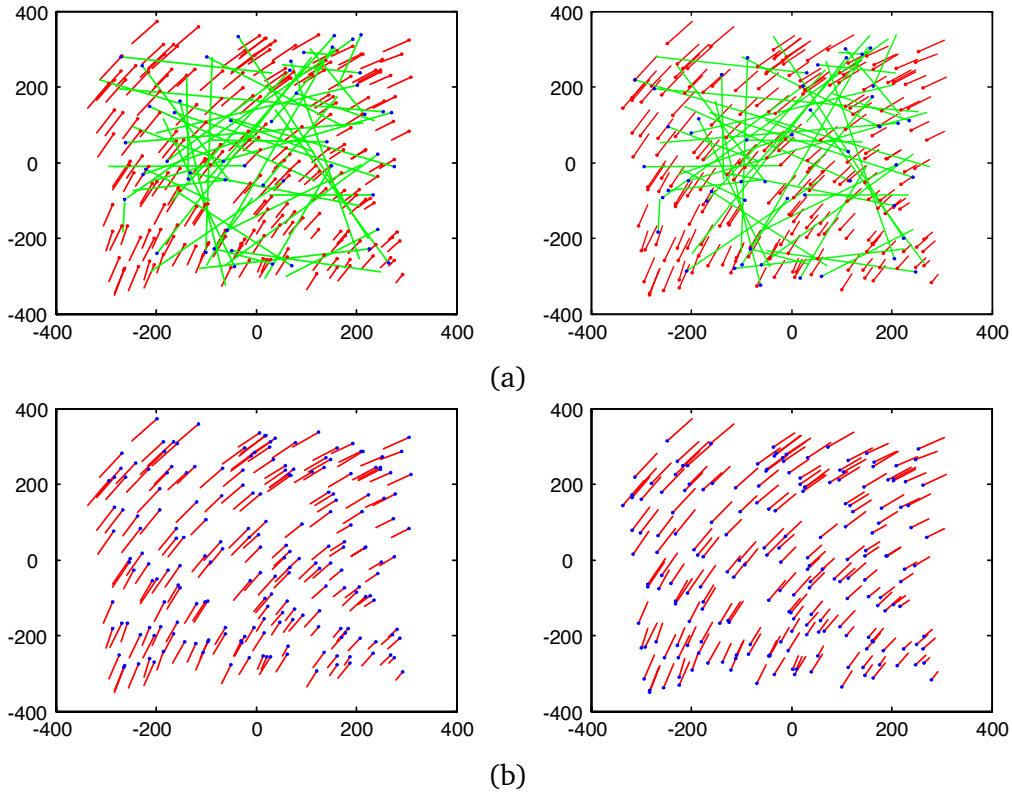



Figure 4.4: Result of outlier removal. (a) Initial feature matches in two images with outliers; and (b) final detected correspondences after RANSAC algorithm.

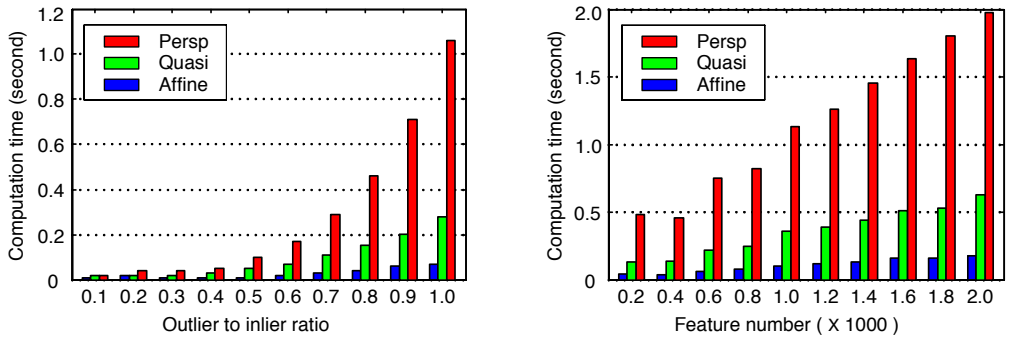
quasi-perspective and perspective projection, respectively, and evaluated the reprojection error defined as

$$\varepsilon_{2i} = \frac{1}{2}\Big(d(\mathbf{x}_i, \mathbf{H}_q^{-1}\mathbf{x}_i')^2 + d(\mathbf{x}_i', \mathbf{H}_q\mathbf{x}_i)^2\Big) \tag{4.29}$$

where $'d(*,*)'$ denotes the Euclidean distance between two feature points. The histogram distribution of the errors under different noise levels is shown in Fig. 4.3. It is clear that the error obtained from $\mathbf{H}_q$ is higher than that from $\mathbf{H}$. The homography under the affine model is the same as $\mathbf{H}_q$ as noted in Section 4.3.2.

### 4.5.2 Outlier removal

We randomly added 50 mismatches to the initial generated correspondences. The initial matches with disparities and outliers are shown in Fig. 4.4. We adopt the RANSAC paradigm to remove outliers and estimate the quasi-fundamental matrix. As shown in Fig. 4.4, all mismatches were detected by the algorithm.



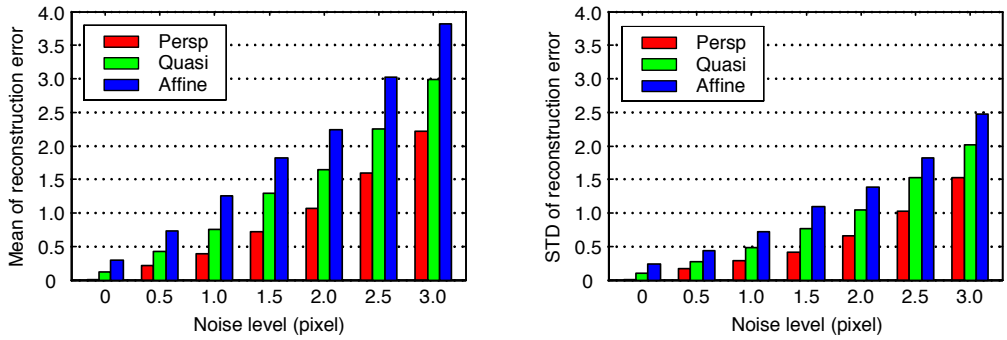

(a) (b)

Figure 4.5: Average computation time under different camera models. (a) The time with respect to different outlier-to-inlier ratios; and (b) the time with respect to different feature point numbers.

We compared the average computation time in estimating the fundamental matrix under different models. Only linear algorithm was adopted and the minimal subsets for $\mathbf{F}$, $\mathbf{F}_q$, $\mathbf{F}_a$ are set as 8, 6, and 4, respectively. The program was implemented with Matlab R14 on Dell Inspiron 600m laptop of Pentium®1.8GHz CPU. In the first case, we select 200 correspondences and vary the outlier-to-inlier ratio from 0.1 to 1.0. In the second case, we set the outlier ratio at 0.8 and vary the feature number from 200 to 2000. The result is shown in Fig. 4.5. It is evident that the algorithm runs significantly faster using the quasi-perspective model than using the perspective projection model, especially for larger data sets and higher outlier ratios.

### 4.5.3 Reconstruction result

We reconstructed the 200 data points under different camera models according to the algorithm presented in Section 4.4. The reconstruction is defined up to a Euclidean rotation and translation with respect to the ground truth. We recover these parameters and register the result with the ground truth. The reconstruction error is defined as point-wise distance between the recovered structure and its ground truth. In order to obtain a statistically meaningful result, we vary the image noise level from 0 to 3 pixels in steps of 0.5, and take 100 independent tests at each noise level. The mean and standard deviation of the errors are shown in Fig. 4.6. The experimental results show that the reconstruction accuracy obtained using the quasi-perspective model lies in between those obtained by the full perspective and affine projection models. The quasi-perspective model is more accurate than the affine model.




(a) (b)

Figure 4.6: Evaluation on 3D reconstruction accuracy by different models. (a) The mean of the reconstruction errors at different noise levels; and (b) the corresponding standard deviation.

Table 4.3: The reprojection error under different projection models.

| Model | | Persp | Quasi | Affine |
|---|---|---|---|---|
| Error | Stone dragon | 0.72 | 0.80 | 0.86 |
| | Medusa head | 0.97 | 1.04 | 1.13 |

## 4.6 Evaluations on Real Images

### 4.6.1 Test on stone dragon images

We tested and compared different models on many real images and we report two results here. The correspondences in the tests were established by a matching system based on SIFT and the epipolar constraint [80], and the camera parameters were calibrated offline via the method in Zhang [104].

The stone dragon images were captured by a Canon G3 camera in Chung Chi College of the Chinese University of Hong Kong. The image resolution is $1024 \times 768$, and 4261 reliable features were established by the system [80]. We recover the fundamental matrix via Gold Standard algorithm and reconstructed the 3D Euclidean structure according to the process in Section 4.4. Figure 4.7 shows the matched features, reconstructed VRML model with texture, and the corresponding wireframe model viewed from different viewpoints. The structure of the dragon is correctly recovered using the quasi-perspective model. The distributions of epipolar residual errors using the three models are compared in Figure 4.8, and the reprojection errors are listed in Table 4.3.

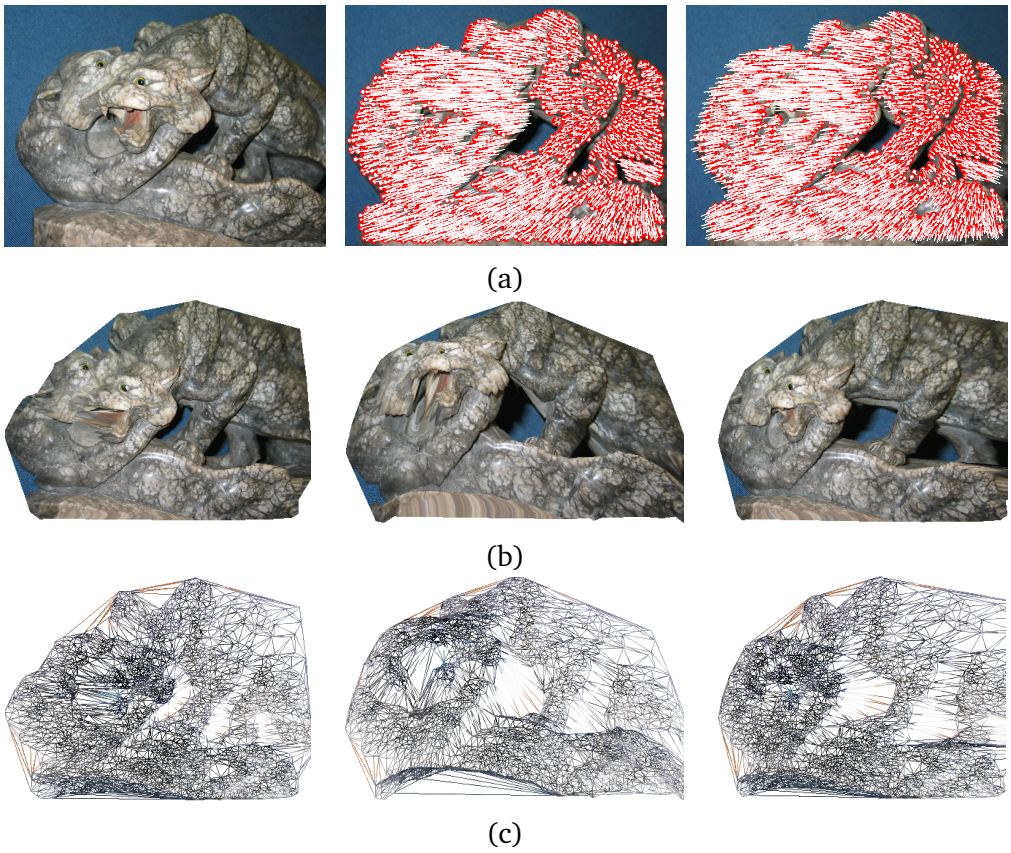
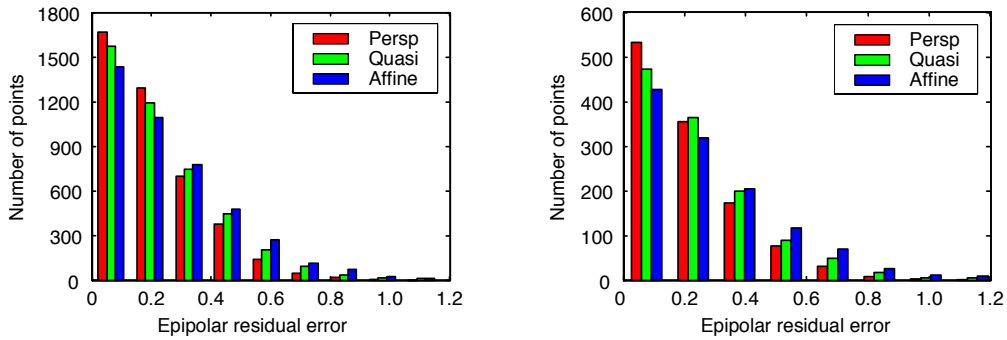


Figure 4.7: Reconstruction result of stone dragon images. (a) Two images of the stone dragon overlaid by tracked features with relative disparities; (b) reconstructed VRML model of the scene shown from different viewpoints with texture mapping; and (c) the corresponding wireframe of the VRML model.

### 4.6.2 Test on Medusa head images

The Medusa head image sequence was downloaded from Dr. Pollefeys's homepage which was taken in the ancient city of Sagalassos, Turkey. It was recorded by a Sony TRV900 camcorder with a resolution of $720 \times 576$. We selected two adjacent frames and in total 1216 correspondences were established. The camera's intrinsic parameters were estimated by the method [88]. We recover the fundamental matrix and reconstruct its structure by the proposed algorithm. The reconstructed VRML model and the corresponding wireframe model are shown in Figure 4.9. The result is realistic and visually correct despite of the unavailability of the ground truth. As a quantitative evaluation, we calculated the reprojection errors by the three projection models as tabulated in Table 4.3. The histogram




(a) (b)

Figure 4.8: Histogram distribution of the epipolar residual errors under different camera models. (a) Fountain base images; (b) stone dragon images; and (c) Medusa head images.

distribution of the epipolar residual errors by different models are shown in Figure 4.8. The result by the quasi-perspective model is better than that by the affine assumption.

## 4.7 Conclusion

In this chapter, we have investigated the one-view and two-view geometry of the quasi-perspective projection model and presented some special properties of the quasi fundamental matrix, plane induced homography, and 3D reconstruction under the proposed model. Both theoretical analysis and experimental results show that the quasi-perspective model is a good trade-off between the simplicity of the affine assumption and the accuracy of the full perspective projection model. The result is consistent with our previous study on structure and motion factorization [83]. The proposed method is suitable to deal with images with small baseline.

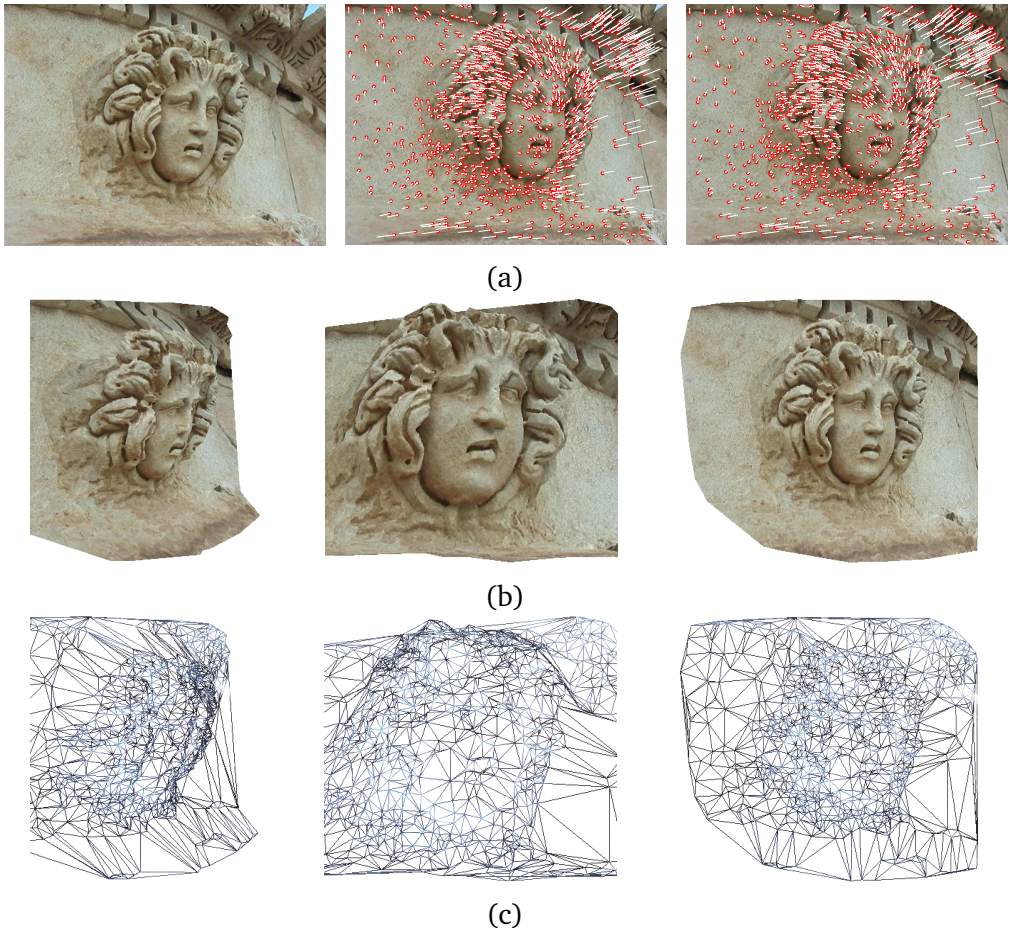


Figure 4.9: Reconstruction result of Medusa head images. (a) Two images of the fountain base overlaid by tracked features with relative disparities; (b) reconstructed VRML model of the scene shown from different viewpoints with texture; and (c) the corresponding triangulated wireframe of the VRML model.

# Chapter 5

# SfM Based on Quasi-Perspective Projection Model

Previous studies on structure and motion factorization are either based on a simplified affine assumption or a general perspective projection model. The affine approximation is widely adopted due to its simplicity, whereas the extension to perspective model suffers from difficulties in projective depth recovery. To fill the gap between simplicity of affine and accuracy of perspective model, we propose a quasi-perspective model based algorithm for structure and motion recovery of both rigid and nonrigid objects. In this chapter, we first establish a framework of rigid and nonrigid factorization under quasi-perspective assumption. Then, we propose an extended Cholesky decomposition to recover the rotation part of the Euclidean upgrading matrix. Finally, we prove that the last column of the upgrading matrix corresponds to a global scale and translation of the camera thus may be set freely. The proposed algorithm is validated and evaluated extensively on synthetic and real image sequences.

## 5.1 Introduction

The factorization algorithm is a powerful and efficient method for structure and motion recovery. Since Tomasi and Kanade [74] first introduced the algorithm in the early 90's, numerous extensions and generalizations have been proposed. Most early studies on the problem assume a rigid object and an affine camera model [62][65]. The main difficulty of its extension to perspective projection lies in the recovery of depth scales. One method is to estimate the depths in a pair-wise fashion via epipolar constraint [71][77], which may be unstable due to possible error accumulation. Another method is based on nonlinear iteration

by minimizing reprojections [47]. These methods rely on the accuracy of initial affine solution. Oliensis and Hartley [56] recently proved that no iteration converges sensibly. Wang and Wu [88] proposed a hybrid method to initialize the depth scales via a projective reconstruction.

In recent years, many extensions stemming from the factorization algorithm were proposed to relax the rigidity constraint to multiple moving objects [16][44] and articulated objects [97][98]. Bregler *et al.* [11] firstly established the framework of nonrigid factorization using shape bases. The method was extensively investigated and developed under the affine assumption [10][21][75][95]. It has been extended to perspective projection in [81][96]. Rabaud and Belongie [66] relaxed the Bregler's assumption and proposed to solve the problem by a manifold-learning framework. Wang *et al.* [82] introduced a rotation constrained power factorization algorithm. Hartley and Vidal [34] proposed a closed form solution to the nonrigid shape and motion with known camera constraints.

The affine camera model is widely adopted in factorization due to its simplicity. However, the accuracy of this approximation is not satisfactory in many applications. Perspective projection based algorithms are computationally intensive and thus convergence is not guaranteed. In this chapter, we apply the quasi-perspective projection model [83] to both a rigid and a nonrigid factorization framework. This is a trade-off between the simplicity of the affine and the accuracy of the full perspective projection. It is demonstrated to be more accurate than affine approximation since the projective depths in quasi-perspective projection are implicitly embedded in the motion and shape matrices. Thus, the difficult problem of depth recovery in perspective factorization is circumvented. Part of this chapter was published in International Journal of Computer Vision [86].

The remaining part of this chapter is organized as follows. The factorization algorithm is briefly reviewed in Section 5.2. The proposed quasi-perspective factorization algorithm for rigid objects is detailed in Section 5.3. The nonrigid factorization under quasi-perspective projection is presented in Section 5.4. Some experimental evaluations on synthetic and real image sequences are reported in Section 5.5 and 5.6 respectively.

## 5.2 Background on Factorization

We already introduced the rigid and nonrigid factorization under the affine and perspective projection models in previous chapters. For convenience of discussion, we present a brief review in this section on the expressions of different factorization algorithms.

Under perspective projection, a 3D point $\mathbf{X}_j$ is imaged at $\mathbf{x}_{ij}$ in the $i$-th frame

according to equation

$$\lambda_{ij}\mathbf{x}_{ij} = \mathbf{P}_i\mathbf{X}_j = \mathbf{K}_i[\mathbf{R}_i, \mathbf{t}_i]\mathbf{X}_j. \tag{5.1}$$

If we adopt affine projection model and register all image points to the centroid. Then, the projection process (5.1) is simplified to the form

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i\bar{\mathbf{X}}_j. \tag{5.2}$$

Given $n$ tracked features of an object across a sequence of $m$ frames. The structure and motion factorization under affine assumption (5.2) can be expressed as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_m \end{bmatrix}}_{\mathbf{M}_{2m\times 3}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n \end{bmatrix}}_{\bar{\mathbf{S}}_{3\times n}}. \tag{5.3}$$

The factorization is usually performed by SVD decomposition of the tracking matrix $\mathbf{W}$ with rank-3 constraint. When the perspective projection model (5.1) is adopted, the factorization can be modeled as

$$\underbrace{\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix}}_{\dot{\mathbf{W}}_{3m\times n}} = \underbrace{\begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_m \end{bmatrix}}_{\mathbf{M}_{3m\times 4}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n \\ 1, \cdots, 1 \end{bmatrix}}_{\mathbf{S}_{4\times n}}. \tag{5.4}$$

The rank of the projective depth-scaled tracking matrix $\dot{\mathbf{W}}$ is at most 4 if a consistent set of scalars are present.

When an object is nonrigid, we follow Bregler's assumption [11] to model the nonrigid structure by a linear combination of some shape bases.

$$\bar{\mathbf{S}}_i = \sum_{l=1}^{k} \omega_{il}\mathbf{B}_l. \tag{5.5}$$

With this assumption, the nonrigid factorization under affine camera model is expressed as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \omega_{11}\mathbf{A}_1 & \cdots & \omega_{1k}\mathbf{A}_1 \\ \vdots & \ddots & \vdots \\ \omega_{m1}\mathbf{A}_m & \cdots & \omega_{mk}\mathbf{A}_m \end{bmatrix}}_{\mathbf{M}_{2m\times 3k}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \end{bmatrix}}_{\bar{\mathbf{B}}_{3k\times n}} \tag{5.6}$$

and the rank of the nonrigid tracking matrix is at most $3k$. Similarly, the factorization under perspective projection can be formulated as follows [96].

$$\underbrace{\begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \cdots & \lambda_{mn}\mathbf{x}_{mn} \end{bmatrix}}_{\dot{\mathbf{W}}_{3m\times n}} = \underbrace{\begin{bmatrix} \omega_{11}\mathbf{P}_1^{(1:3)} & \cdots & \omega_{1k}\mathbf{P}_1^{(1:3)} & \mathbf{P}_1^{(4)} \\ \vdots & \ddots & \vdots & \vdots \\ \omega_{m1}\mathbf{P}_m^{(1:3)} & \cdots & \omega_{mk}\mathbf{P}_m^{(1:3)} & \mathbf{P}_m^{(4)} \end{bmatrix}}_{\mathbf{M}_{3m\times(3k+1)}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \\ \mathbf{1}^T \end{bmatrix}}_{\mathbf{B}_{(3k+1)\times n}}. \tag{5.7}$$

The rank of the correctly scaled tracking matrix in (5.7) is at most $3k+1$. Just as its rigid counterpart, the most difficult issue for perspective factorization is to determine the projective depths that are consistent with (5.1).

## 5.3 Quasi-Perspective Rigid Factorization

Under the assumption that the camera is far away from the object with small lateral rotations, we proposed a quasi-perspective projection model to simplify the imaging process as follows.

$$\mathbf{x}_{ij} = \mathbf{P}_{qi}\mathbf{X}_{qj} = (\mu_i\mathbf{P}_i)(\ell_j\mathbf{X}_j). \tag{5.8}$$

In quasi-perspective projection (5.8), the projective depths are implicitly embedded in the scalars of the homogeneous structure $\mathbf{X}_{qj}$ and the projection matrix $\mathbf{P}_{qi}$. Thus, the difficult problem of estimating the unknown depths is circumvented. The model is more general than the affine projection model (5.2), where all projective depths are simply assumed to be equal. Under the quasi-perspective assumption, the factorization equation of a tracking matrix is expressed as

$$\begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{x}_{m1} & \cdots & \mathbf{x}_{mn} \end{bmatrix} = \begin{bmatrix} \mu_1\mathbf{P}_1 \\ \vdots \\ \mu_m\mathbf{P}_m \end{bmatrix} [\ell_1\mathbf{X}_1, \cdots, \ell_n\mathbf{X}_n] \tag{5.9}$$

which can be written concisely as

$$\tilde{\mathbf{W}}_{3m\times n} = \mathbf{M}_{3m\times 4}\mathbf{S}_{4\times n}. \tag{5.10}$$

The form is similar to perspective factorization (5.4). However, the projective depths in (5.9) are embedded in the motion and shape matrices, hence there is no need to estimate them explicitly. By performing SVD on the tracking matrix and imposing the rank-4 constraint, $\tilde{\mathbf{W}}$ may be factorized as $\hat{\mathbf{M}}_{3m\times 4}\hat{\mathbf{S}}_{4\times n}$. However, like all other factorization algorithms, the decomposition is not unique since it is

defined up to a nonsingular linear transformation $\mathbf{H}_{4\times4}$ as $\mathbf{M}=\hat{\mathbf{M}}\mathbf{H}$ and $\mathbf{S}=\mathbf{H}^{-1}\hat{\mathbf{S}}$. Due to the special form of (5.9), the upgrading matrix has some special properties compared to that under affine and perspective projection. We will present the computational details in the following subsections.

### 5.3.1 Euclidean upgrading matrix

We adopt the metric constraints to compute the upgrading matrix $\mathbf{H}_{4\times4}$. Let us decompose the matrix into two parts as

$$\mathbf{H}=[\mathbf{H}_l|\mathbf{H}_r] \tag{5.11}$$

where $\mathbf{H}_l$ denotes the first three columns, and $\mathbf{H}_r$ denotes the fourth column. Suppose $\hat{\mathbf{M}}_i$ is the $i$-th triple rows of $\hat{\mathbf{M}}$, we have

$$\hat{\mathbf{M}}_i\mathbf{H}=[\hat{\mathbf{M}}_i\mathbf{H}_l|\hat{\mathbf{M}}_i\mathbf{H}_r] \tag{5.12}$$

where

$$\hat{\mathbf{M}}_i\mathbf{H}_l=\mu_i\mathbf{P}_i^{(1:3)}=\mu_i\mathbf{K}_i\mathbf{R}_i \tag{5.13}$$

$$\hat{\mathbf{M}}_i\mathbf{H}_r=\mu_i\mathbf{P}_i^{(4)}=\mu_i\mathbf{K}_i\mathbf{t}_i \tag{5.14}$$

where $\mathbf{P}_i^{(1:3)}$ and $\mathbf{P}_i^{(4)}$ denote the first three columns and the fourth column of $\mathbf{P}_i$. In the following, we will show how to compute $\mathbf{H}_l$ and $\mathbf{H}_r$.

#### Recovering $\mathbf{H}_l$

Let us denote $\mathbf{Q}=\mathbf{H}_l\mathbf{H}_l^T$, which is a $4\times4$ symmetric matrix. As in previous factorization studies [31][65], we adopt a simplified camera model with only one parameter as $\mathbf{K}_i=diag(f_i,f_i,1)$. Then, from

$$\begin{aligned}\mathbf{C}_i&=\hat{\mathbf{M}}_i\mathbf{Q}\hat{\mathbf{M}}_i^T=\left(\hat{\mathbf{M}}_i\mathbf{H}_l\right)\left(\hat{\mathbf{M}}_i\mathbf{H}_l\right)^T=(\mu_i\mathbf{K}_i\mathbf{R}_i)(\mu_i\mathbf{K}_i\mathbf{R}_i)^T\\&=\mu_i^2\mathbf{K}_i\mathbf{K}_i^T=\mu_i^2\begin{bmatrix}f_i^2&&\\&f_i^2&\\&&1\end{bmatrix}\end{aligned} \tag{5.15}$$

we obtain the following constraints.

$$\begin{cases}\mathbf{C}_i(1,2)=\mathbf{C}_i(2,1)=0\\\mathbf{C}_i(1,3)=\mathbf{C}_i(3,1)=0\\\mathbf{C}_i(2,3)=\mathbf{C}_i(3,2)=0\\\mathbf{C}_i(1,1)-\mathbf{C}_i(2,2)=0\end{cases} \tag{5.16}$$

Since the factorization (5.9) can be defined up to a global scalar as $\tilde{\mathbf{W}} = \mathbf{MS} = (\varepsilon\mathbf{M})(\mathbf{S}/\varepsilon)$, we set $\mu_1 = 1$ to avoid the trivial solution of $\mathbf{Q} = \mathbf{0}$. Thus, we have $4m+1$ linear constraints in total on the ten unknowns of $\mathbf{Q}$, which can be solved via least squares. Ideally, $\mathbf{Q}$ is a positive semidefinite symmetric matrix, the matrix $\mathbf{H}_l$ can be recovered from $\mathbf{Q}$ via extended Cholesky decomposition as follows.

**Definition 5.1 (Vertical extended upper triangular matrix)**
Suppose $\mathbf{U}$ is a $n \times k$ $(n > k)$ matrix. We call $\mathbf{U}$ a vertical extended upper triangular matrix if it is of the form

$$\mathbf{U}_{ij} = \begin{cases} u_{ij} & \text{if } i \leq j + (n-k) \\ 0 & \text{if } i > j + (n-k) \end{cases} \tag{5.17}$$

where $\mathbf{U}_{ij}$ denotes the $(i,j)$-th element of $\mathbf{U}$, and $u_{ij}$ is a scalar. For example, a $n \times (n-1)$ vertical extended upper triangular matrix can be written explicitly as

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1(n-1)} \\ u_{21} & u_{22} & \cdots & u_{2(n-1)} \\ & u_{32} & \cdots & u_{3(n-1)} \\ & & \ddots & \vdots \\ & & & u_{n(n-1)} \end{bmatrix}. \tag{5.18}$$

**Proposition 5.1 (Extended Cholesky Decomposition)**
Suppose $\mathbf{Q}_n$ is a $n \times n$ positive semidefinite symmetric matrix of rank $k$ $(k < n)$. Then, it can be decomposed as $\mathbf{Q}_n = \mathbf{H}_k \mathbf{H}_k^T$, where $\mathbf{H}_k$ is a $n \times k$ matrix of rank $k$. Furthermore, the decomposition can be written as $\mathbf{Q}_n = \Lambda_k \Lambda_k^T$ with $\Lambda_k$, a $n \times k$ vertical extended upper triangular matrix. The degree-of-freedom of the matrix $\mathbf{Q}_n$ is $nk - \frac{1}{2}k(k-1)$, which is the number of unknowns in $\Lambda_k$.

**Proof** Since $\mathbf{A}$ is a $n \times n$ positive semidefinite symmetric matrix of rank $k$, it can be decomposed by SVD as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{U}^T = \mathbf{U} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_k & & \\ & & & 0 & \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix} \mathbf{U}^T \tag{5.19}$$

where $\mathbf{U}$ is a $n \times n$ orthogonal matrix, and $\Sigma$ is a diagonal matrix with $\sigma_i$ the singular value of $\mathbf{A}$. Consequently we get

$$\mathbf{H}_k = \mathbf{U}^{(1:k)} \begin{bmatrix} \sqrt{\sigma_1} & & \\ & \ddots & \\ & & \sqrt{\sigma_k} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{ku} \\ \mathbf{H}_{kl} \end{bmatrix} \tag{5.20}$$

such that $\mathbf{A} = \mathbf{H}_k \mathbf{H}_k^T$, where $\mathbf{U}^{(1:k)}$ denotes first $k$ columns of $\mathbf{U}$, $\mathbf{H}_{ku}$ denotes upper $(n-k) \times k$ submatrix of $\mathbf{H}_k$, and $\mathbf{H}_{kl}$ denotes lower $k \times k$ submatrix of $\mathbf{H}_k$. By applying RQ-decomposition to $\mathbf{H}_{kl}$, we have $\mathbf{H}_{kl} = \Lambda_{kl} \mathbf{O}_k$, where $\Lambda_{kl}$ is an upper triangular matrix, and $\mathbf{O}_k$ is an orthogonal matrix.

Let us denote $\Lambda_{ku} = \mathbf{H}_{ku} \mathbf{O}_k^T$ and construct a $n \times k$ vertical extended upper triangular matrix

$$\Lambda_k = \begin{bmatrix} \Lambda_{ku} \\ \Lambda_{kl} \end{bmatrix}. \tag{5.21}$$

Then, we have

$$\mathbf{H}_k = \Lambda_k \mathbf{O}_k,\ \mathbf{A} = \mathbf{H}_k \mathbf{H}_k^T = (\Lambda_k \mathbf{O}_k)(\Lambda_k \mathbf{O}_k)^T = \Lambda_k \Lambda_k^T. \tag{5.22}$$

It is easy to verify that the matrix $\mathbf{A}$ has $nk - \frac{1}{2}k(k-1)$ degrees of freedom, which is just the number of unknowns in $\Lambda_k$. ■

The proposition can be taken as an extension of the Cholesky Decomposition to the case of positive semidefinite symmetric matrix, while Cholesky Decomposition can only deal with positive definite symmetric matrix. From the above proposition, we obtain the following result.

**Corollary 5.2**
The matrix $\mathbf{Q}$ recovered from (5.16) is a $4 \times 4$ positive semidefinite symmetric matrix of rank 3. It can be decomposed as $\mathbf{Q} = \mathbf{H}_l \mathbf{H}_l^T$, where $\mathbf{H}_l$ is a $4 \times 3$ rank 3 matrix. The decomposition can be further written as $\mathbf{Q} = \Lambda_3 \Lambda_3^T$ with $\Lambda_3$ a $4 \times 3$ vertical extended upper triangular matrix.

In practice, the matrix $\mathbf{H}_l$ can be easily computed as follows. Suppose the SVD decomposition of $\mathbf{Q}$ is $\mathbf{U}_4 \Sigma_4 \mathbf{U}_4^T$, where $\mathbf{U}_4$ is a $4 \times 4$ orthogonal matrix, $\Sigma_4 = diag(\sigma_1, \sigma_2, \sigma_3, 0)$ is a diagonal matrix with $\sigma_i$ the singular value of $\mathbf{Q}$. Thus, we have

$$\mathbf{H}_l = \mathbf{U}^{(1:3)} \begin{bmatrix} \sqrt{\sigma_1} & & \\ & \sqrt{\sigma_2} & \\ & & \sqrt{\sigma_3} \end{bmatrix}. \tag{5.23}$$

Then, the vertical extended upper triangular matrix $\Lambda_3$ can be constructed from $\mathbf{H}_l$ as in (5.21). From the number of unknowns in $\Lambda_3$, we know that $\mathbf{Q}$ is only defined up to nine degrees of freedom.

In Corollary 5.2, we claim that the symmetric matrix $\mathbf{Q}$ can be decomposed into $\Lambda_3\Lambda_3^T$. In practice, we can simply decompose the matrix into $\mathbf{H}_l\mathbf{H}_l^T$ as shown in (5.23), it is unnecessary to recover $\Lambda_3$ since the upgrading matrix (5.11) is not unique. However, when the data is corrupted by noise, the recovered matrix $\mathbf{Q}$ may be negative definite and the decomposition of (5.23) is impossible. In such case, we suggest the following alternative estimation method.

Let us denote

$$\Lambda_3 = \begin{bmatrix} h1 & h2 & h3 \\ h4 & h5 & h6 \\ & h7 & h8 \\ & & h9 \end{bmatrix} \tag{5.24}$$

and substitute the matrix $\mathbf{Q}$ in (5.15) with $\Lambda_3\Lambda_3^T$. Then, a best estimation of $\Lambda_3$ in (5.24) can be obtained via minimizing the following cost function

$$J_1 = \min_{(\Lambda_3)} \frac{1}{2}\sum_{i=1}^{m}\Big(\mathbf{C}_i^2(1,2)+\mathbf{C}_i^2(1,3)+\mathbf{C}_i^2(2,3)+(\mathbf{C}_i(1,1)-\mathbf{C}_i(2,2))^2\Big). \tag{5.25}$$

The minimization scheme can be solved using any nonlinear optimization techniques, such as gradient descent or Levenberg-Marquardt (LM) algorithm. By introducing the vertical extended upper triangular matrix (5.24), we can reduce three unknowns in matrix $\mathbf{Q}$.

#### Recovering $\mathbf{H}_r$

In this section we recover the right part $\mathbf{H}_r$ of the upgrading matrix (5.11). From quasi-perspective equation (5.8), we have

$$\mathbf{x}_{ij} = (\mu_i\mathbf{P}_i^{(1:3)})(\ell_j\bar{\mathbf{X}}_j) + (\mu_i\mathbf{P}_i^{(4)})\ell_j. \tag{5.26}$$

For all features in the $i$-th frame, we take a summation of their coordinates as below.

$$\sum_{j=1}^{n}\mathbf{x}_{ij} = \mu_i\mathbf{P}_i^{(1:3)}\sum_{j=1}^{n}(\ell_j\bar{\mathbf{X}}_j) + \mu_i\mathbf{P}_i^{(4)}\sum_{j=1}^{n}\ell_j \tag{5.27}$$

where $\mu_i\mathbf{P}_i^{(1:3)}$ can be recovered from $\hat{\mathbf{M}}_i\mathbf{H}_l$, $\mu_i\mathbf{P}_i^{(4)} = \hat{\mathbf{M}}_i\mathbf{H}_r$. Since the world coordinate system can be chosen freely, we may set the origin of the world system at the gravity center of the scaled space points as

$$\sum_{j=1}^{n}(\ell_j\bar{\mathbf{X}}_j) = 0. \tag{5.28}$$

On the other hand, we may simply set

$$\sum_{j=1}^{n} \ell_j = 1 \tag{5.29}$$

since the reconstruction is defined up to a global scalar. Thus, Equation (5.27) is simplified to

$$\hat{\mathbf{M}}_i \mathbf{H}_r = \sum_{j=1}^{n} \mathbf{x}_{ij} = \begin{bmatrix} \sum_j u_{ij} \\ \sum_j v_{ij} \\ n \end{bmatrix} \tag{5.30}$$

which provides three linear constraints on the four unknowns of $\mathbf{H}_r$. Therefore, we obtain $3m$ equations from the sequence and $\mathbf{H}_r$ can be recovered via linear least squares.

From the above analysis, we note that the solution of $\mathbf{H}_r$ is not unique as it is dependant on the selection of the world origin $\sum_{j=1}^{n}(\ell_j \bar{\mathbf{X}}_j)$ and the global scalar $\sum_{j=1}^{n} \ell_j$. Actually, $\mathbf{H}_r$ may be set freely as shown in the following proposition.

**Proposition 5.3**
Suppose $\mathbf{H}_l$ in (5.11) is already recovered. Let us choose an arbitrary four dimensional vector $\tilde{\mathbf{H}}_r$ that is independent of the three columns of $\mathbf{H}_l$, and construct a matrix as

$$\tilde{\mathbf{H}} = [\mathbf{H}_l | \tilde{\mathbf{H}}_r].$$

Then, $\tilde{\mathbf{H}}$ must be a valid upgrading matrix. i.e., $\tilde{\mathbf{M}} = \hat{\mathbf{M}}\tilde{\mathbf{H}}$ is a valid Euclidean motion matrix, and $\tilde{\mathbf{S}} = \tilde{\mathbf{H}}^{-1}\hat{\mathbf{S}}$ corresponds to a valid Euclidean shape matrix.

**Proof** Suppose the correct transformation matrix is $\mathbf{H} = [\mathbf{H}_l | \mathbf{H}_r]$, then, from

$$\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}} = \begin{bmatrix} \ell_1 \bar{\mathbf{X}}_1, \cdots, \ell_n \bar{\mathbf{X}}_n \\ \ell_1, \quad \cdots, \quad \ell_n \end{bmatrix} \tag{5.31}$$

we obtain a correct Euclidean structure $[\bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n]$ of the object under a certain world coordinate frame by dehomogenizing the shape matrix $\mathbf{S}$. The arbitrary constructed matrix $\tilde{\mathbf{H}} = [\mathbf{H}_l | \tilde{\mathbf{H}}_r]$ and the correct matrix $\mathbf{H}$ are defined up to a $4 \times 4$ invertible matrix $\mathbf{G}$ as

$$\mathbf{H} = \tilde{\mathbf{H}}\mathbf{G}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{g} \\ \mathbf{0}^T & s \end{bmatrix} \tag{5.32}$$

where $\mathbf{I}_3$ is a $3 \times 3$ identity matrix, $\mathbf{g}$ is a 3-vector, $\mathbf{0}$ is a zero 3-vector, and $s$ is a nonzero scalar. Under the transformation matrix $\tilde{\mathbf{H}}$, the motion $\hat{\mathbf{M}}$ and shape $\hat{\mathbf{S}}$ are transformed to

$$\tilde{\mathbf{M}} = \hat{\mathbf{M}}\tilde{\mathbf{H}} = \hat{\mathbf{M}}\mathbf{H}\mathbf{G}^{-1} = \mathbf{M}\begin{bmatrix} \mathbf{I}_3 & -\mathbf{g}/s \\ \mathbf{0}^T & 1/s \end{bmatrix} \tag{5.33}$$

$$\begin{aligned} \tilde{\mathbf{S}} &= \tilde{\mathbf{H}}^{-1}\hat{\mathbf{S}} = (\mathbf{H}\mathbf{G}^{-1})^{-1}\hat{\mathbf{S}} = \mathbf{G}(\mathbf{H}^{-1}\hat{\mathbf{S}}) \\ &= s\begin{bmatrix} \ell_1(\bar{\mathbf{X}}_1 + \mathbf{g})/s & \cdots & \ell_n(\bar{\mathbf{X}}_n + \mathbf{g})/s \\ \ell_1 & \cdots & \ell_n \end{bmatrix}. \end{aligned} \tag{5.34}$$

As shown in (5.34), the new shape $\tilde{\mathbf{S}}$ is actually the original structure that undergoes a translation $\mathbf{g}$ and a scale $1/s$, which does not change the Euclidean structure. From (5.33), we have $\tilde{\mathbf{M}}^{(1:3)} = \mathbf{M}^{(1:3)}$, which indicates that the first-three-columns of the new motion matrix (rotation term) do not change. While the last column, which corresponds to translation factor, is modified in accordance with the translation and scale changes of the structure.

Therefore, the constructed matrix $\tilde{\mathbf{H}}$ is a valid transformation matrix that can upgrade the factorization from projective space to the Euclidean space. ■

According to Proposition 5.3, the value of $\mathbf{H}_r$ can be set randomly as any 4-vector that is independent of $\mathbf{H}_l$. A practical selection method may be as follows.

Suppose the SVD decomposition of $\mathbf{H}_l$ is

$$\mathbf{H}_l = \mathbf{U}_{4\times4}\Sigma_{4\times3}\mathbf{V}_{3\times3}^T = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4]\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{bmatrix}[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^T \tag{5.35}$$

where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices, and $\Sigma$ is a diagonal matrix composed of the three singular values. Let us choose an arbitrary value $\sigma_r$ between the biggest and the smallest singular values of $\mathbf{H}_l$, then we may set

$$\mathbf{H}_r = \sigma_r\mathbf{u}_4, \quad \mathbf{H} = [\mathbf{H}_l, \mathbf{H}_r]. \tag{5.36}$$

Such a kind of construction guarantees that $\mathbf{H}$ is invertible and has the same condition number as $\mathbf{H}_l$, so that we can obtain a good precision in computing the inverse $\mathbf{H}^{-1}$.

After recovering the Euclidean motion and shape matrices, the intrinsic parameters and pose of the camera associated with each frame can be easily computed as follows.

$$\mu_i = \left\|\mathbf{M}_{i(3)}^{(1:3)}\right\| \tag{5.37}$$

$$f_i = \frac{1}{\mu_i}\left\|\mathbf{M}_{i(1)}^{(1:3)}\right\| = \frac{1}{\mu_i}\left\|\mathbf{M}_{i(2)}^{(1:3)}\right\| \tag{5.38}$$

$$\mathbf{R}_i = \frac{1}{\mu_i}\mathbf{K}_i^{-1}\mathbf{M}_i^{(1:3)},\ \mathbf{T}_i = \frac{1}{\mu_i}\mathbf{K}_i^{-1}\mathbf{M}_i^{(4)} \tag{5.39}$$

where $\mathbf{M}_{i(t)}^{(1:3)}$ denotes the $t$-th row of $\mathbf{M}_i^{(1:3)}$. The result is obtained under quasi-perspective assumption, which is a close approximation to the general perspective projection. The solution may be further optimized to perspective projection by minimizing the image reprojection residuals.

$$J_2 = \min_{(\mathbf{K}_i,\mathbf{R}_i,\mathbf{T}_i,\mu_i,\mathbf{X}_j)} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|\bar{\mathbf{x}}_{ij} - \hat{\mathbf{x}}_{ij}\right|^2 \tag{5.40}$$

where $\hat{\mathbf{x}}_{ij}$ denotes the reprojected image point computed via perspective projection (5.1). The minimization process is termed as bundle adjustment, which is usually solved via Levenberg-Marquardt (LM) iterations [32].

### 5.3.2 Algorithm outline

Given the tracking matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{3m\times n}$ across a sequence with small camera movements. The implementation of the quasi-perspective rigid factorization algorithm is summarized as follows.

1. Balance the tracking matrix via point-wise and image-wise rescalings, as in [71], to improve its numerical stability;
2. Perform rank-4 SVD decomposition on the tracking matrix to obtain a set of solution of $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$;
3. Compute the left part of upgrading matrix $\mathbf{H}_l$ according to (5.23), or (5.25) for negative definite matrix $\mathbf{Q}$;
4. Construct $\mathbf{H}_r$ and $\mathbf{H}$ according to (5.36);
5. Recover the Euclidean motion matrix $\mathbf{M} = \hat{\mathbf{M}}\mathbf{H}$ and shape matrix $\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}}$;
6. Estimate the camera parameters and pose from (5.37) to (5.39);
7. Optimize the solution via bundle adjustment (5.40).

In the above analysis, as well as in other factorization algorithms, we usually assume one-parameter-camera model as in (5.15) so that we can use this constraint to recover the upgrading matrix $\mathbf{H}$. In real applications, we may take

the solution as an initial value and optimize the camera parameters via Kruppa constraints that arise from pairwise images [88].

The essence of quasi-perspective factorization (5.10) is to find a rank-4 approximation $\mathbf{MS}$ of the tracking matrix by minimizing the Frobenius norm $\|\tilde{\mathbf{W}} - \mathbf{MS}\|_F^2$. Most studies adopt a SVD decomposition of $\tilde{\mathbf{W}}$ and truncate it to the desired rank. However, when the tracking matrix is not complete, such as when some features are missing in some frames due to occlusions, it is hard to perform SVD decomposition. In case of missing data, we adopt the power factorization algorithm [33][84] to obtain a least squares solution of $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$. The solution is then upgraded the solution to the Euclidean space according to the proposed scheme.

## 5.4 Quasi-Perspective Nonrigid Factorization

### 5.4.1 Problem formulation

For nonrigid factorization, we follow Bregler's assumption (5.5) to represent a nonrigid shape by weighted combination of $k$ shape bases. Under quasi-perspective projection, the structure is expressed in homogeneous form with nonzero scalars. Let us denote the scale-weighted nonrigid structure associated with the $i$-th frame and the $l$-th scale weighted shape basis as

$$\bar{\mathbf{S}}_i = [\ell_1\bar{\mathbf{X}}_1, \cdots, \ell_n\bar{\mathbf{X}}_n],\ \mathbf{B}_l = [\ell_1\bar{\mathbf{X}}_1^l, \cdots, \ell_n\bar{\mathbf{X}}_n^l]. \tag{5.41}$$

Then from (5.5), we have

$$\bar{\mathbf{X}}_i = \sum_{l=1}^{k} \omega_{il}\bar{\mathbf{X}}_i^l,\ \forall\, i = 1, \cdots, n. \tag{5.42}$$

Let us multiply a weight scale $\ell_i$ on both sides as

$$\ell_i\bar{\mathbf{X}}_i = \ell_i \sum_{l=1}^{k} \omega_{il}\bar{\mathbf{X}}_i^l = \sum_{l=1}^{k} \omega_{il}(\ell_i\bar{\mathbf{X}}_i^l) \tag{5.43}$$

then, we can immediately have the following result.

$$\mathbf{S}_i = \begin{bmatrix} \bar{\mathbf{S}}_i \\ \ell^T \end{bmatrix} = \begin{bmatrix} \sum_{l=1}^{k} \omega_{il}\mathbf{B}_l \\ \ell^T \end{bmatrix}. \tag{5.44}$$

We call (5.44) extended Bregler's assumption in homogeneous case. Under this extension, the quasi-perspective projection of the $i$-th frame can be formulated as

$$\tilde{\mathbf{W}}_i = (\mu_i\mathbf{P}_i)\mathbf{S}_i = [\mu_i\mathbf{P}_i^{(1:3)}, \mu_i\mathbf{P}_i^{(4)}] \begin{bmatrix} \sum_{l=1}^{k} \omega_{il}\mathbf{B}_l \\ \ell^T \end{bmatrix} \tag{5.45}$$

$$= [\omega_{i1}\mu_i \mathbf{P}_i^{(1:3)}, \cdots, \omega_{ik}\mu_i \mathbf{P}_i^{(1:3)}, \mu_i \mathbf{P}_i^{(4)}] \begin{bmatrix} \mathbf{B}_1 \\ \cdots \\ \mathbf{B}_k \\ \ell^T \end{bmatrix}.$$

Thus, the nonrigid factorization under quasi-perspective projection can be expressed as

$$\tilde{\mathbf{W}}_{3m\times n} = \begin{bmatrix} \omega_{11}\mu_1 \mathbf{P}_1^{(1:3)} & \cdots & \omega_{1k}\mu_1 \mathbf{P}_1^{(1:3)} & \mu_1 \mathbf{P}_1^{(4)} \\ \vdots & \ddots & \vdots & \vdots \\ \omega_{m1}\mu_m \mathbf{P}_m^{(1:3)} & \cdots & \omega_{mk}\mu_m \mathbf{P}_m^{(1:3)} & \mu_m \mathbf{P}_m^{(4)} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \\ \ell^T \end{bmatrix} \tag{5.46}$$

or represented concisely in matrix form as

$$\tilde{\mathbf{W}}_{3m\times n} = \mathbf{M}_{3m\times(3k+1)} \mathbf{B}_{(3k+1)\times n}. \tag{5.47}$$

The factorization expression is similar to (5.7). However, the difficulties in estimating the projective depths are circumvented.

### 5.4.2 Euclidean upgrading matrix

The rank of the tracking matrix is at most $3k+1$, and the factorization is defined up to a transformation matrix $\mathbf{H} \in \mathbb{R}^{(3k+1)\times 3k+1)}$. Suppose the SVD factorization of a tracking matrix with rank constraint is $\tilde{\mathbf{W}} = \hat{\mathbf{M}}\hat{\mathbf{B}}$. Similar to the rigid case, we adopt the metric constraint to compute the upgrading matrix. Let us denote the matrix into $k+1$ parts as

$$\mathbf{H} = [\mathbf{H}_1, \cdots, \mathbf{H}_k | \mathbf{H}_r] \tag{5.48}$$

where $\mathbf{H}_l \in \mathbb{R}^{(3k+1)\times 3}(l = 1, \cdots, k)$ denotes the $l$-th triple columns of $\mathbf{H}$, and $\mathbf{H}_r$ denotes the last column of $\mathbf{H}$. Then, we have

$$\hat{\mathbf{M}}_i \mathbf{H}_l = \omega_{il}\mu_i \mathbf{P}_i^{(1:3)} = \omega_{il}\mu_i \mathbf{K}_i \mathbf{R}_i \tag{5.49}$$

$$\hat{\mathbf{M}}_i \mathbf{H}_r = \mu_i \mathbf{P}_i^{(4)} = \mu_i \mathbf{K}_i \mathbf{t}_i. \tag{5.50}$$

Similar to (5.15) in the rigid case, Let us denote $\mathbf{C}_{ii'} = \hat{\mathbf{M}}_i \mathbf{Q}_l \hat{\mathbf{M}}_{i'}^T$ with $\mathbf{Q}_l = \mathbf{H}_l \mathbf{H}_l^T$, we obtain

$$\begin{aligned} \mathbf{C}_{ii'} &= \hat{\mathbf{M}}_i \mathbf{Q}_l \hat{\mathbf{M}}_{i'}^T = (\omega_{il}\mu_i \mathbf{K}_i \mathbf{R}_i)(\omega_{i'l}\mu_{i'} \mathbf{K}_{i'} \mathbf{R}_{i'})^T \\ &= \omega_{il}\omega_{i'l}\mu_i\mu_{i'} \mathbf{K}_i (\mathbf{R}_i \mathbf{R}_{i'}) \mathbf{K}_{i'}^T \end{aligned} \tag{5.51}$$

where $i$ and $i'$ $(=1,\cdots,m)$ correspond to different frame numbers, $l=1,\cdots,k$ corresponds to different shape bases. Assuming a simplified camera model with only one parameter as $\mathbf{K}_i = diag(f_i, f_i, 1)$, we have

$$\mathbf{C}_{ii} = \hat{\mathbf{M}}_i \mathbf{Q}_l \hat{\mathbf{M}}_i^T = \omega_{il}^2 \mu_i^2 \begin{bmatrix} f_i^2 & & \\ & f_i^2 & \\ & & 1 \end{bmatrix} \tag{5.52}$$

from which we obtain following four constraints.

$$\begin{cases} f_1(\mathbf{Q}_l) = \mathbf{C}_{ii}(1,2) = 0 \\ f_2(\mathbf{Q}_l) = \mathbf{C}_{ii}(1,3) = 0 \\ f_3(\mathbf{Q}_l) = \mathbf{C}_{ii}(2,3) = 0 \\ f_4(\mathbf{Q}_l) = \mathbf{C}_{ii}(1,1) - \mathbf{C}_{ii}(2,2) = 0 \end{cases}. \tag{5.53}$$

The above constraints are similar to (5.16) in the rigid case. However, the matrix $\mathbf{Q}_l$ in (5.52) is a $(3k+1)\times(3k+1)$ symmetric matrix. According to Proposition 5.1, $\mathbf{Q}_l$ has $9k$ degrees of freedom, since it can be decomposed into the product of $(3k+1)\times 3$ vertical extended upper triangular matrix. Given $m$ frames, we have $4m$ linear constraints on $\mathbf{Q}_l$. It appears that if we have enough features and frames, the matrix $\mathbf{Q}_l$ can be solved linearly by stacking all constraints in (5.53). Unfortunately, only the rotation constraints may be insufficient when an object deforms at varying speed, since most of these constraints are redundant. Xiao *et al.* [96] proposed a basis constraint to solve this ambiguity.

The main idea of basis constraint is to select $k$ frames that include independent shapes and treat them as a set of bases. Suppose the first $k$ frames are independent of each other, then their corresponding weighting coefficients can be set as

$$\omega_{il} = \begin{cases} 1 & \text{if } i,l = 1,...,k \text{ and } i = l \\ 0 & \text{if } i,l = 1,...,k \text{ and } i \neq l \end{cases}. \tag{5.54}$$

From (5.51) we obtain following basis constraint.

$$\mathbf{C}_{ii'} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \text{if } i = 1,...,k,\ i' = 1,...,m, \text{ and } i \neq l. \tag{5.55}$$

Given $m$ images, (5.55) can provide $9m(k-1)$ linear constraints to the matrix $\mathbf{Q}_l$ (some of the constraints are redundant since $\mathbf{Q}_l$ is symmetric). By combining the rotation constraint (5.53) and the basis constraint (5.55) together, the matrix $\mathbf{Q}_l$ can be computed linearly. Then, $\mathbf{H}_l, l=1,\cdots,k$ may be decomposed from $\mathbf{Q}_l$ according to following Proposition.

**Proposition 5.4**
The matrix $\mathbf{Q}_l$ is a $(3k+1)\times(3k+1)$ positive semidefinite symmetric matrix of rank 3. It can be decomposed as $\mathbf{Q}=\mathbf{H}_l\mathbf{H}_l^T$, where $\mathbf{H}_l$ is a $(3k+1)\times 3$ rank-3 matrix. The decomposition can be further written as $\mathbf{Q}=\Lambda_3\Lambda_3^T$ with $\Lambda_3$ being a $(3k+1)\times 3$ vertical extended upper triangular matrix.

The result can be easily derived from Proposition 5.1. It is easy to verify that the Proposition 5.3 is still valid for the nonrigid case. Thus, the vector $\mathbf{H}_r$ in (5.48) can be set as an arbitrary $(3k+1)$-vector that is independent of all columns in $\{\mathbf{H}_l\}$, $l=1,\cdots,k$. After recovering the Euclidean upgrading matrix, the camera parameters, motions, shape bases, and weighing coefficients can be easily decomposed from the upgraded motion and shape matrices.

$$\mathbf{M}=\hat{\mathbf{M}}\mathbf{H},\ \mathbf{B}=\mathbf{H}^{-1}\hat{\mathbf{B}}. \tag{5.56}$$

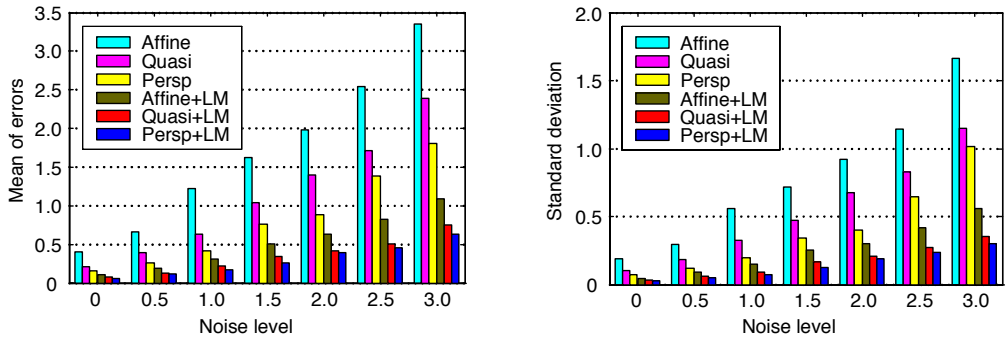
## 5.5 Evaluations on Synthetic Data

In this section, the proposed rigid and nonrigid factorization algorithms are evaluated using synthetic data in terms of accuracy and efficiency.

### 5.5.1 Evaluations of rigid factorization

During simulations, we randomly generated 200 points within a cube of $20\times 20\times 20$ in space and simulated 10 images from these points by perspective projection. The image size is set at $800\times 800$. The camera parameters are set as follows: the focal lengths are set randomly between 900 and 1100, the principal point is set at the image center, and the skew is zero. The rotation angles are set randomly between $\pm 5^\circ$. The X and Y positions of the cameras are set randomly between $\pm 15$, while the Z positions are set evenly from 200 to 220. The imaging condition is very close to quasi-perspective assumption.

We add Gaussian white noise to the initially generated 10 images (i.e., all tracked features in the tracking matrix), and vary the noise level (the standard deviation of the Gaussian noise) from 0 to 3 units with steps of 0.5. At each noise level, we reconstruct the 3D structure of the object which is defined up to a similarity transformation with the ground truth. We register the reconstructed model with the ground truth and calculate the reconstruction error, which is defined as mean point-wise distance between the reconstructed structure and the




(a) (b)

Figure 5.1: Evaluation on the accuracy of rigid factorization. (a) The mean of reconstruction errors by different algorithms at different noise levels; and (b) the corresponding standard deviation of reconstruction errors.

ground truth. The mean and standard deviation of the errors on 100 independent tests are shown in Figure 5.1.

The proposed algorithm (Quasi) is compared with [62] under the affine assumption (Affine) and [31] under the perspective projection (Persp). We also perform a bundle adjustment optimization scheme through Levenberg-Marquardt (LM) algorithm to upgrade the solution to the perspective projection. It is evident that the proposed method performs much better than that of affine, and the optimized solution (Quasi+LM) is very close to the perspective projection with optimization (Persp+LM).

The proposed model is based on the assumption of large relative camera-to-object distance and small camera rotations. We studied the influence of these two factors to different camera models. In the first case, we vary the relative distance from 4 to 18 in steps of 2. At each relative distance, we generate 20 images with the following parameters. The rotation angles are confined between $\pm 5^\circ$, the X and Y positions of the camera are set randomly between $\pm 15$. We recover the structure and compute the reconstruction error for each group of images. The mean reconstruction error by different methods is shown in Figure 5.2. In the second case, we increase the rotation angles to the range of $\pm 20^\circ$, and retain other camera parameters the same as in the first case. The mean reconstruction error is given in Figure 5.2. These results are evaluated on 100 independence tests with 1-unit Gaussian noise.

Based on experimental evidence we have the following Conclusion. (i) The error by the quasi-perspective projection is consistently less than that by the affine, especially at small relative distances; (ii) both reconstruction errors by the affine and the quasi-perspective projection increase greatly when the relative distance

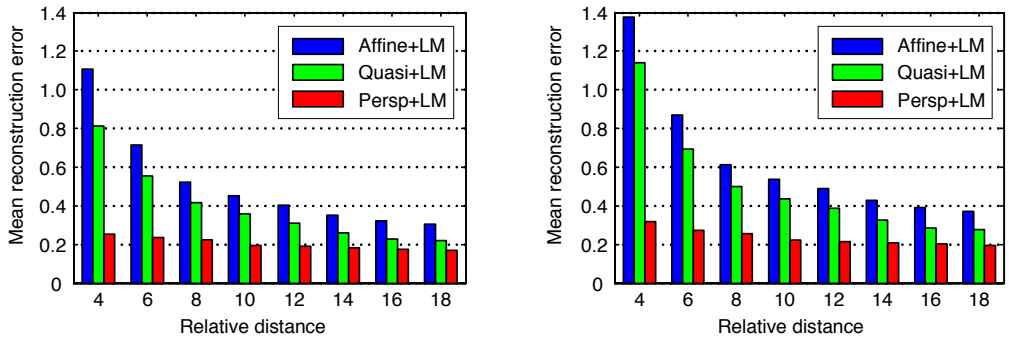



Figure 5.2: The mean reconstruction error of different projection models with respect to varying relative distance. (a) The rotation angle of the camera varies in a range of $\pm 5°$; and (b) the rotation angle varies in a range of $\pm 20°$.

Table 5.1: The average computation time of different algorithms (unit: second).

| Frame number | | 5 | 10 | 50 | 100 | 150 | 200 |
|---|---|---|---|---|---|---|---|
| Time (s) | Affine | 0.015 | 0.015 | 0.031 | 0.097 | 0.156 | 0.219 |
| | Quasi | 0.015 | 0.016 | 0.047 | 0.156 | 0.297 | 0.531 |
| | Persp | 0.281 | 0.547 | 3.250 | 6.828 | 10.58 | 15.25 |

is less than 6, since both models are based on large distance assumption; (iii) the error at each relative distance increases with the rotation angles, especially at small relative distances, since the projective depths are related to rotation angles; and (4) theoretically, the relative distance and rotation angles have no influence on the result of full perspective projection. However, we see that the error by the perspective projection also increases slightly with the increase in rotation angles and the decrease in relative distance. This is because we estimate the projective depths iteratively starting with an affine assumption [31]. The iteration easily gets stuck to local minima due to bad initialization.

We compared the computation time of different factorization algorithms without LM optimization. The program was implemented with Matlab 6.5 on a PC with Intel Pentium®4 3.6GHz CPU. In this test, we use all 200 feature points and vary the frame number from 5 to 200, so as to generate different data size. The actual computation time for different data sets are tabulated in Table 5.1, where the computation time for the perspective projection is taken for 10 iterations (it usually takes about 30 iterations to compute the projective depths in perspective factorization). Clearly, the computation time required by the quasi-perspective factorization is close to that of the affine assumption,

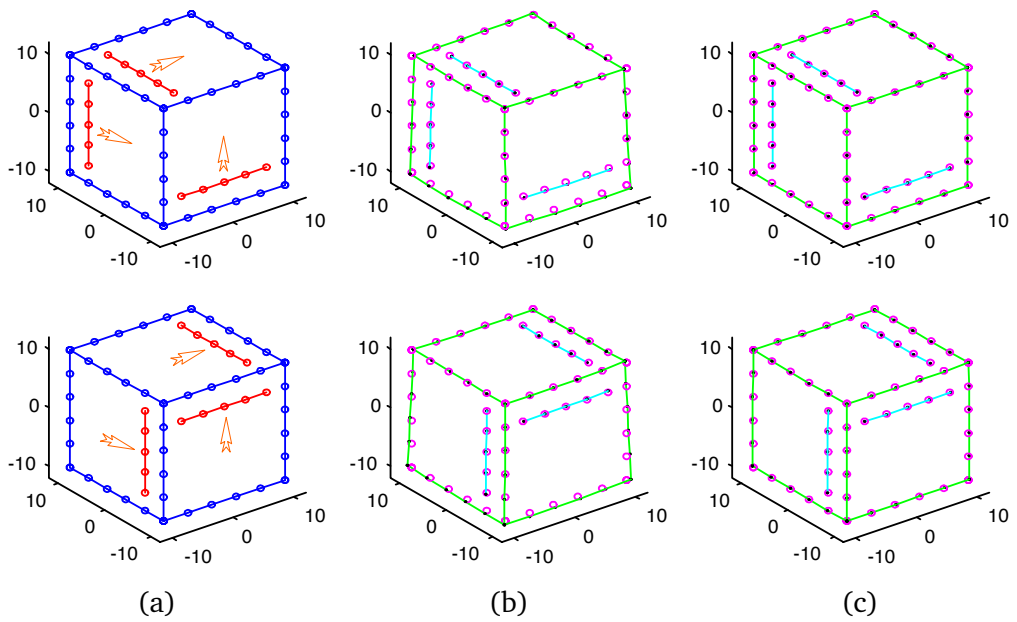


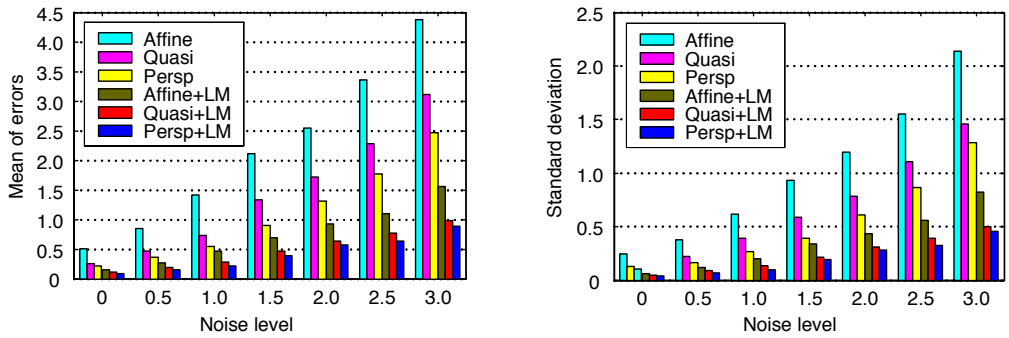

Figure 5.3: Reconstruction results of nonrigid factorization. (a) Two synthetic cubes with moving points in space; (b) the quasi-perspective factorization result of the two frames (in black dots) superimposed with the ground truth (in pink circles); and (c) the final structures after optimization.

whereas the perspective factorization is computationally more intensive than other methods.

### 5.5.2 Evaluations of nonrigid factorization

In this test, we generated a synthetic cube with six evenly distributed points on each visible edge. There are three sets of moving points on adjacent surfaces of the cube that move at a constant speed as shown in Figure 5.3, each moving set is composed of five points. The cube with moving points can be taken as a nonrigid object with two shape bases. We generated ten frames with the same camera parameters as in the first test for the rigid case. We reconstructed the structure associated with each frame by the proposed method. The result is shown in Figure 5.3, from which we can see that the structure after optimization is visually the same as the ground truth, while the result before optimization is a little bit deformed due to perspective effect.

We compared our method with the nonrigid factorization under the affine assumption [95] and that under the perspective projection [96]. The mean and standard deviation of the reconstruction errors with respect to different noise




(a) (b)

Figure 5.4: Evaluation of nonrigid factorization. The mean (a) and standard deviation (b) of the reconstruction errors by different algorithms at different noise levels.

levels are shown in Figure 5.4. It is obvious that the proposed method performs significantly better than that under the affine camera model.

## 5.6 Evaluations on Real Image Sequences

We tested our proposed method on many real sequences, and we report two results in the following.

### 5.6.1 Test on fountain base sequence

There are seven images in the fountain base sequence, which were taken at the Sculpture Park of Windsor by Canon Powershot a G3 camera. The image resolution is $1024 \times 768$. In order to ensure large overlap of the object to be reconstructed, the camera undergoes small movement during image acquisition, hence the quasi-perspective assumption is satisfied for the sequences. We established the initial correspondences by utilizing the technique outlined in [80] and eliminated outliers iteratively as in [59]. Totally 4218 reliable features were tracked across the sequence as shown in Figure 5.5. We recovered the 3D structure of the object and camera motions by utilizing the proposed algorithm, as well as some previous methods. Figure 5.5 shows the reconstructed VRML model and the corresponding wireframes from different viewpoints. The model looks realistic and most details are correctly recovered by the proposed method.

In order to compare the algorithms quantitatively, we reproject the reconstructed 3D structure back to the images and calculate the reprojection errors, i.e. distances between the detected and the reprojected image points. Figure 5.6

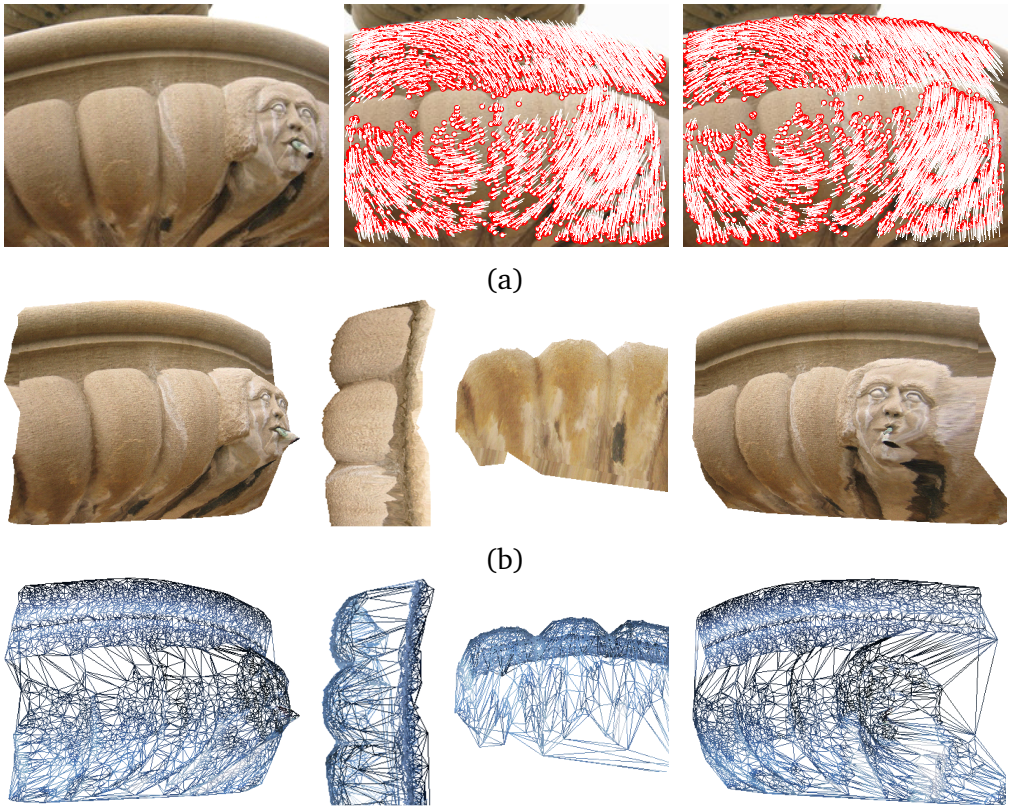


Figure 5.5: Reconstruction results of fountain base sequence. (a) Three images from the sequence, where the tracked features with relative disparities are overlaid to the second and the third images; (b) the reconstructed VRML model of the scene shown from different viewpoints with texture mapping; and (c) the corresponding triangulated wireframe of the reconstructed model.

shows the histogram distributions of the errors using nine bins. The corresponding mean ('Mean') and standard deviation ('STD') of the errors are listed in Table 5.2. It can be seen that the reprojection error by the proposed model is much smaller than that under the affine assumption.

### 5.6.2 Test on Franck sequence

The Franck face sequence was downloaded from the European working group on face and gesture recognition. We selected 60 frames with various facial expressions for the test. The image resolution is $720 \times 576$, and there are 68 tracked feature across the sequence. Figure 5.7 shows the reconstructed models of

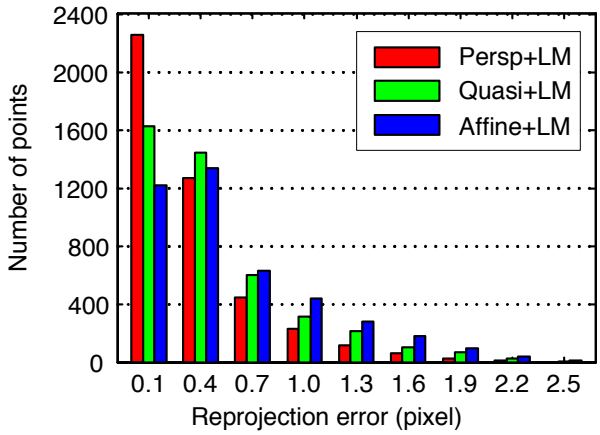



Figure 5.6: The histogram distributions of the reprojection errors by different algorithms in the test of fountain base sequence.

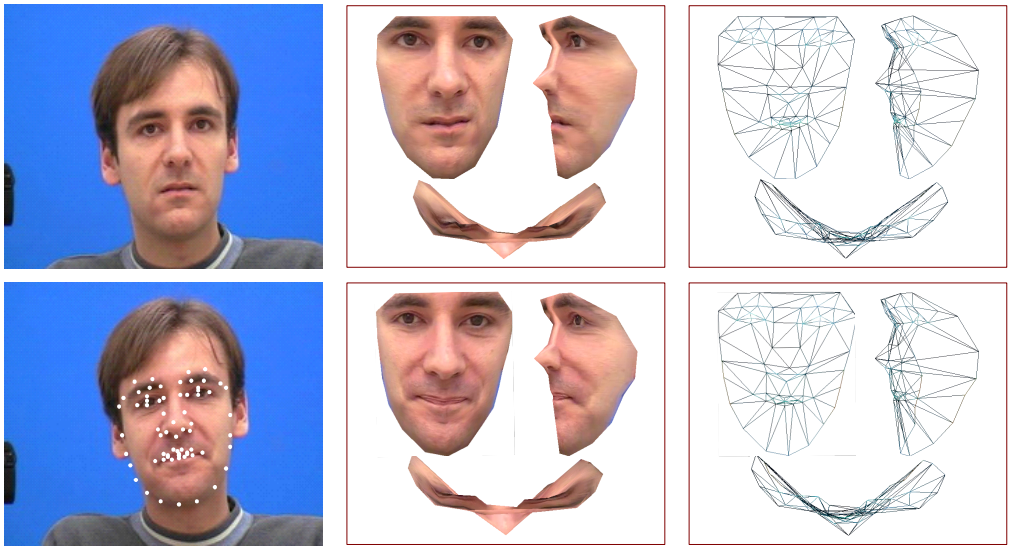
Table 5.2: Camera parameters of the first four frames and reprojection errors in real sequence test.

| Method | $f_1$ | $f_2$ | $f_3$ | $f_4$ | Mean | STD | $\mathbf{E}_{rep}$ |
|---|---|---|---|---|---|---|---|
| Quasi+LM | 2140.5 | 2143.6 | 2139.4 | 2142.8 | 0.418 | 0.285 | 2.473 |
| Affine+LM | 2153.4 | 2155.7 | 2151.2 | 2153.1 | 0.629 | 0.439 | 3.189 |
| Persp+LM | 2131.7 | 2135.3 | 2131.2 | 2134.5 | 0.240 | 0.168 | 1.962 |

two frames utilizing the proposed nonrigid factorization method. Different facial expressions are correctly recovered. As a comparison, the relative reprojection error $\mathbf{E}_{rep}$ generated from different methods are listed in Table 5.2. All tests illustrate that the accuracy by the proposed method is fairly close to that of full perspective projection, and considerably better than the affine assumption.

## 5.7 Conclusion

In this chapter, we have proposed a quasi-perspective factorization algorithm for both rigid and nonrigid objects and elaborated the computation details of the Euclidean upgrading matrix. The proposed method avoids the difficulties associated with computation of the projective depths under the perspective factorization. It is computationally simple with better accuracy than the affine approximation. The proposed model is suitable for structure and motion factorization of short sequences with small camera motions. Experiments demonstrates improvements of our algorithm over the existing techniques. For a long image sequences taken



(a) (b) (c)

Figure 5.7: Reconstruction of different facial expressions in Franck face sequence. (a) Two frames from the sequence with the 68 tracked features overlaid to the lower frame; (b) the front, side, and top views of the reconstructed VRML models with texture mapping; and (c) the corresponding triangulated wireframe of the reconstructed models.

around an object, we can simply divide the sequence into several subsequences with small movements, then register and merge the results of every subsequences to reconstruct the structure of the entire object.

# Chapter 6

# Spatial-and-Temporal-Weighted Factorization

In this chapter, a spatial-and-temporal-weighted factorization algorithm is proposed to handle significant noise contained in the tracking data. The main novelties and contributions of the proposed algorithm include: (i) The image reprojection residual of a feature point is demonstrated to be generally proportional to the error magnitude associated with the image point; (ii) the error distributions are estimated from a new perspective, the reprojection residuals. The image errors are modeled both spatially and temporally to cope with different kinds of uncertainties. Previous studies have considered only the spatial information; and (iii) based on the estimated error distributions, a spatial-and-temporal-weighted factorization algorithm is proposed to improve the overall accuracy and robustness of traditional approaches. Unlike existing approaches, the proposed technique does not require prior information of image measurement and is easy to implement. Extensive experiments on synthetic data and real images validate the proposed method.

## 6.1 Introduction

Most factorization methods usually assume error-free tracking data. The performance will degenerate in the presence of outliers or significant noise. In recent years, the problem of robust factorization has been a concern, and some practical methods have been proposed to handle noisy and erroneous data [100].

Aguitar and Moura [3] proposed a scalar-weighted SVD algorithm by minimizing the weighted square errors. Anandan and Irani [7] proposed a covariance-weighted factorization to factorize noisy correspondences with a high degree of

directional uncertainty. Gruber and Weiss [29] formulated the problem as a factor analysis and derived an expectation maximization (EM) algorithm to incorporate prior knowledge and enhance the robustness to missing data and uncertainties. Zelnik-Manor *et al.* [103] defined a new type of motion consistency based on temporal consistency, and applied it to multi-body factorization with directional uncertainty. Zaharescu and Horaud [100] introduced a Gaussian/uniform mixture model and incorporated it with the EM algorithm. Buchanan and Fitzgibbon [12] presented a comprehensive comparison on a number of factorization algorithms. Their study strongly supports second order nonlinear optimization strategy.

Previous methods of robust factorization are either based on isotropic or directional assumption of the feature uncertainties, which may be recovered during the stage of feature detection and matching. Unfortunately, this information is hard to retrieve in practice, and unavailable in many situations. By analyzing the image reprojection residuals, we find that the errors associated with image features are closely related to the reprojection residuals. Based on this observation, the image uncertainties are modeled spatially and temporally, and a spatial-and-temporal-weighted factorization algorithm is proposed in this chapter. Experiments show that the proposed method improves the accuracy and robustness of the classical affine factorization algorithm. A preliminary work on this research was published in IEEE Trans. on Circuits and Systems for Video Technology [89].

The remainder of this chapter is organized as follows. Some preliminary background on structure and motion factorization is offered in Section 6.2. The proposed spatial-and-temporal-weighted factorization is elaborated in Section 6.3. Experimental evaluations and comparisons on synthetic and real images are described in Section 6.4 and 6.5, respectively. Finally, a short conclusion is drawn in Section 6.6.

## 6.2 Background on Structure and Motion Factorization

In this section, a brief introduction to structure and motion factorization is presented to facilitate understanding of the discussed problem in this chapter.

Under a simplified affine camera model, if the imaged points in each image are registered to the centroid and relative image coordinates are employed, the

imaging process of an image sequence can be modeled as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{f1} & \cdots & \bar{\mathbf{x}}_{fn} \end{bmatrix}}_{\mathbf{W}_{m\times n}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_f \end{bmatrix}}_{\mathbf{M}_{m\times 3}} \underbrace{\begin{bmatrix} \bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n \end{bmatrix}}_{\bar{\mathbf{S}}_{3\times n}} \tag{6.1}$$

where $f$ is the frame number. Please note that, for the convenience of discussion, we use $f$ to denote the frame number in this chapter, and let $m = 2f$. Equation (6.1) is usually written concisely as:

$$\mathbf{W}_{m\times n} = \mathbf{M}_{m\times 3}\bar{\mathbf{S}}_{3\times n} \tag{6.2}$$

where the matrix

$$\mathbf{W} = \{\bar{\mathbf{x}}_{ij}\} = \begin{bmatrix} \begin{bmatrix} u_{11} \\ v_{11} \end{bmatrix} & \cdots & \begin{bmatrix} u_{1n} \\ v_{1n} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} u_{f1} \\ v_{f1} \end{bmatrix} & \cdots & \begin{bmatrix} u_{fn} \\ v_{fn} \end{bmatrix} \end{bmatrix}_{m\times n} \tag{6.3}$$

is called the tracking matrix or measurement matrix. Suppose the tracking matrix $\mathbf{W}$ is available, the purpose of structure and motion factorization is to recover the motion matrix $\mathbf{M}$ and the shape matrix $\mathbf{S}$ as shown in (6.1).

The classical factorization algorithms utilize all image measurements and give them equal consideration. Thus, the problem is equivalent to minimizing the following error function:

$$Err(\mathbf{M}, \mathbf{S}) = \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W} - \mathbf{MS}\|_F^2 \tag{6.4}$$

where $\|\cdot\|_F$ stands for the Frobenius norm of a matrix. The above algorithm may yield poor results in the presence of significant noise or outliers. In order to handle these challenging situations, researchers proposed a robust factorization approach to increase the robustness of the algorithm. One common practice is to introduce a weight matrix into the cost function in (6.4) as below:

$$Err(\mathbf{M}, \mathbf{S}) = \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|(\mathbf{W} - \mathbf{MS}) \otimes \Sigma\|_F^2 \tag{6.5}$$

where $'\otimes'$ denotes the Hadamard product, which is an element-by-element multiplication of two matrices; and $\Sigma = \{\sigma_{ij}\}$ is the uncertainty matrix whose entries are weights derived from the confidence of the image measurements [3][7]. However, the weight matrix is usually unavailable and is hard to retrieve.

Figure 6.1: Illustration of feature uncertainties. Different features have different uncertainties due to their spatial distribution in the image, as well as temporal distribution across the sequence. The features in the middle image have larger uncertainties than other images due to bad imaging quality of the frame.

## 6.3 Weighted Factorization

A spatial-and-temporal-weighted factorization algorithm is proposed in this section. First, the uncertainty of a feature is demonstrated to be generally in proportional to its reprojection residual, which in return is employed to determine its uncertainty spatially and temporally. Then, a weighted factorization scheme is derived from the estimated uncertainties.

### 6.3.1 Feature uncertainty modeling

Measurement errors are inevitable in the process of feature detection and tracking. As a result, the key step of all robust methods is the modeling of error distributions. The strategy most people use is to model the uncertainty of each feature based on the information such as sharpness and intensity contrast around its neighborhood. Some researchers modeled the errors isotropically with different variances [3][74]; while others adopted directional distribution to describe the uncertainties [7][49].

As shown in Figure 6.1, the uncertainty of very 'sharp' and 'smooth' features are highly directional in elliptical shape; while the uncertainty of a regular corner point is isotropic and has smaller magnitude than the very 'smooth' ones. The uncertainty is usually estimated during the process of feature detection and tracking or given as prior information. Nonetheless, this information is unavailable in many applications.

The accuracy of the tracked features is affected by many factors. Previous studies determine the uncertainties based only on the spatial information, such

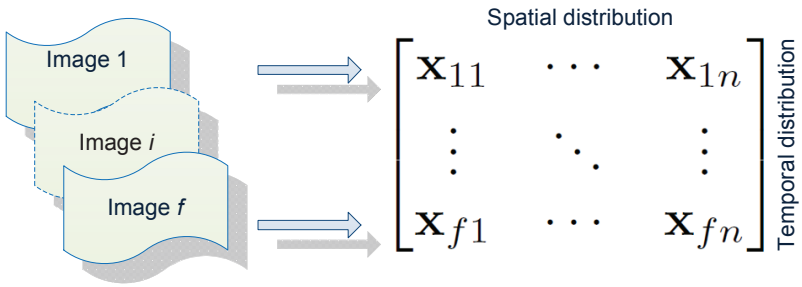



Figure 6.2: The spatial distribution and the temporal distribution of the tracking features.

as feature location, sharpness, and detection reliability, etc. In this chapter, the uncertainty is derived from a new perspective and not only the spatial information, but also the temporal information is introduced to model the error distribution of image measurement. The temporal uncertainty reflects to the errors among different frames caused by the variation of imaging conditions, such as illumination and stability.

As shown in Figure 6.1, the same feature may have different uncertainties across the sequence depending on the imaging quality of different frames, however, this temporal error is rarely considered in the existing approaches.

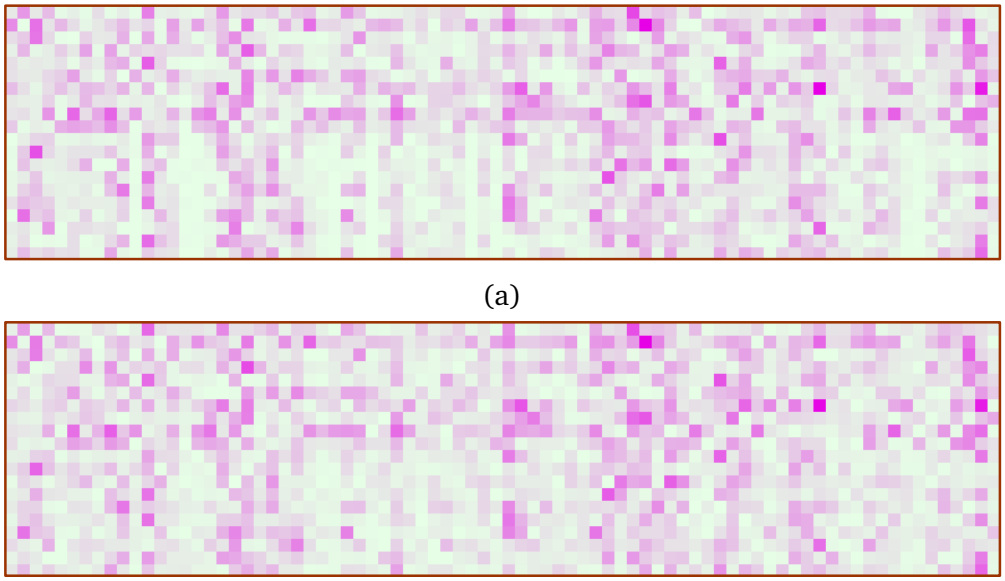
Consider the tracking matrix across the sequence, as shown in Figure 6.2. Different columns stand for the spatial distribution of the tracked features, while different rows contain the tracked data of different frames. This will be referred as temporal information hereafter.

Suppose $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ are the rank-3 affine factorization of a noise contaminated tracking matrix $\mathbf{W}$, the reprojection residuals can be computed by reprojecting the solution back onto all images and arranging in the following form.

$$\mathbf{E}_{rep} = \mathbf{W} - \tilde{\mathbf{M}}\tilde{\mathbf{S}} = \begin{bmatrix} e_{11} & \cdots & e_{1n} \\ \vdots & \ddots & \vdots \\ e_{m1} & \cdots & e_{mn} \end{bmatrix}_{m \times n} . \tag{6.6}$$

As will be investigated in Section 6.4, extensive experiments show that the residual error of each image point is generally proportional to the error magnitude added to that point. For example, let us arrange the image noise in the same form as the tracking matrix and denote them in a matrix form as

$$\mathbf{N}_{noise} = \begin{bmatrix} n_{11} & \cdots & n_{1n} \\ \vdots & \ddots & \vdots \\ n_{m1} & \cdots & n_{mn} \end{bmatrix}_{m \times n} \tag{6.7}$$



(a)

(b)

Figure 6.3: (a) Illustration of the normalized absolute value of the real added Gaussian noise matrix $\mathbf{N}_{noise}$, where the intensity of each pixel corresponds the error magnitude at that pixel; (b) illustration of the normalized absolute value of the reprojection residual matrix $\mathbf{E}_{rep}$.

where $n_{ij}$ is the noise or error added to the $(i, j)$-th element of the tracking matrix $\mathbf{W}$. Fig. 6.3(a) illustrates an example of the real Gaussian noise $\mathbf{N}_{noise}$ added to an image sequence of ten frames with 80 tracked features. Fig. 6.3(b) depicts the reprojection residuals (6.6) by using a regular affine factorization algorithm. It is evident that the reprojection residuals have very similar distribution as the added noise. Please refer to Section 6.4 for the analysis and more results.

The above observation suggests that the image errors can be largely estimated by virtue of the reprojection residuals. In the following, the feature errors in an image will be modeled spatially and temporally from the residuals.

**Spatial error estimation:** Let us write the reprojection residual matrix (6.6) into the following form.

$$\mathbf{E}_{rep} = \left[\mathbf{e}_1^s, \cdots, \mathbf{e}_n^s\right]_{m\times n} \tag{6.8}$$

where $\mathbf{e}_j^s$ is the $j$-th column of $\mathbf{E}_{rep}$. Obviously, $\mathbf{e}_j^s$ contains the reprojection error of the feature $j$ across the sequence. Assume the image noise is Gaussian, then

the average noise level of $\mathbf{e}_j^s$ can be embodied by its standard deviation

$$\delta_j^s = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(e_{ij} - \mu_j^s)^2} \tag{6.9}$$

where

$$\mu_j^s = \frac{1}{m}\sum_{i=1}^{m} e_{ij} \tag{6.10}$$

is the mean of the error vector $\mathbf{e}_j^s$. Stacking the noise levels $\delta_j^s$ of every features into a vector form yields

$$\mathbf{E}^s = \left[\delta_1^s, \cdots, \delta_n^s\right]^T. \tag{6.11}$$

As discussed above, $\mathbf{E}^s$ can actually reflect the average noise distribution among different features. Therefore, it is named as spatial error of the tracking data. Equation (6.11) provides a new perspective in recovering the spatial errors from available data.

**Temporal error estimation:** Let us rewrite the residual matrix (6.6) into the following from.

$$\mathbf{E}_{rep} = \begin{bmatrix} \mathbf{e}_1^t \\ \vdots \\ \mathbf{e}_m^t \end{bmatrix}_{m\times n} \tag{6.12}$$

where $\mathbf{e}_i^t = [e_{i1}, \cdots, e_{in}]$ is the reprojection residuals of all features in the $i$-th frame. Similarly, the noise level of the $i$-th frame is reflected by the standard deviation of $\mathbf{e}_i^t$ as shown below.

$$\delta_i^t = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(e_{ij} - \frac{1}{n}\sum_{i=1}^{n} e_{ij}\right)^2}. \tag{6.13}$$

Similar to the definition of (6.11), the temporal error is defined by stacking $\delta_i^t$ for every frames.

$$\mathbf{E}^t = \left[\delta_1^t, \cdots, \delta_m^t\right]^T. \tag{6.14}$$

Evidently, $\mathbf{E}^t$ embodies the average noise distribution at different time instants across the sequence. The temporal information was rarely considered in previous studies.

### 6.3.2 Spatial-and-temporal-weighted factorization

Based on the above estimated spatial and temporal errors, an uncertainty matrix of the tracking data is calculated and applied to the weighted factorization (6.5).

The weighted factorization can be modeled by considering only the spatial error, temporal error, or both errors.

**Spatial-weighted factorization:** If only the spatial error (6.11) is considered, the weight matrix can be defined in the form of

$$\Sigma_1 = \begin{bmatrix} \begin{pmatrix} \frac{\delta_0^s}{\delta_1^s} \\ \vdots \\ \frac{\delta_0^s}{\delta_1^s} \end{pmatrix} & \begin{pmatrix} \frac{\delta_0^s}{\delta_2^s} \\ \vdots \\ \frac{\delta_0^s}{\delta_2^s} \end{pmatrix} & \cdots & \begin{pmatrix} \frac{\delta_0^s}{\delta_n^s} \\ \vdots \\ \frac{\delta_0^s}{\delta_n^s} \end{pmatrix} \\ & & \cdots & \\ & & \cdots & \end{bmatrix}_{m \times n} \tag{6.15}$$

where $\Sigma_1$ is an $m \times n$ matrix with each column having identical entries; and $\delta_0^s$ is a normalization scalar. In practice, one can choose $\delta_0^s = \min(\mathbf{E}^s)$ such that each element in the weight matrix is no more than 1. Due to the special form of $\Sigma_1$, the weighted tracking matrix can be expressed as

$$\mathbf{W}^s = \mathbf{W} \otimes \Sigma_1 = \mathbf{W} \begin{bmatrix} \frac{\delta_0^s}{\delta_1^s} & & \\ & \ddots & \\ & & \frac{\delta_0^s}{\delta_n^s} \end{bmatrix}_{n \times n} = \mathbf{W}\Sigma^s \tag{6.16}$$

where

$$\Sigma^s = diag\Big(\frac{\delta_0^s}{\delta_1^s}, \frac{\delta_0^s}{\delta_2^s}, \cdots, \frac{\delta_0^s}{\delta_n^s}\Big) \tag{6.17}$$

is called spatial-weight matrix. Note that the size of the weight matrix $\Sigma_1$ is of dimension $m \times n$, the same as the tracking matrix; while the spatial-weight matrix $\Sigma^s$ is an $n \times n$ diagonalized matrix. Consequently, the spatial-weighted factorization (SWF) is given by

$$\begin{aligned} Err^s(\mathbf{M}, \mathbf{S}) &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|(\mathbf{W} - \mathbf{MS}) \otimes \Sigma_1\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|(\mathbf{W} - \mathbf{MS})\Sigma^s\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}\Sigma^s - \mathbf{M}(\mathbf{S}\Sigma^s)\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}^s - \mathbf{MS}^s\|_F^2. \end{aligned} \tag{6.18}$$

The factorization of (6.18) is exactly the same as the regular affine factorization algorithm (6.4) of the tracking matrix $\mathbf{W}^s$, where the motion matrix $\mathbf{M}$ is obtained directly, while the shape matrix can be simply recovered from $\mathbf{S}^s$ as follows.

$$\mathbf{S} = \mathbf{S}^s(\Sigma^s)^{-1}. \tag{6.19}$$

**Temporal-weighted factorization:** The weight matrix can be written as follows when only the temporal error (6.14) is present.

$$\Sigma_2 = \begin{bmatrix} \overbrace{\delta_0^t/\delta_1^t \cdots \delta_0^t/\delta_1^t}^{} \\ \vdots \quad \vdots \quad \vdots \\ \underbrace{\delta_0^t/\delta_m^t \cdots \delta_0^t/\delta_m^t}_{} \end{bmatrix}_{m\times n} \tag{6.20}$$

where $\delta_0^t = \min(\mathbf{E}^t)$ is the normalization scalar and each row in $\Sigma_2$ has $n$ identical entries. Consequently, the temporal-weighted tracking matrix can be simply denoted as the following form.

$$\mathbf{W}^t = \mathbf{W} \otimes \Sigma_2 = \begin{bmatrix} \frac{\delta_0^t}{\delta_1^t} & & \\ & \ddots & \\ & & \frac{\delta_0^t}{\delta_m^t} \end{bmatrix}_{m\times m} \mathbf{W} = \Sigma^t \mathbf{W} \tag{6.21}$$

where the $m \times m$ diagonal matrix

$$\Sigma^t = diag\Big(\frac{\delta_0^t}{\delta_1^t}, \frac{\delta_0^t}{\delta_2^t}, \cdots, \frac{\delta_0^t}{\delta_m^t}\Big) \tag{6.22}$$

is named as temporal-weight matrix. Unlike the SWF algorithm, the temporal-weighted factorization (TWF) is equivalent to left multiplication of the temporal-weight matrix as follows.

$$\begin{aligned} Err^t(\mathbf{M},\mathbf{S}) &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|(\mathbf{W}-\mathbf{M}\mathbf{S})\otimes\Sigma_2\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\Sigma^t(\mathbf{W}-\mathbf{M}\mathbf{S})\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\Sigma^t\mathbf{W}-(\Sigma^t\mathbf{M})\mathbf{S}\|_F^2 \\ &= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}^t-\mathbf{M}^t\mathbf{S}\|_F^2. \end{aligned} \tag{6.23}$$

Therefore, the factorization of the temporal-weighted tracking matrix directly yields the structure matrix $\mathbf{S}$; while the motion matrix can be decoupled from $\mathbf{M}^t$ by left multiplication of the inverse of the temporal-weight matrix.

$$\mathbf{M} = (\Sigma^t)^{-1}\mathbf{M}^t. \tag{6.24}$$

**Spatial-and-temporal-weighted factorization:** Based on the above analysis, the spatial-and-temporal-weighted factorization (STWF) can be obtained simply by combining the equations (6.18) and (6.23) together as follows.

$$Err^{st}(\mathbf{M},\mathbf{S}) = \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\Sigma^t(\mathbf{W}-\mathbf{M}\mathbf{S})\Sigma^s\|_F^2$$

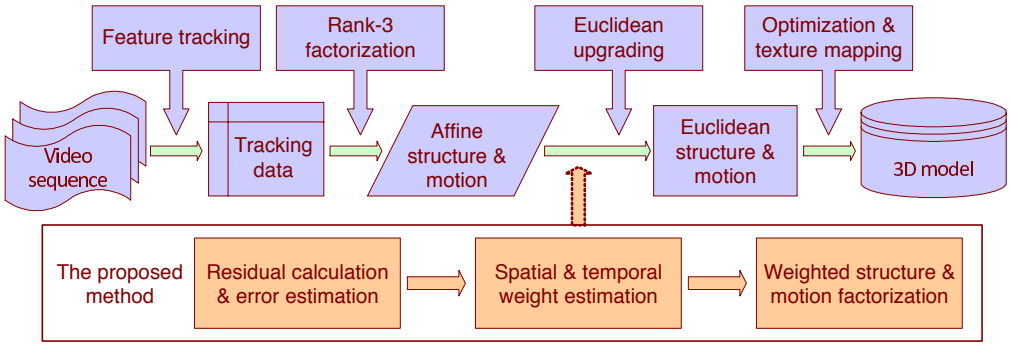



Figure 6.4: Outline of the proposed method and implementation flowchart.

$$= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\Sigma^t \mathbf{W} \Sigma^s - (\Sigma^t \mathbf{M})(\mathbf{S}\Sigma^s)\|_F^2$$
$$= \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}^{st} - \mathbf{M}^t \mathbf{S}^s\|_F^2. \tag{6.25}$$

The factorization of the spatial-and-temporal-weighted tracking matrix yields a temporal-weighted motion matrix $\mathbf{M}^t$ and a spatial-weighted shape matrix $\mathbf{S}^s$, which can be decoupled from equations (6.24) and (6.19), respectively.

Due to the introduction of the spatial-weight and temporal-weight matrices, it is evident that the complicated element-by-element multiplication of the weight matrix in (6.5) is converted into simple matrix multiplication in the proposed weighted factorization (6.18), (6.23), and (6.25). Consequently, the decoupling process for the final structure and motion matrices is greatly simplified.

### 6.3.3 Implementation details

The structure and motion recovered in the above weighted factorization is in affine space. The solution is then upgraded to the Euclidean space by imposing metric constraint to the motion matrix. Finally, a scheme of bundle adjustment is usually carried out to refine the final results. The proposed approach is shown in Figure 6.4 and the implementation details are outlined as follows.

1. Seek correspondences of feature points across all frames of the sequence;
2. Register the tracked features in each image to their centroid;
3. Perform an initial affine factorization and calculate the reprojection residuals according to (6.6);
4. Estimate the spatial error and the temporal error from (6.11) and (6.14);

5. Construct the spatial-weight matrix (6.17) and the temporal-weight matrix (6.22);

6. Perform the spatial-and-temporal-weighted factorization according to (6.25);

7. Recover the motion matrix and the shape matrix from (6.24) and (6.19);

8. Determine the metric upgrading matrix and upgrade the solution to the Euclidean space;

9. Refine the solution via bundle adjustment if necessary.

## 6.4 Evaluations on Synthetic Data

The proposed technique was validated and evaluated extensively on synthetic data and compared with the previous algorithms. During the simulation, 100 space points were randomly generated within a cube of $40 \times 40 \times 40$, and a sequence of 50 images were produced from these points by the affine projection. The following settings were used in the test: image resolution: $800 \times 800$; focal lengths varying randomly from 500 to 550; rotation angles randomly between $-60^\circ$ and $+60^\circ$; camera positions randomly inside a sphere with a diameter of 40; average distance from the cameras to the object: 600. These imaging conditions were very close to the assumption of affine projection.

### 6.4.1 Recovery of spatial and temporal errors

For the generated images, Gaussian white noise was added to the imaged points and the standard deviation of the noise levels was set at 3. First, the rank-3 affine factorization of the tracking matrix was performed to recover the initial motion and shape matrices. Then, the solution was reprojected back onto the images and the reprojection residual matrix was calculated via (6.6). Fig. 6.5 shows the actual reprojection residuals of the first 50 points in one images. It is evident that the variational trend of the reprojection residuals and the real added noise are similar to each other and their magnitudes are largely in proportion.

From the reprojection residuals, the spatial error (6.11) and the temporal error (6.14) of the entire sequence were recovered. As shown in Fig. 6.6, from which we can see that the two errors recovered by the proposed technique are very close to the ground truth of the noise added to the images.

In the above tests, the images were generated via affine projection model, which is the first-order approximation of the full perspective projection model. In practice, the images are actually formulated by a full perspective model.

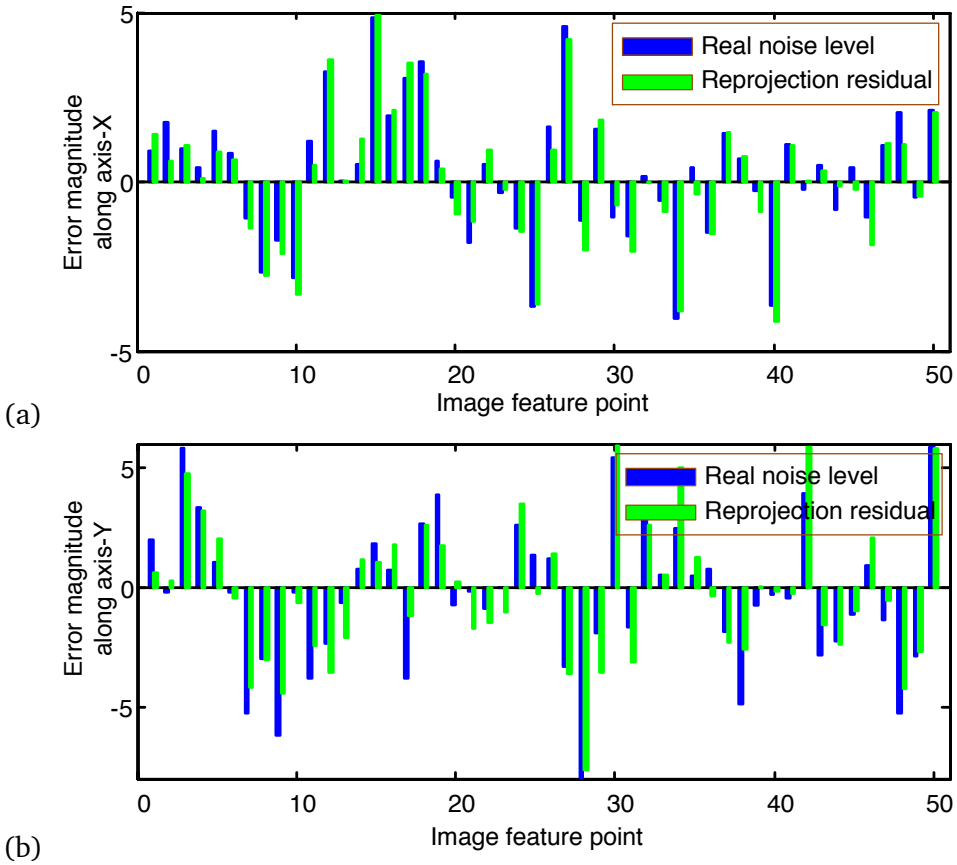



Figure 6.5: The real values of the added Gaussian noise (unit: pixel) and the reprojection residuals (unit: pixel) associated with the first 50 features in the first image. (a) The values along X direction; and (b) the values along Y direction.

Consequently, the errors always exist for rank-3 affine factorization, even though the images are captured free of noise.

In order to verify whether the proposed approach can be applied to the general perspective model, another set of images was generated by a full perspective projection model using the same data and camera parameters. The spatial and temporal errors were then estimated from the perspective images using the proposed approach. As shown in Fig. 6.6, the recovered errors are largely close to the ground truth in terms of both variational trend and magnitude, although they are not as close as the results under the affine assumption.

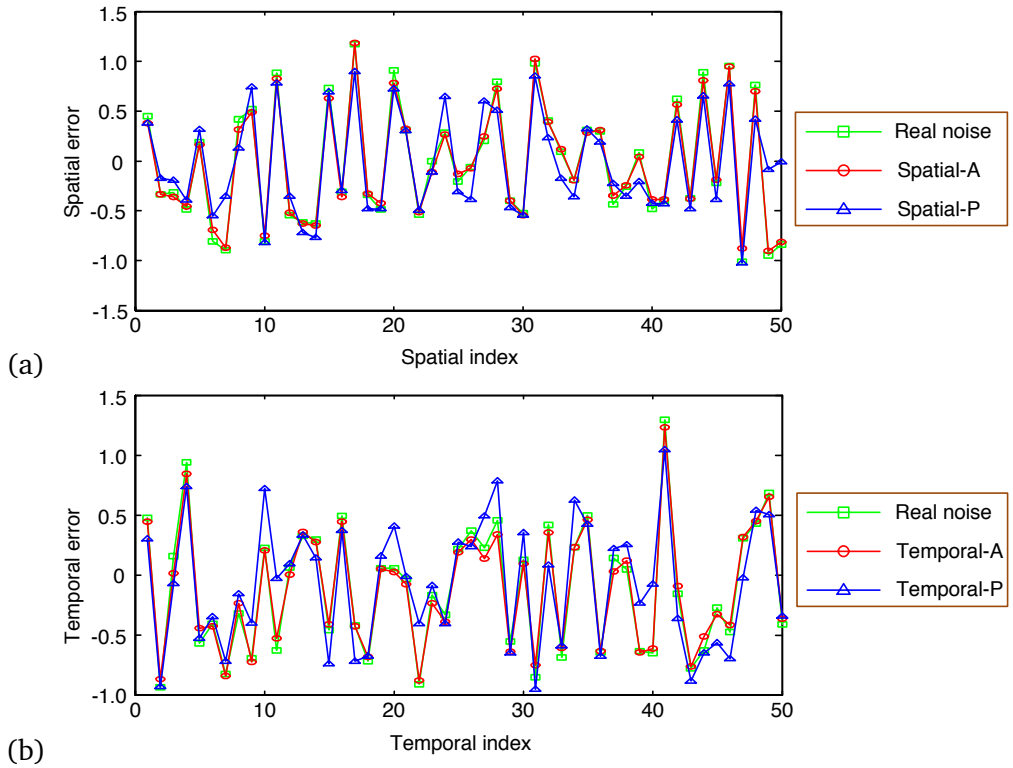



Figure 6.6: (a) The spatial error of the synthetic sequence; (b) the temporal error of the synthetic sequence. Where only the first 50 values are plotted; 'Real noise' stands for the real added noise; 'Spatial-A' and 'Temporal-A' denote the spatial and temporal errors under affine model; and 'Spatial-P' and 'Temporal-P' denote the spatial and temporal errors under perspective projection model.

### 6.4.2 Weighted factorization under affine model

The proposed factorization algorithm was evaluated and compared using the affine generated images in the following with respect to three different kinds of added noise to the images.

**Spatial noise:** In this case, significant spatial noise was added to the images, i.e., the magnitude of the added noise mainly varied with respect to the features; while the inter-frame noise variation was insignificant, as shown in Fig. 6.7(a). Using the noisy data, the structure and motion were recovered by the proposed SWF, TWF, and STWF algorithms. As a comparison, three benchmark algorithms, i.e. the regular rank-3 affine factorization without uncertainty weight [87], the scalar-weighted factorization [3], and the covariance-weighted factorization [1], were also implemented. Since the above factorizations are in affine space, their

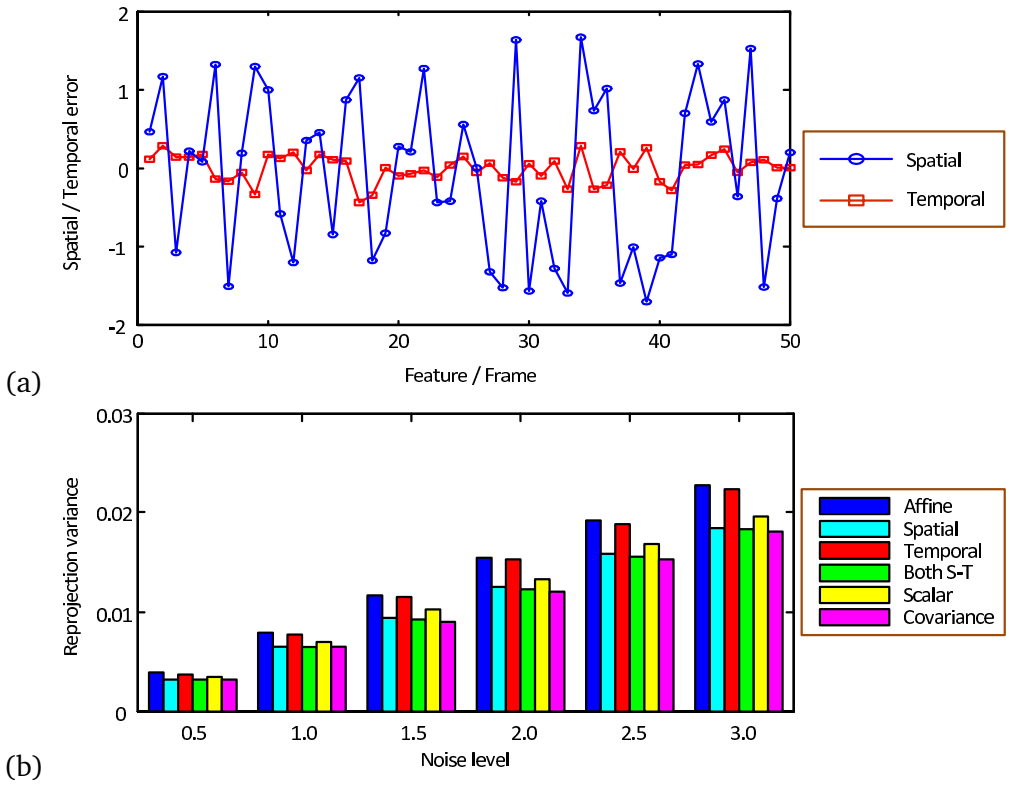



Figure 6.7: (a) The spatial and temporal noises (unit: pixel) added to the images, where the temporal noise is insignificant; (b) the reprojection variation (unit: pixel) obtained using different algorithms with respect to different noise levels.

performances are evaluated and compared by means of the reprojection variance, which is defined as follows:

$$E_{rv} = \frac{1}{mn}\|\mathbf{W}_0 - \tilde{\mathbf{M}}\tilde{\mathbf{S}}\|_F^2 \tag{6.26}$$

where $\mathbf{W}_0$ is the noise-free tracking matrix; $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{S}}$ are the estimated motion and shape matrices, respectively. In order to obtain a statistically meaningful comparison, the temporal noise level (i.e., the standard deviation of the noise) was varied from 1 to 3 pixels in steps of 0.5 during the test, and 100 independent tests were performed at each noise level; while the spatial noise level was set at four times of the temporal noise level. The mean reprojection variances (6.26) obtained by different algorithms are shown in Fig. 6.7(b).

Based on this test results, the following Conclusion can be drawn: (i) the SWF and STWF algorithms perform similarly, and the accuracy of the solution is significantly improved by the two algorithms; (ii) the TWF algorithm has

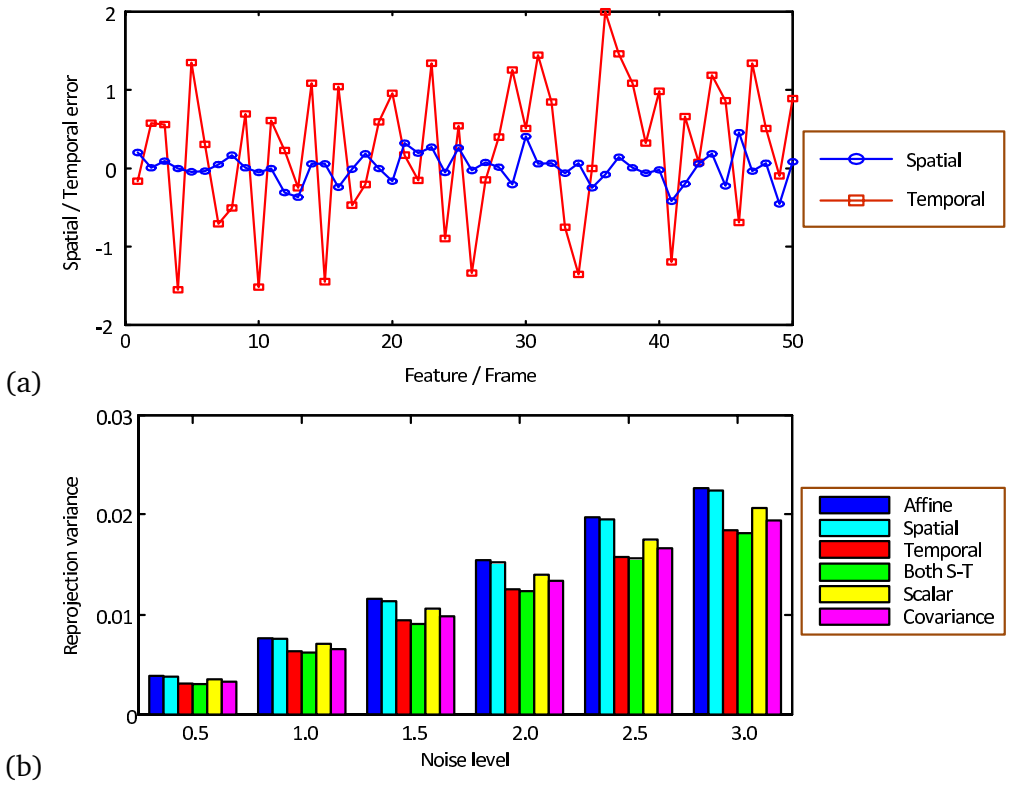



Figure 6.8: (a) The spatial and temporal noises (unit: pixel) added to the images, where the spatial noise is insignificant; (b) the reprojection variation (unit: pixel) obtained using different algorithms with respect to different noise levels.

only a minor effect since the temporal error is insignificant; (iii) the scalar-weighted factorization does not make big improvement, because the algorithm assumes orthographic camera model and isotropic uncertainty; and (iv) the covariance-weighted algorithm assumes directional uncertainty and has comparable performance as the STWF in this case.

**Temporal noise:** In this test, a significant temporal noise was added to the images over minor spatial noise. As shown in Fig. 6.8(a), the noise variation among different features is insignificant. Similar to the test in the first case, the reprojection variance recovered by different algorithms were evaluated. The mean errors at different noise levels are shown in Fig. 6.8(b).

It is evident from Fig. 6.8 that both the TWF and STWF algorithms significantly outperform the regular affine factorization, as well as the scalar-weighted and the covariance-weighted algorithms. The SWF algorithm shows minor improvements due to the insignificance of the spatial error. The performance of the scalar-

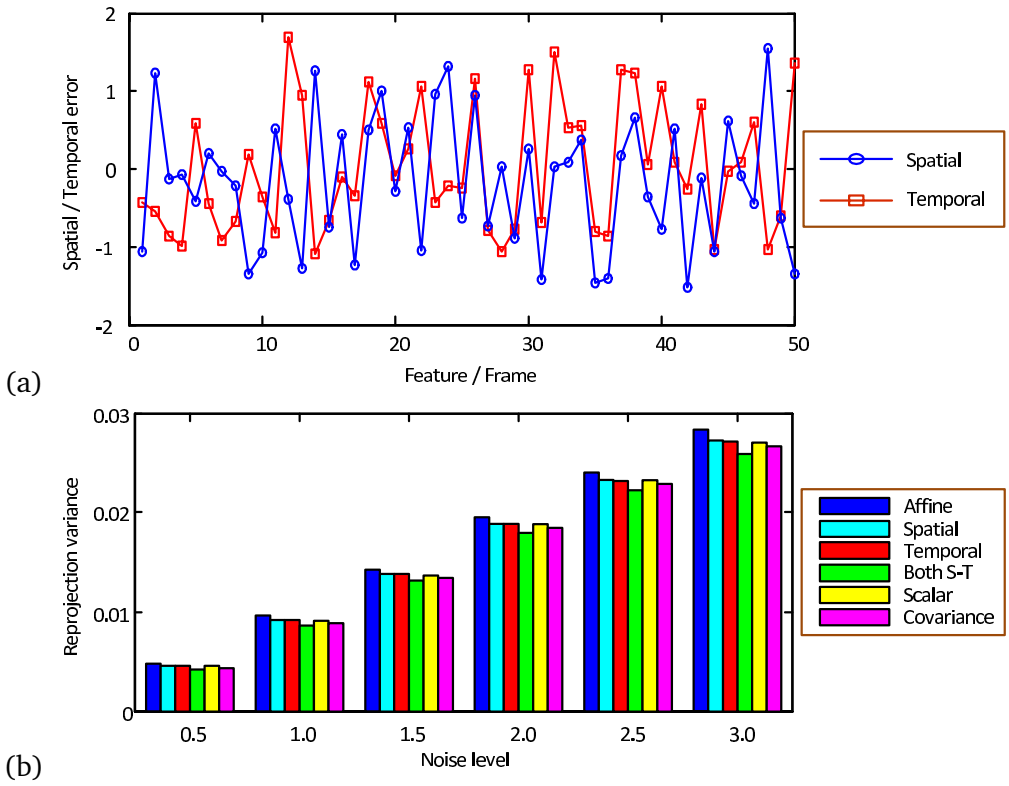



Figure 6.9: (a) The spatial and temporal noises (unit: pixel) added to the images, where both noises are present significantly; (b) the reprojection variation (unit: pixel) obtained using different algorithms with respect to different noise levels.

weighted and the covariance-weighted algorithms lie in between the regular affine factorization and the proposed STWF, since they only model the noise spatially without explicitly consideration of the influence of temporal noise. The proposed TWF and the STWF are specifically designed to deal with the temporal noise, it is not surprising that both of them exhibit the best performance in this experiment.

**Spatial noise + temporal noise:** All algorithms were evaluated under a general situation, where a significant spatial noise was applied among different features and an equivalent amount of temporal noise was applied across the sequence, as shown in Fig. 6.9(a). Using the noise contaminated date, the mean reprojection variances of different algorithms with respect to varied noise levels are plotted in Fig. 6.9(b).

It can be concluded from this test that (i) the STWF performs the best over other approaches since it can handle both kinds of errors simultaneously. How-

ever, as the spatial error and temporal error are coupled with each other, the performance of STWF is not as significant as when only one kind of error is applied; (ii) the SWF, TWF, and scalar-weighted factorization have comparable improvement over the regular affine factorization; and (iii) the performance of the covariance-weighted algorithm lies in between, since it could not effectively handle the temporal noise.

In many real applications, the temporal error is usually coupled with the spatial error in image measurements due to the variation in imaging quality and illumination. Therefore, the STWF algorithm is the best choice for any situation. In addition, as will be discussed in the following, the STWF algorithm does not require extra computational cost.

### 6.4.3 Weighted factorization under perspective projection

The above tests were performed using affine-generated images. As a comparison, the same tests were also carried out using the data from full perspective projection with other parameters remained the same. The corresponding results, as shown in Fig. 6.10, are similar to those under the affine model. Nevertheless, some differences are obvious: (i) the overall error magnitude is larger when using the data generated by the perspective model than those from the affine assumption; (ii) all weighted factorization algorithms do not have noticeable effect when the noise levels are lower than 1 pixel. This phenomenon is easy to explain: all factorization algorithms discussed in the paper are based on the affine assumption, however, the data here are produced by the perspective projection. Therefore, the data are "*inaccurate*" to the algorithms even when they are free of noise. As a consequence, the small added noise is overwhelmed by the "*inaccurate*" data. This conclusion also applies to the scalar-weighted and the covariance-weighted algorithms; and (iii) the proposed STWF has the best performance over other weighted algorithms.

### 6.4.4 Computational complexity

In this test, the real computational cost of different approaches were compared. All algorithms were implemented in Matlab 6.5 on an AMD Athlon$^{TM}$dual core processor desktop with 2.2GHz CPU; the feature point number was set at 100, and the frame number was varied from 50 to 300 in steps of 50 so as to generate different sizes of the tracking matrix. Table 8.1 shows the real computation time of the STWF algorithm with respect to its counterparts.

As listed in Table 8.1, the scalar-weighted factorization is more computation efficient than other algorithms, this is because it is a rank-1 factorization, which

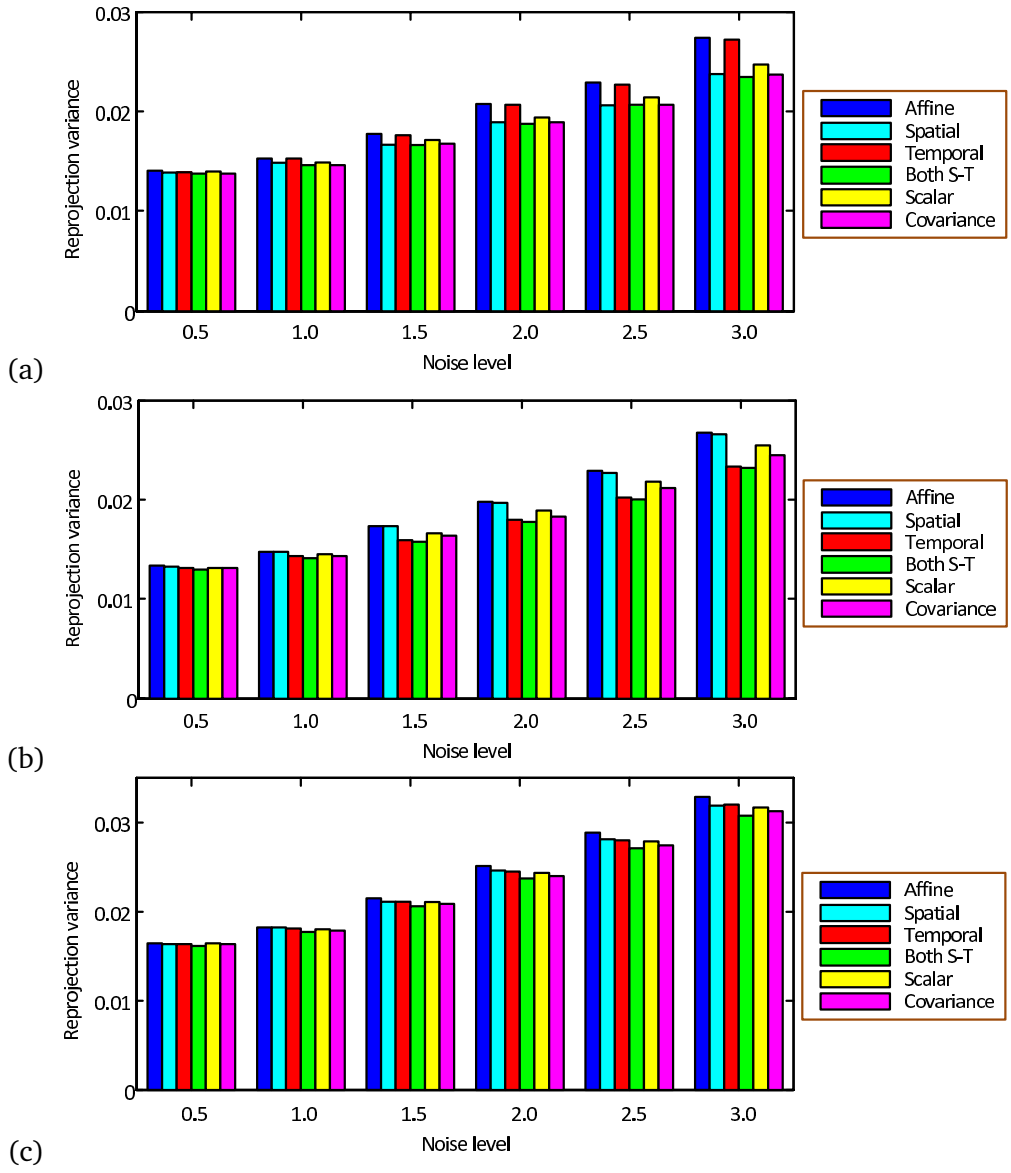



Figure 6.10: The reprojection variation obtained using the data generated from full perspective projection with respect to different noise levels. (a) Significant spatial noise v.s. insignificant temporal noises; (b) significant temporal noise v.s. insignificant spatial noises; (c) both spatial and temporal noises are significant.

Table 6.1: Real computation time of different algorithms (unit: second).

| Frame no. | 50 | 100 | 150 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|
| Affine | 0.016 | 0.047 | 0.094 | 0.156 | 0.250 | 0.354 |
| STWF | 0.047 | 0.110 | 0.235 | 0.495 | 0.859 | 1.297 |
| Scalar | 0.010 | 0.023 | 0.035 | 0.047 | 0.058 | 0.071 |
| Covariance | 0.045 | 0.106 | 0.226 | 0.487 | 0.848 | 1.283 |

can be implemented using power method, while other approaches use computational intensive SVD decomposition. The proposed STWF utilize the solution of the affine factorization to compute the weight matrices and then recalculate the factorization. Consequently, its computation time is in general about two times longer than that of the regular affine factorization. The covariance-weighted algorithm also includes two steps of SVD decomposition, thus, it has comparable computational complexity as the STWF. The time by the proposed SWF and TWF is not tabulated in the table. In general, they have similar computational cost as the STWF, since the only difference between them lies in the times of matrices inversion and multiplication.

It should be noted that the time of global optimization was not included here as this is the most computationally intensive stage in the process and its computation time varies with the iteration times. If the optimization step is included, all algorithms will have very close computation time.

## 6.5 Evaluations on Real Sequences

The method was tested on many real image sequences. The results on three data sets are reported here. These sequences were captured by a Canon Powershot G3 camera at Queens park, Stratford, Ontario. The image resolution was $1024 \times 768$, and the image correspondences were established by the feature tracking system based on the SIFT and epipolar constraints [80].

### 6.5.1 Test on garden sequence

The garden sequence consists of eight images, with 2189 features in total tracked across the sequence. As shown in Figure 6.11(a), although the tracking result seems reliable visually, there are indeed many errors as the scene contains a large portion of homogeneous and repetitive textures. It is, thus, hard to track accurately for this type of scene.

Using the proposed technique, the spatial and temporal errors were recovered

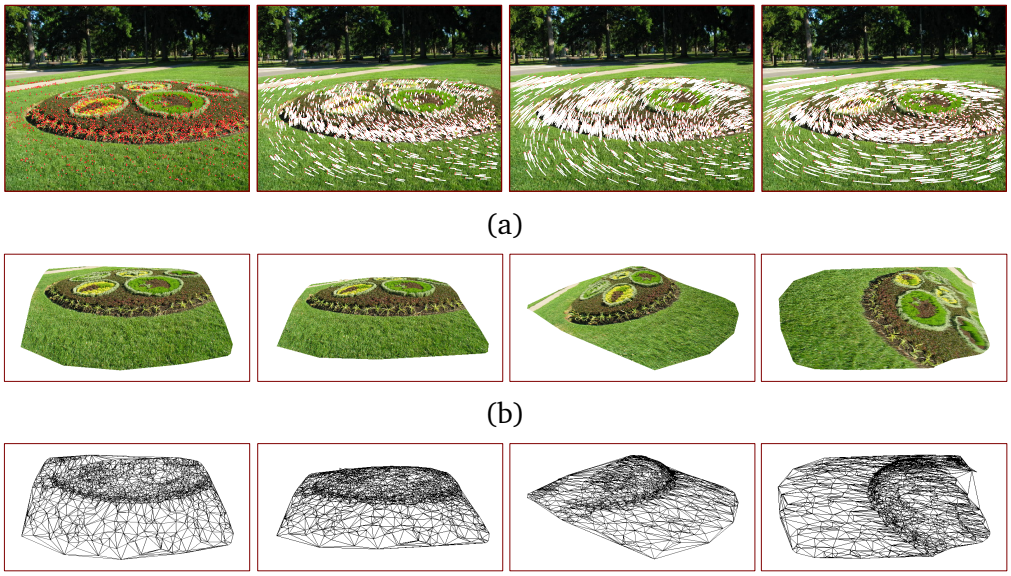


Figure 6.11: Reconstruction results of the garden sequence. (a) Four frames from the sequence, where the tracked features with disparities to the first image are overlaid to the images; (b) the reconstructed VRML model of the scene shown from different viewpoints with texture mapping; and (c) the corresponding triangulated wireframe of the VRML model.

and the 3D structure of the scene was reconstructed using the STWF algorithm, followed by a global optimization scheme. As shown in Figure 6.11, the structure has been correctly recovered and looks realistic. Since there is no ground truth of the scene, the following defined image reprojection error is used to evaluate the performance of different algorithms.

$$\bar{E}_{rep} = \frac{1}{nf} \sum_{i,j} dis(x_{ij},\ \hat{x}_{ij}) \tag{6.27}$$

where $x_{ij}$ and $\hat{x}_{ij}$ are the detected and reprojected image points, respectively; $dis(\bullet, \bullet)$ stands for the Euclidean distance of two points. The reprojection errors by the regular affine algorithm and the proposed algorithm are tabulated in Table 6.2, where 'Affine+LM', 'STWF+LM', 'Scalar+LM', and 'Covariance+LM' stand for the results after global optimization using the Levenberg-Marquardt algorithm [87]. It is evident that the error is reduced by the weighted algorithms. Like the results in the previous simulations, the proposed STWF algorithm gives the lowest reprojection error.

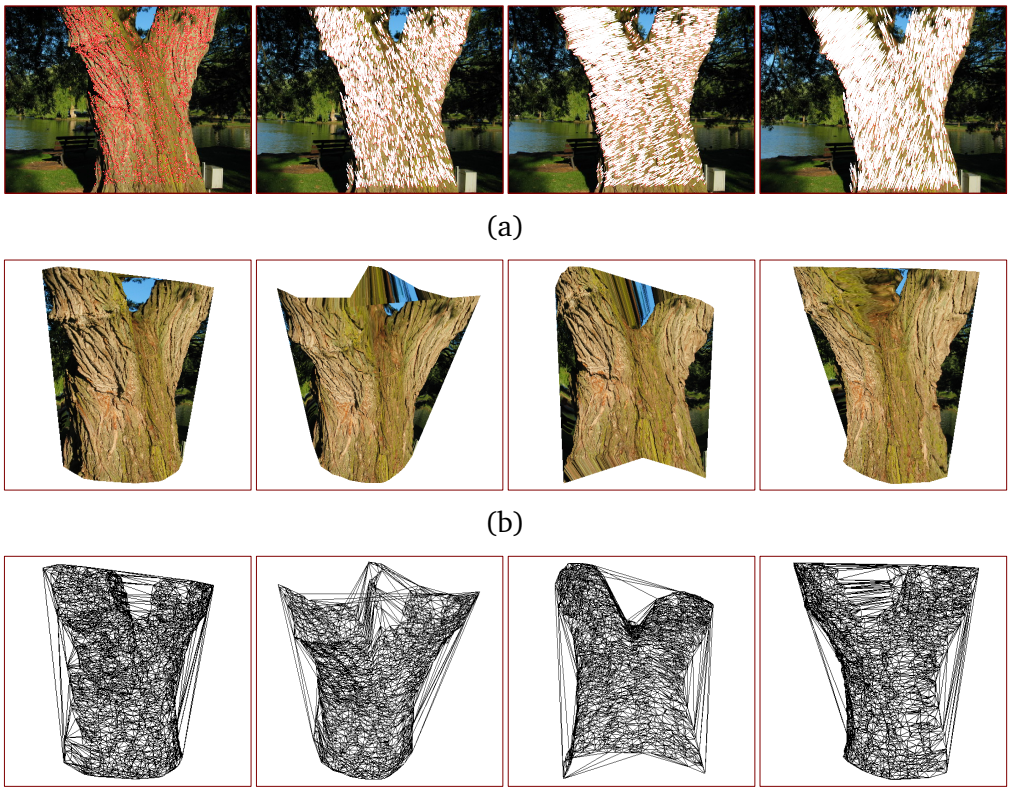


Figure 6.12: Reconstruction results of the tree trunk sequence. (a) Four frames from the sequence overlaid with the tracked features with disparities to the first image; (b) the reconstructed VRML model of the trunk shown from different viewpoints with texture mapping; and (c) the corresponding triangulated wireframe of the VRML model.

Table 6.2: The reprojection errors by different algorithms in real sequence tests.

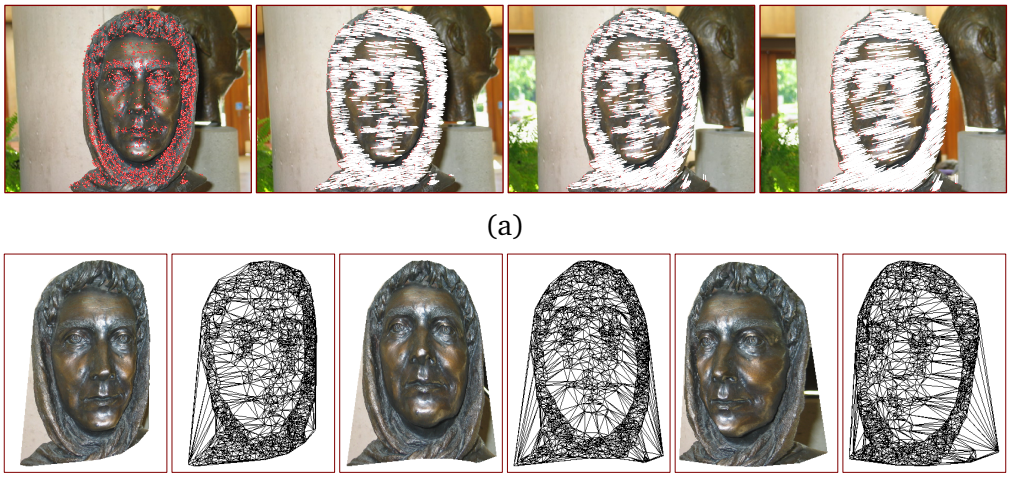
| Method | Garden | Trunk | Head |
|---|---|---|---|
| Affine | 0.785 | 0.566 | 0.608 |
| STWF | 0.697 | 0.493 | 0.534 |
| Scalar | 0.746 | 0.529 | 0.577 |
| Covariance | 0.712 | 0.506 | 0.543 |
| Affine+LM | 0.406 | 0.315 | 0.327 |
| STWF+LM | 0.349 | 0.274 | 0.281 |
| Scalar+LM | 0.374 | 0.297 | 0.301 |
| Covariance+LM | 0.355 | 0.286 | 0.294 |

### 6.5.2 Test on tree trunk sequence

There are nine images in the tree trunk sequence, and 4224 features were tracked across the sequence. As shown in Figure 6.12(a), the features were tracked densely, however, there are many errors due to the homogeneous texture of the trunk. We calculated the spatial- and temporal-weight matrices from the affine reprojection residuals and recovered the 3D structure and camera motions using the proposed algorithm. Figure 6.12 shows the reconstructed VRML model with texture mapping and the corresponding triangulated wireframes from different viewpoints. The model looks realistic and most details are correctly recovered by the proposed method. The reprojection errors obtained from different algorithms are listed in Table 6.2, from which we can see that the proposed STWF outperforms other weighted algorithms.

### 6.5.3 Test on statue sequence

The head statue sequence consists of seven images with 2676 features being tracked. Figure 6.13(a) shows the four frames overlaid with the tracked features. Figure 6.13(b) shows the recovered 3D model and the corresponding wireframe by the proposed STWF algorithm. The structure of the statue is correctly recovered. As a comparison with the regular affine factorization, the scalar-weighted, and the covariance-weighted algorithms. The reprojection errors are tabulated in Table 6.2, from which it is clear that the solution of the regular affine factorization is improved by the weighted algorithms, and the STWF exhibits the best performance.



(a)

(b)

Figure 6.13: Reconstruction results of the head statue sequence. (a) Four frames from the sequence overlaid with the tracked features with disparities to the first image; (b) the reconstructed VRML model and the corresponding triangulated wireframe of the head shown from different viewpoints.

## 6.6 Conclusion

The chapter has illustrated empirically that the errors contained in images can be modeled spatially and temporally from the image reprojection residuals. Thus, the feature uncertainties could be derived from the errors and a spatial-and-temporal-weighted factorization algorithm was proposed to handle the erroneous image measurements and improve the accuracy of the final solutions. Unlike other existing weighted factorization algorithms, the proposed technique requires no prior information of the error distributions of the tracked features. Extensive tests and evaluations validated the algorithm and demonstrated its advantages over the previous methods. Our recently study demonstrated that the proposed technique can also effectively handle outliers and missing data. A further study will be presented in the following two chapters.

# Chapter 7

# Robust SfM of Rigid Objects

The chapter presents a robust structure from motion algorithm for rigid objects. First, a rank-4 augmented affine factorization algorithm is proposed to overcome the difficulty in image alignment for imperfect data. Then, a robust structure and motion factorization scheme is proposed to handle outlying and missing data. The novelty and main contribution of the paper are as follows: (i) The rank-4 factorization algorithm is a new addition to previous affine factorization family using rank-3 constraint; (ii) it is demonstrated that image reprojection residuals are in general proportional to the error magnitude in the tracking data. Thus, the outliers can be detected directly from the distribution of image reprojection residuals, which are then used to estimate the weight matrix of inlying measurement; and (iii) the robust factorization scheme is demonstrated empirically to be more efficient and accurate than other robust algorithms. Extensive experiments on synthetic data and real images validate the proposed approach.

## 7.1 Introduction

Most factorization methods usually assume all features are tracked across the sequence. In the presence of missing data, SVD factorization can not be used directly. Some approaches propose to solve the motion and shape matrices alternatively, such as the alternative factorization [41], power factorization [33], and factor analysis [29]. In practice, outlying data are inevitable during the process of feature tracking, and as a consequence, the performance of the algorithm will degenerate. The most popular strategy in computer vision field is RANSAC, Least Median of Squares [32], and other similar hypothesis-and-test frameworks [67]. However, these methods are usually designed for two or three views.

In recent years, the problem of robust factorization has received a lot of attention [2][63][100]. Aguitar and Moura [3] proposed a scalar-weighted SVD algorithm that minimizes the weighted square errors. Gruber and Weiss [29] formulated the problem as a factor analysis and derived an Expectation Maximization (EM) algorithm to enhance the robustness to missing data and uncertainties. Zelnik-Manor *et al.* [103] defined a new type of motion consistency based on temporal consistency, and applied it to multi-body factorization with directional uncertainty. Zaharescu and Horaud [100] introduced a Gaussian mixture model and incorporate it with the EM algorithm. Huynh *et al.* [39] proposed an iterative approach to correct the outliers with 'pseudo' observations.

Ke and kanade [41] proposed a robust algorithm to handle outliers by minimizing a $L1$ norm of the reprojection errors. Eriksson and Hengel [23] introduced the $L1$ norm to the Wiberg algorithm to handle missing data and outliers. Okatani *et al.* [55] proposed to incorporate a damping factor into the Wiberg method to solve the problem. Yu *et al.* [99] presented a Quadratic Program formulation for robust multi-model fitting of geometric structures. Wang *et al.* [92] proposed an adaptive kernel-scale weighted hypotheses to segment multiple-structure data even in the presence of a large number of outliers. Paladini *et al.* [60] proposed an alternating bilinear approach to solve nonrigid SfM by introducing a globally optimal projection step of the motion matrices onto the manifold of metric constraints. Wang *et al.* [89] proposed a spatial-and-temporal-weighted factorization approach to handle significant noise in the measurement.

In this chapter, the outlying data are detected from a new viewpoint via image reprojection residuals by exploring the fact that the reprojection residuals are in general proportional to measurement errors of the tracking data. This chapter first proposes an augmented rank-4 factorization algorithm to circumvent the problem of image registration in the presence of missing and outlying data. Then, an alternative weighted factorization algorithm is presented to handle the missing features and image uncertainty. Finally, a robust factorization scheme is proposed to handle outliers. A preliminary work on this research was published at the WACV conference [91].

## 7.2 Background on Structure and Motion Factorization

Under affine projection model, a 3D point $\bar{\mathbf{X}}_j = [x_j, y_j, z_j]^T$ is projected onto an image point $\bar{\mathbf{x}}_{ij} = [u_{ij}, v_{ij}]^T$ in frame $i$ according to the imaging equation

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i \bar{\mathbf{X}}_j + \mathbf{c}_i \tag{7.1}$$

where the matrix $\mathbf{A}_i$ is a $2 \times 3$ affine projection matrix; and $\mathbf{c}_i$ is a translation term of the frame. Under the affine projection, the mapping from space to the

image is linear. From (7.1), the imaging process of all space points to the $i$-th frame can be denoted as:

$$[\bar{\mathbf{x}}_{i1}, \bar{\mathbf{x}}_{i2}, \cdots, \bar{\mathbf{x}}_{in}] = \mathbf{A}_i[\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \cdots, \bar{\mathbf{X}}_n] + \mathbf{C}_i \tag{7.2}$$

where $\mathbf{C}_i = [\mathbf{c}_i, \mathbf{c}_i, \cdots, \mathbf{c}_i]$ is the translation matrix of frame $i$. Therefore, the imaging equation of an image sequence can be formulated by stacking Equation (7.2) frame by frame.

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_m \end{bmatrix}}_{\mathbf{M}_{2m\times 3}} \underbrace{[\bar{\mathbf{X}}_1, \cdots, \bar{\mathbf{X}}_n]}_{\bar{\mathbf{S}}_{3\times n}} + \underbrace{\begin{bmatrix} \mathbf{C}_1 \\ \vdots \\ \mathbf{C}_m \end{bmatrix}}_{\mathbf{C}_{2m\times n}} \tag{7.3}$$

where $m$ is the frame number; and $n$ is the number of features. It is easy to verify that $\mathbf{c}_i$ in (7.1) is the image of the centroid of all space points. Thus, if all imaged points in each image are registered to the centroid and relative image coordinates with respect to the centroid are employed, the translation term vanishes, i.e. $\mathbf{c}_i = \mathbf{0}$ and $\mathbf{C} = \mathbf{0}$. Consequently, the imaging process (7.3) is written concisely as

$$\mathbf{W}_{2m\times n} = \mathbf{M}_{2m\times 3}\bar{\mathbf{S}}_{3\times n}. \tag{7.4}$$

It is obvious from (7.3) that the tracking matrix is highly rank-deficient, and the rank of $\mathbf{W}$ is at most 3 if the translation term $\mathbf{C}$ is removed. In practice, the rank of a real tracking matrix is definitely greater than 3 due to image noise and affine approximation error. Thus, one need to find a rank-3 approximation of the tracking matrix. A common practice is to perform SVD decomposition on the matrix $\mathbf{W}$ and truncate it to rank 3, then the motion matrix $\mathbf{M}$ and the shape matrix $\mathbf{S}$ can be easily decomposed from the tracking matrix.

## 7.3 Rank-4 Structure from Motion

Previous studies on affine structure from motion are based on rank-3 factorization (7.4) due to its conciseness and simplicity. One necessary condition of the rank-3 factorization is that all image measurements are registered to the corresponding centroid of each frame. However, when some tracked features are missing, or contaminated with outliers or significant noise, it is impossible to reliably retrieve the centroid. As will be shown in the experiments, the misalignment of the centroid will cause a big error to the final solution. Previous studies were either simply ignoring this problem or hallucinating the missing points with pseudo observations, which may lead to a biased solution. In this section, an augmented rank-4 factorization algorithm is proposed to solve this problem.

### 7.3.1 Rank-4 affine factorization

Let us rewrite the affine imaging process (7.1) as:

$$\bar{\mathbf{x}}_{ij} = \left[\mathbf{A}_i | \mathbf{c}_i\right] \mathbf{X}_j \tag{7.5}$$

where the space point $\bar{\mathbf{X}}_j$ is denoted by a 4-dimensional homogeneous form as $\mathbf{X}_j = [\bar{\mathbf{X}}_j, 1]^T$. Then, the projection for the entire sequence is formulated as:

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\left[\begin{array}{c|c} \mathbf{A}_1 & \mathbf{c}_1 \\ \vdots & \vdots \\ \mathbf{A}_m & \mathbf{c}_m \end{array}\right]}_{\mathbf{M}_{2m\times 4}} \underbrace{\left[\mathbf{X}_1, \cdots, \mathbf{X}_n\right]}_{\mathbf{S}_{4\times n}}. \tag{7.6}$$

The factorization can be written concisely as:

$$\mathbf{W}_{2m\times n} = \mathbf{M}_{2m\times 4}\mathbf{S}_{4\times n}. \tag{7.7}$$

Obviously, the rank of the tracking matrix becomes 4 in this case. Thus, given the tracking matrix, the factorization can be simply obtained via SVD decomposition by imposing the rank-4 constraint.

In Equation (7.6), if all image points are perfectly registered to the corresponding centroids of every views and the origin of world system is set at the gravity center of the space points, i.e. $\sum_{j=1}^{n} \bar{\mathbf{X}}_j = \mathbf{0}$. Then, the last column of the motion matrix vanishes since $\mathbf{c}_i = \mathbf{0}$, the expression (7.5) is equivalent to rank-3 factorization (7.4). Nonetheless, the registration is hard to achieve due to image noise, outliers, or missing data in the tracking matrix, while the rank-4 factorization (7.5) does not require any image registration since it is derived directly from the affine projection model (7.1). Therefore, the rank-4 factorization always hold true for any tracking data.

Suppose the rank-4 SVD decomposition yields a set of solutions $\hat{\mathbf{M}}_{m\times 4}\hat{\mathbf{S}}_{4\times n}$. The decomposition is not unique since it is defined up to a nonsingular linear transformation $\mathbf{H} \in \mathbb{R}^{4\times 4}$ as $\mathbf{M} = \hat{\mathbf{M}}\mathbf{H}$ and $\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}}$. The upgrading matrix will be recovered in the following section.

### 7.3.2 Euclidean upgrading matrix

The upgrading matrix $\mathbf{H}$ is a $4 \times 4$ nonsingular matrix which can be denoted as:

$$\mathbf{H} = [\mathbf{H}_{1:3} | \mathbf{h}_4] \tag{7.8}$$

where $\mathbf{H}_{1:3}$ denotes the first three columns, and $\mathbf{h}_4$ is the fourth column. Suppose $\hat{\mathbf{M}}_i$ is the $i$-th two-row submatrix of $\hat{\mathbf{M}}$, then the upgraded motion matrix can be written as

$$\mathbf{M}_i = \hat{\mathbf{M}}_i\mathbf{H} = \left[\hat{\mathbf{M}}_i\mathbf{H}_{1:3} | \hat{\mathbf{M}}_i\mathbf{h}_4\right] = \left[\mathbf{A}_i | \mathbf{c}_i\right]. \tag{7.9}$$

The left part of $\mathbf{M}_i$ in (7.9) can be written as:

$$\hat{\mathbf{M}}_i \mathbf{H}_{1:3} = \mathbf{A}_i = f_i \begin{bmatrix} \mathbf{r}_{i1}^T \\ \mathbf{r}_{i2}^T \end{bmatrix} \tag{7.10}$$

where $f_i$ is the focal length of cameras, and $\mathbf{r}_{i1}^T$ and $\mathbf{r}_{i2}^T$ are the first two rows of the camera rotation matrix. Let us denote $\mathbf{Q} = \mathbf{H}_{1:3}\mathbf{H}_{1:3}^T$, then, $\mathbf{Q}$ can be constrained from (7.10) as:

$$\hat{\mathbf{M}}_i \mathbf{Q} \hat{\mathbf{M}}_i^T = \left(\hat{\mathbf{M}}_i \mathbf{H}_{1:3}\right)\left(\hat{\mathbf{M}}_i \mathbf{H}_{1:3}\right)^T = f_i^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7.11}$$

The above equation provides two independent constraints to $\mathbf{Q}$, which is a $4 \times 4$ positive semidefinite symmetric matrix with nine degree-of-freedom since it is defined up to a scale. Thus, the matrix $\mathbf{Q}$ can be linearly solved via least squares given five or more images. Furthermore, matrix $\mathbf{H}_{1:3}$ can be decomposed from $\mathbf{Q}$ via extended Cholesky decomposition as proved in [86].

After recovering $\mathbf{H}_{1:3}$, the last column of the upgrading matrix is then determined straightforwardly. From the expression (7.9), the projection Equation (7.5) can be written as

$$\bar{\mathbf{x}}_{ij} = \hat{\mathbf{M}}_i \mathbf{H}_{1:3} \bar{\mathbf{X}}_j + \hat{\mathbf{M}}_i \mathbf{h}_4. \tag{7.12}$$

It can be easily proved from (7.12) that the last column $\mathbf{h}_4$ corresponds to the translation from the world coordinate system to the image system. Under a given coordinate system, different values of $\mathbf{h}_4$ will only change the origin of the world system, however, it has no influence to the Euclidean structure of the reconstructed object. Thus, $\mathbf{h}_4$ can be set freely as any 4-vector which is independent of the columns of $\mathbf{H}_{1:3}$ so as to guarantee the nonsingularity of the resulted upgrading matrix. Practically, $\mathbf{h}_4$ may be constructed as follows.

Suppose the SVD decomposition of $\mathbf{H}_{1:3}$ is

$$\begin{aligned} \mathbf{H}_{1:3} &= \mathbf{U}_{4\times4} \Sigma_{4\times3} \mathbf{V}_{3\times3}^T \\ &= [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4] \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{bmatrix} [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]^T \end{aligned} \tag{7.13}$$

where $\mathbf{U}$ and $\mathbf{V}$ are two orthogonal matrices, and $\Sigma$ is a diagonal matrix of the three singular values. Then, $\mathbf{h}_4$ can be simply set as

$$\mathbf{h}_4 = \sigma_2 \mathbf{u}_4 \tag{7.14}$$

where $\sigma_2$ is the second singular value, and $\mathbf{u}_4$ is the last column of $\mathbf{U}$. The construction guarantees a good numerical stability in computing the inverse of $\mathbf{H}$, since the constructed matrix $\mathbf{H}$ has the same condition number as $\mathbf{H}_{1:3}$.

### 7.3.3 Algorithm of rank-4 affine factorization

The above proposed rank-4 affine factorization algorithm is summarized as follows.

**Algorithm 1: Rank-4 Affine Factorization**

```
   Input:  tracking matrix W
       1.  Perform SVD decomposition on the tracking matrix;
       2.  Obtain a set of rank-4 solutions of M̂ and Ŝ;
       3.  Recover the Euclidean upgrading matrix matrix H;
       4.  Compute the Euclidean structure and camera motion
parameters from H^{-1}Ŝ and M̂H.
   Output:  the Euclidean structure and motion parameters
```

## 7.4 Alternative and Weighted Factorization

SVD decomposition is a convenient way of structure and motion factorization, however, SVD only works when all features are tracked across the sequence. In practice, missing data are inevitable since some features may be missing during the process of tracking due to occlusion or other factors. Researchers proposed different alternative factorization approaches to handle missing data [33]. In this section, a two-step alternative and weighted factorization algorithms are introduced to handle missing data and image uncertainties.

### 7.4.1 Alternative factorization algorithm

The essence of structure and motion factorization (7.6) is equivalent to finding rank-4 matrices $\mathbf{M}$ and $\mathbf{S}$ by minimizing the following Frobenious norm.

$$\begin{aligned} &\underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}-\mathbf{M}\mathbf{S}\|_F^2 \\ &\text{subject to: } \mathbf{M}\in\mathbb{R}^{2m\times 4}, \mathbf{S}\in\mathbb{R}^{4\times n}. \end{aligned} \tag{7.15}$$

The basic idea of the two step factorization is to minimize the cost function (7.15) over $\mathbf{S}$ and $\mathbf{M}$ alternatively until convergence, while leaving the other one fixed, i.e.,

$$f(\mathbf{S}) = \underset{\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}-\mathbf{M}\mathbf{S}\|_F^2 \tag{7.16}$$

$$f(\mathbf{M}) = \underset{\mathbf{M}}{\operatorname{argmin}} \|\mathbf{W} - \mathbf{MS}\|_F^2. \tag{7.17}$$

Each cost function of the algorithm is indeed a convex function, so a global minimum can always be found. The algorithm converges fast if the tracking matrix is close to rank-4, even with a random initialization. This idea was adopted in several papers [33] [39] [41].

Different with SVD decomposition, the minimization process is carried out by least squares. Let us rewrite the cost function (7.16) with respect to each feature as follows.

$$f(\mathbf{s}_j) = \underset{\mathbf{s}_j}{\operatorname{argmin}} \|\mathbf{w}_j - \mathbf{M}\mathbf{s}_j\|_F^2 \tag{7.18}$$

where $\mathbf{w}_j$ is the $j$-th column of $\mathbf{W}$, and $\mathbf{s}_j$ is the $j$-th column of $\mathbf{S}$. Thus, the least squares solution of $\mathbf{S}$ can be given as

$$\mathbf{s}_j = \mathbf{M}^\dagger \mathbf{w}_j,\ j = 1, \cdots, n \tag{7.19}$$

where $\mathbf{M}^\dagger = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ is the Moore-Penrose pseudoinverse of matrix $\mathbf{M}$. The solution (7.19) can easily handle the missing data in the tracking matrix. For example, if some entries in $\mathbf{w}_j$ are unavailable, one can simply delete those elements in $\mathbf{w}_j$ and the corresponding columns in $\mathbf{M}^\dagger$, or just set those entries in $\mathbf{w}_j$ as zeros, then, $\mathbf{s}_j$ can be solved from (7.19) using the available entries via least squares.

Similarly, the second cost function (7.17) can be rewrite as

$$f(\mathbf{m}_i^T) = \underset{\mathbf{m}_i^T}{\operatorname{argmin}} \|\mathbf{w}_i^T - \mathbf{m}_i^T\mathbf{S}\|_F^2 \tag{7.20}$$

which yields the following least-square solution of the motion matrix $\mathbf{M}$.

$$\mathbf{m}_i^T = \mathbf{w}_i^T\mathbf{S}^\dagger,\ i = 1, \cdots, m \tag{7.21}$$

where the pseudoinverse $\mathbf{S}^\dagger = \mathbf{S}^T(\mathbf{S}\mathbf{S}^T)^{-1}$, $\mathbf{m}_i^T$ and $\mathbf{w}_i^T$ denote the $i$-th row of the matrices $\mathbf{M}$ and $\mathbf{W}$, respectively. In case of missing elements, one can simply reset those entries in $\mathbf{w}_i^T$ as zeros.

The alternative algorithm is summarized as follows.

**Algorithm 2: Alternative Factorization Algorithm**

```
Input:  matrix W and matrix M
Repeat the following two steps until convergence
    1.  Updata shape matrix via (7.19)
    2.  Updata motion matrix via (7.21)
Output:  matrices M and S
```

### 7.4.2 Alternative weighted factorization

Measurement errors are inevitable in the process of feature detection and tracking. If prior knowledge about distribution of the errors is available, then all elements of the approximation error can be weighted by taking account of the error distribution so as to increase the robustness and accuracy of the algorithm. The basic idea is to give each image measurement a weight according to its uncertainty. Reliable features get higher weights, while unreliable features are assigned lower weights. The weighted rank-4 factorization is formulated as follows.

$$\begin{aligned} &\underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\boldsymbol{\Sigma} \otimes (\mathbf{W} - \mathbf{M}\mathbf{S})\|_F^2 \\ &\text{subject to: } \mathbf{M} \in \mathbb{R}^{2m\times 4}, \mathbf{S} \in \mathbb{R}^{4\times n} \end{aligned} \tag{7.22}$$

where $'\otimes'$ denotes the Hadamard product, which is an element-by-element multiplication of two matrices; $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ is an $2m \times n$ uncertainty matrix whose entries are weights derived from the confidence of the image measurements.

The general weighted factorization could not be solved analytically in terms of the singular value decomposition. Many researchers have proposed different schemes to solve the problem [3][7][100]. In this section, the solution of (7.22) will be obtained using the alternative factorization algorithm by solving $\mathbf{S}$ and $\mathbf{M}$ alternatively as follows.

$$f(\mathbf{S}) = \underset{\mathbf{s}_j}{\operatorname{argmin}} \|\boldsymbol{\Sigma}_j \otimes (\mathbf{w}_j - \mathbf{M}\mathbf{s}_j)\|_F^2 \tag{7.23}$$

$$f(\mathbf{M}) = \underset{\mathbf{m}_i^T}{\operatorname{argmin}} \|\boldsymbol{\Sigma}_i^T \otimes (\mathbf{w}_i^T - \mathbf{m}_i^T\mathbf{S})\|_F^2 \tag{7.24}$$

where $\boldsymbol{\Sigma}_j$ denotes the $j$-th column of $\boldsymbol{\Sigma}$, and $\boldsymbol{\Sigma}_i^T$ the $i$-th row of $\boldsymbol{\Sigma}$. The close-form solutions of the shape and motion matrices can be obtained by least squares.

$$\mathbf{s}_j = \left(\operatorname{diag}(\boldsymbol{\Sigma}_j)\mathbf{M}\right)^{\dagger}\left(\operatorname{diag}(\boldsymbol{\Sigma}_j)\mathbf{w}_j\right),\ j = 1,\cdots,n \tag{7.25}$$

$$\mathbf{m}_i^T = \left(\mathbf{w}_i^T \operatorname{diag}(\boldsymbol{\Sigma}_i^T)\right)\left(\mathbf{S}\operatorname{diag}(\boldsymbol{\Sigma}_i^T)\right)^{\dagger},\ i = 1,\cdots,m \tag{7.26}$$

where $(\bullet)^{\dagger}$ denotes the pseudoinverse of a matrix, and $'\operatorname{diag}(\bullet)'$ stands for the diagonal matrix generated by a vector. Equations (7.25) and (7.26) give the least-square solutions of $\mathbf{S}$ and $\mathbf{M}$. Similar to the alternative factorization, when there are any missing data in the tracking matrix, one can simply set those entries in $\mathbf{w}_j$ as zeros.

The alternative weighted factorization algorithm is summarized as follows, where the initial values of the motion matrix $\mathbf{M}$ and weight matrix $\boldsymbol{\Sigma}$ will be discussed in next section.

**Algorithm 3: Alternative Weighted Factorization**

```
Input:  matrices W, M, and Σ
Repeat the following two steps until convergence
    1.  Updata shape matrix S via (7.25)
    2.  Updata motion matrix M via (7.26)
Output:  matrices M and S
```

## 7.5 Outlier Detection and Robust Factorization

Based on the rank-4 factorization algorithm proposed in the foregoing sections. A fast and practical scheme for outlier detection is proposed in this section.

### 7.5.1 Outlier detection scheme

Outlying data are inevitable in the process of feature tracking. The most popular strategy in computer vision field is RANSAC (RANdom SAmple Consensus) [26], Least Median of Squares (LMedS), and some other similar techniques in the family [14]. However, all these methods are computationally intensive. Another class of robust algorithm that can work with outliers is based on $L1$ factorization by minimizing the $L1$ norm of the reprojection errors [41].

The RANSAC approach initiates a model from a minimal set of data and then scores the model by the number of feature points within a given threshold. It can deal with data having more than 50% outliers. However, since the inlying data are also contaminated by noise, the initiated model using minimal data may be biased.

Both the SVD-based and the alternative factorization-based algorithms yield a set of least-square solutions. The best fit model is obtained by minimizing the sum of the squared residuals between the observed data and the fitted values provided by the model. Extensive experiments show that the least-square algorithms usually yield reasonable solutions even in the presence of certain amount of outliers and the reprojection residuals of these outlying data are usually much larger than those associated with inliers.

Suppose $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ are a set of solutions of the rank-4 factorization of a tracking matrix $\mathbf{W}$, the reprojection residuals can be computed by reprojecting the set of

solutions back onto all images. Let us define a residual matrix as follows.

$$\mathbf{E} = \mathbf{W} - \hat{\mathbf{M}}\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{e}_{11} & \cdots & \mathbf{e}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{e}_{m1} & \cdots & \mathbf{e}_{mn} \end{bmatrix}_{2m \times n} \tag{7.27}$$

where

$$\mathbf{e}_{ij} = \bar{\mathbf{x}}_{ij} - \hat{\mathbf{M}}_i \hat{\mathbf{s}}_j = \begin{bmatrix} \Delta u_{ij} \\ \Delta v_{ij} \end{bmatrix} \tag{7.28}$$

is the residual of point $(i, j)$ in both directions. The reprojection error of the point is defined by the Euclidean distance $\|\mathbf{e}_{ij}\|$ of the image point and its reprojection, and the reprojection error of the entire sequence is defined by an $m \times n$ matrix

$$\mathbf{Err} = \{\|\mathbf{e}_{ij}\|\}_{m \times n}. \tag{7.29}$$

Figure 7.1 shows an example of the distribution of the error matrix (7.29), where 40 images of 100 random space points were generated via the affine projection. The image resolution is $800 \times 800$, and the images are corrupted by Gaussian noise and 10% outliers. The added noise level is 3-unit, and the outliers are also simulated by Gaussian noise whose magnitude is set at 15-unit. The real added noise and outliers are illustrated by an image as shown in Figure 7.1 (a), where the grayscale of each pixel corresponds to the inverse magnitude of the error on that point, the darker the pixel, the larger the error magnitude on that point. The distribution of the real added outliers is depicted as a binary image in Figure 7.1 (c), which corresponds to the darker points in Figure 7.1 (a).

Using the corrupted data, a set of motion and shape matrices were estimated by employing the rank-4 factorization algorithm and the error matrix was then computed. The distribution of the reprojection error (7.29) is illustrated in Figure 7.1 (b) with each pixel corresponds to the reprojection error of that point. It is evident that the reprojection error and the real added noise have a similar distribution. The points with large reprojection errors correspond to those with large noise levels. Figure 7.1 (d) shows the binary image of Figure 7.1 (b) by simply applying a global threshold to the reprojected residuals. Surprisingly, almost all outliers are successfully detected by a single threshold. The distribution of false positive error (the inlier points being classified as outliers by a given threshold) and the false negative error (the outliers not being detected by thresholding) are given in Figure 7.1 (e) and (f), respectively.

Inspired by this observation, an intuitive outlier detection and robust factorization scheme is proposed. The flowchart of the strategy is shown in Figure 7.2, and the computation details is given as follows.

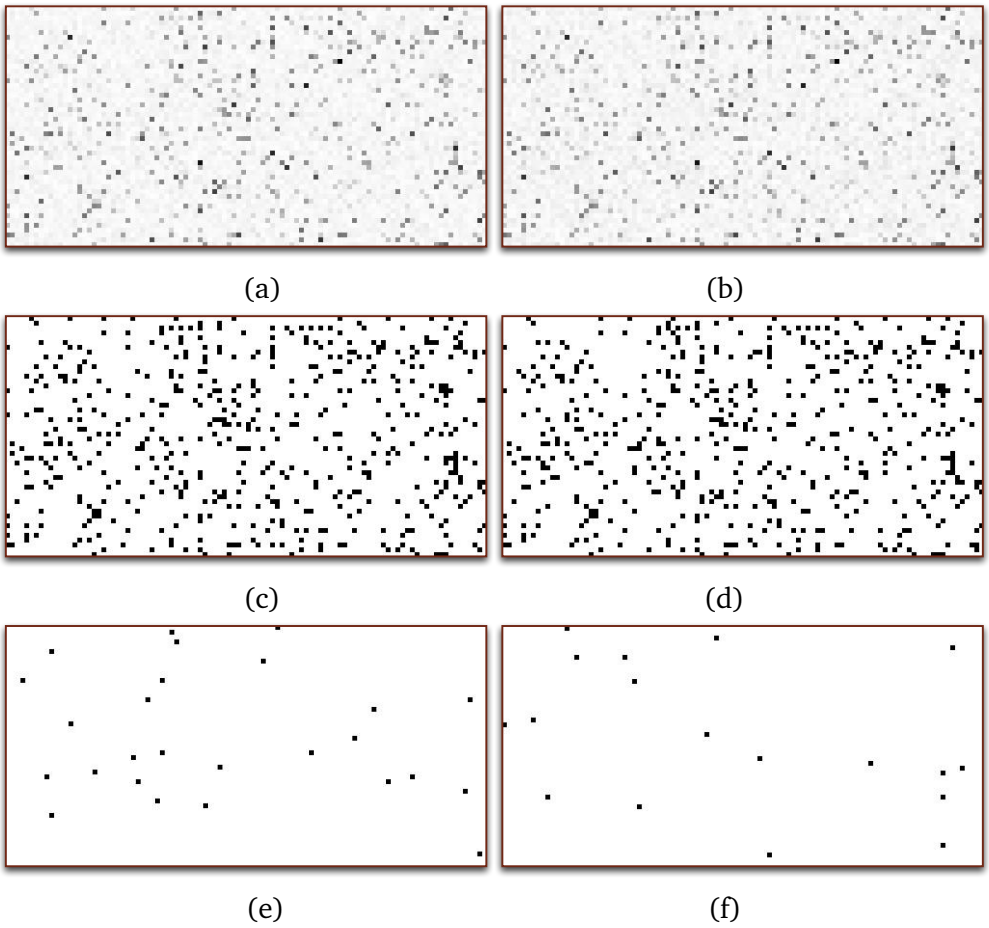



Figure 7.1: (a) Illustration of the normalized absolute value of the real added Gaussian noise and outliers in the tracking matrix, where the intensity of each pixel corresponds the error magnitude at that point; (b) the normalized value of the reprojection error (7.29); (c) the distribution of the added outlying data; (d) the outliers segmented from reprojection error by a single threshold; (e) the distribution of false positive error given by thresholding; and (f) the false negative error given by thresholding.

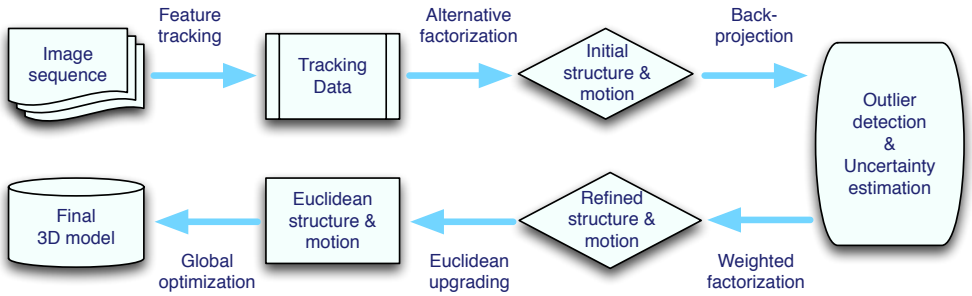



Figure 7.2: The outline and flowchart of the robust structure from motion

---

**Algorithm 4: Robust Structure and Motion Factorization**

---

Input: the tracking matrix $\mathbf{W}$

1. Balance the tracking matrix via point-wise and image-wise rescalings, as in [71], to improve numerical stability.

2. Perform rank-4 affine factorization on the tracking matrix to obtain a set of solutions of $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$.

3. Estimate the reprojection residuals and determine the outlier threshold.

4. Eliminate the outliers and recalculate the matrices $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ using the inlying data via Algorithm 2.

5. Estimate the uncertainty of each inlying feature.

6. Refine the solutions by weighted factorization algorithm.

7. Recover the Euclidean upgrading matrix $\mathbf{H}$ and upgrade the solutions to the Euclidean space as $\mathbf{M} = \hat{\mathbf{M}}\mathbf{H}$ and $\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}}$.

8. Perform a global optimization via bundle adjustment if necessary.

Output: the 3D structure and camera motion parameters recovered from $\mathbf{S}$ and $\mathbf{M}$ respectively.

---

In the above algorithm, steps 3 and 4 can be repeated once to ensure a more refined inlying data and solutions, but in practice, the repetition does not make much difference to final results. During computation, Algorithms 2 and 3 are employed to deal with missing data. Normally, the alternative algorithm can work with random initialization, however, since an initial set of solutions have been obtained in the previous steps, these solutions can be used as initial values in the iteration so as to speed up the convergence.

### 7.5.2 Parameter estimation

The image noise is usually modeled by a Gaussian distribution. It can be verified that the reprojection residuals (7.27) also follows the same distribution as the image noise. Suppose the noise at both coordinate directions in the image is independent and identically distributed (IID), and let $\mathcal{V}(\mathbf{E})$ be the $2mn$-dimensional vector formed by the residual matrix $\mathbf{E}$. Then, $\mathcal{V}(\mathbf{E})$ should also be a Gaussian as shown in Figure 7.3 (b).

Suppose $\mu$ and $\sigma$ are the mean and standard deviation of $\mathcal{V}(\mathbf{E})$, respectively. If the residual vector $\mathcal{V}(\mathbf{E})$ is registered with respect to its mean $\mu$. The outlier threshold is chosen as follows.

$$\theta = \kappa\,\sigma \tag{7.30}$$

where $\kappa$ is a parameter which can be set from 3.0 to 5.0. The points whose absolute values of the registered residuals in either direction, or the registered reprojection errors, are greater than $\theta$ will be classified as outliers.

$$\begin{aligned}\text{outliers} = \ \big\{\mathbf{x}_{i,j}\big|\ &|\Delta u_{ij} - \mu| > \theta \text{ or } |\Delta v_{ij} - \mu| > \theta \\ &\text{or } \big((\Delta u_{ij} - \mu)^2 + (\Delta v_{ij} - \mu)^2\big)^{\frac{1}{2}} > \theta\big\}.\end{aligned} \tag{7.31}$$

Since the residual vector $\mathcal{V}(\mathbf{E})$ contains outliers, which have signifycant influence to the estimation of the mean and standard deviation due to the large deviations of the outliers. In practice, the mean is estimated from the data that are less than the median value of $|\mathcal{V}(\mathbf{E})|$.

$$\mu = \text{mean}\{\mathcal{V}'(\mathbf{E})|\mathcal{V}'(\mathbf{E}) < \text{median}(|\mathcal{V}(\mathbf{E})|)\} \tag{7.32}$$

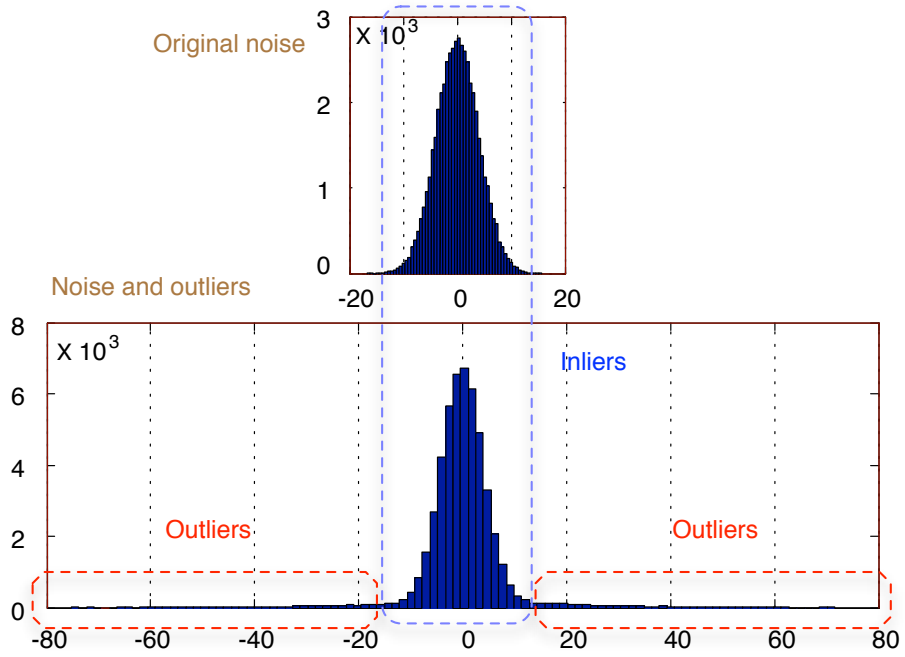
while the standard deviation is estimated from the median absolute deviation (MAD)

$$\sigma = 1.4826\,\text{median}(|\mathcal{V}(\mathbf{E}) - \text{median}(\mathcal{V}(\mathbf{E}))|) \tag{7.33}$$

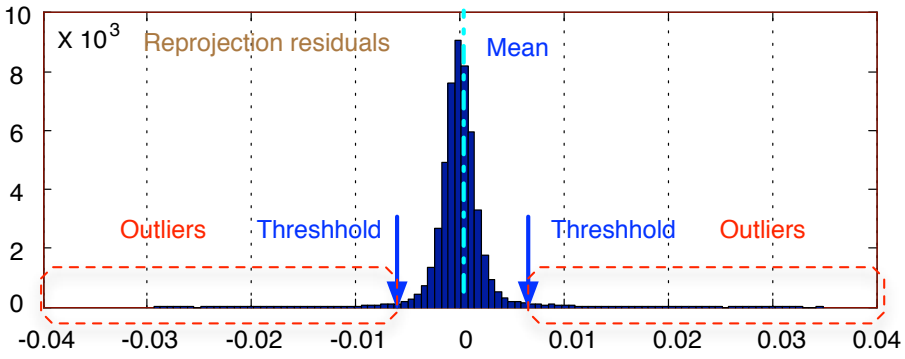
since the MAD is resistant to outliers. The above computation usually guarantees robust estimation of the mean and the standard deviation.

As for the weights in the weighted factorization, most researchers estimate it from the uncertainty of each feature based on the information such as sharpness and intensity contrast around its neighborhood. Some researchers modeled the errors isotropically with different variances [3][74]; others adopted directional distribution to describe the uncertainties [7][49]. The uncertainty is usually estimated during the process of feature detection and tracking or given as prior information. Nonetheless, this information is unavailable in many applications.

In our early study [89], it was shown that the uncertainty of each feature is generally proportional to the reprojection residual of that point. Thus, the




(a)




(b)

Figure 7.3: (a) The histogram distribution of the real added noise and outliers; (b) the histogram distribution of the reprojection residuals.

weight of each point can be estimated from the residual value obtained from the data after eliminating outliers. The point with higher residual value has larger uncertainty and thus gets lower weight. The weight of each point at each coordinate direction is determined as follows in a shape like normal distribution.

$$\omega_{ij} = \frac{1}{\mathcal{N}} \exp\left(-\frac{\mathbf{E}_{ij}^2}{2\sigma^2}\right) \tag{7.34}$$

where the standard deviation $\sigma$ is estimated from Equation (7.33), $\mathbf{E}_{ij}$ is the $(i, j)$-th element of the residual matrix (7.27), and $\mathcal{N}$ is a normalization scale. Clearly, one point may have different weights at the two image directions based on the residual value at each direction. For the missing data and outliers, the corresponding weights are set as $\omega_{ij} = 0$.

## 7.6 Evaluations on Synthetic Data

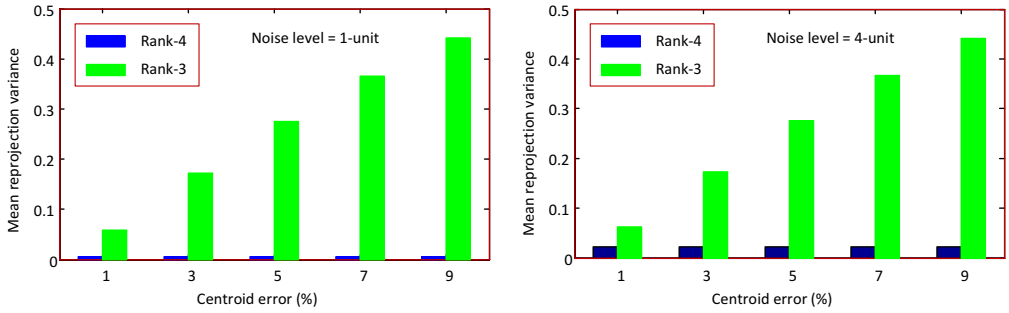
The proposed technique was validated and evaluated extensively on synthetic data and compared with other similar algorithms. During the simulation, 100 space points were randomly generated within a cube of $40 \times 40 \times 40$, and a sequence of 50 images were produced from these points by the affine projection. The following settings were used in the test: image resolution: $800 \times 800$ pixel; focal lengths varying randomly from 500 to 550; rotation angles randomly between $-60^\circ$ and $+60^\circ$; camera positions randomly inside a sphere with a diameter of 40; average distance from the cameras to the object: 600. These imaging conditions are very close to the assumption of the affine projection.

### 7.6.1 Influence of image centroid

The proposed rank-4 factorization algorithm was compared with its rank-3 counterpart with respect to different image centroid. For the generated images, Gaussian white noise was added to the images and the centroid of each image was deviated with a displacement, then, all imaged points were registered to the deviated centroid. This is a simulation of the situation when the centroid could not be reliably recovered due to the missing and outlying data.

Using the misaligned data, the motion and shape matrices were recovered by SVD factorization with the rank-4 and rank-3 constraints, then, the solution was reprojected back onto the images and the reprojection residuals was calculated. The performance of different algorithms were evaluated and compared by means of the mean reprojection variance, which is defined as:

$$E_{rv} = \frac{1}{mn} \|\mathbf{W}_0 - \hat{\mathbf{M}}\hat{\mathbf{S}}\|_F^2 \tag{7.35}$$




(a) (b)

Figure 7.4: The mean reprojection variance with respect to different centroid displacements at the noise level of (a) 1-unit and (b) 4-unit.

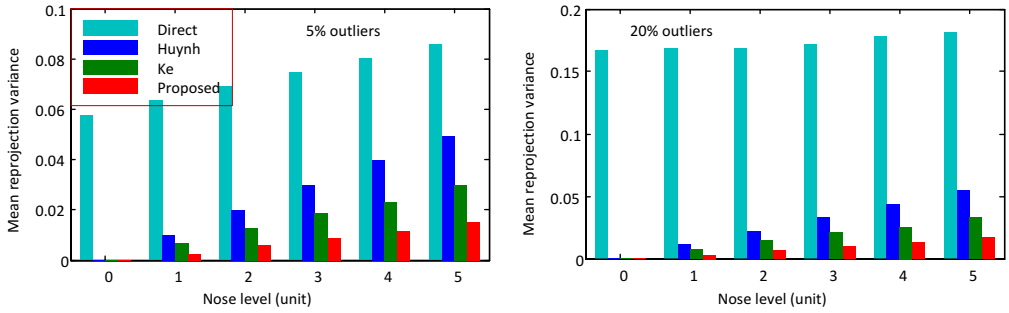
where $\mathbf{W}_0$ is the noise-free tracking matrix; and $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ are the estimated motion and shape matrices, respectively. In order to obtain a statistically meaningful comparison, 100 independent tests were performed at each noise level. The mean reprojection variance with respect to different centroid displacements at different noise levels is shown in Fig. 7.4, where the noise level is defined as the standard deviation of the Gaussian noise.

As shown in Fig. 7.4, it is evident that the miscalculated centroid has no influence on the proposed rank-4 factorization algorithm, however, the centroid errors have huge impact to the performance of the rank-3 based algorithm. The test shows that the influence caused by the centroid error is far more significant than that caused by the image noise. Thus, the rank-4 affine factorization approach is a better choice in practice, especially in the presence of missing data or large image noise.

### 7.6.2 Performance evaluation

In this test, the performance of the proposed robust factorization algorithm was evaluated and compared in terms of accuracy and computational complexity.

For the above generated image sequence, Gaussian noise was added to each image point and the noise level was varied from 1-unit to 5-unit in steps of 1. In the mean time, some outliers were added to the tracking matrix. Using the contaminated data, the foregoing proposed robust algorithm was employed to recover the motion and shape matrices. The mean reprojection variance at different noise levels and outliers ratios is shown in Fig. 7.5. As a comparison, two successful algorithms in the literature were implemented as well, one is an outlier correction scheme proposed by Huynh *et al.* [39], the other one is proposed by Ke and kanade [41] based on minimization of the $L1$ norm.




(a) (b)

Figure 7.5: The mean reprojection variance at different noise levels with (a) 5% outliers and (b) 20% outliers.

Table 7.1: Real computation time of different algorithms (unit: second).

| Frame no. | 50 | 100 | 150 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|
| Huynh | 1.19 | 2.35 | 3.68 | 6.41 | 10.45 | 12.69 |
| Ke | 1.81 | 6.27 | 14.32 | 26.94 | 44.87 | 67.53 |
| Proposed | 1.27 | 3.93 | 8.12 | 14.28 | 22.40 | 32.13 |

The results in Fig. 7.5 were evaluated by 100 independent tests, where 'Direct' stands for normal rank-4 factorization algorithm without outlier rejection. Here the reprojection variance was estimated only using the original inlying data by eliminating the added outliers. It is obvious that the proposed scheme outperforms other algorithms in terms of accuracy. The direct factorization algorithm yields significantly large errors due to the influence of outliers, and the errors increase with the increase of outliers. The experiment also shows that all three robust algorithms are resilient to outliers, as can be see in Fig. 7.5, the ratio of outliers has little influence to the reprojection variance of the three robust algorithms.

The complexity of different approaches was compared in terms of real computation time in the above test. All algorithms were implemented in Matlab 2009 on a Lenovo T500 laptop with 2,26GHz Intel Core Duo CPU. The frame number was varied from 50 to 300 in steps of 50 so as to generate different sizes of the tracking matrix, and 10% outliers were added in the data. Table 8.1 shows the real computation time of different algorithm. Obviously, the complexity of the proposed scheme lies in between of [39] and [41]. This is because Huynh's method does not include the weighted factorization step, while the minimization of $L1$ norm in [41] is computation more intensive than the alternative factorization algorithm.

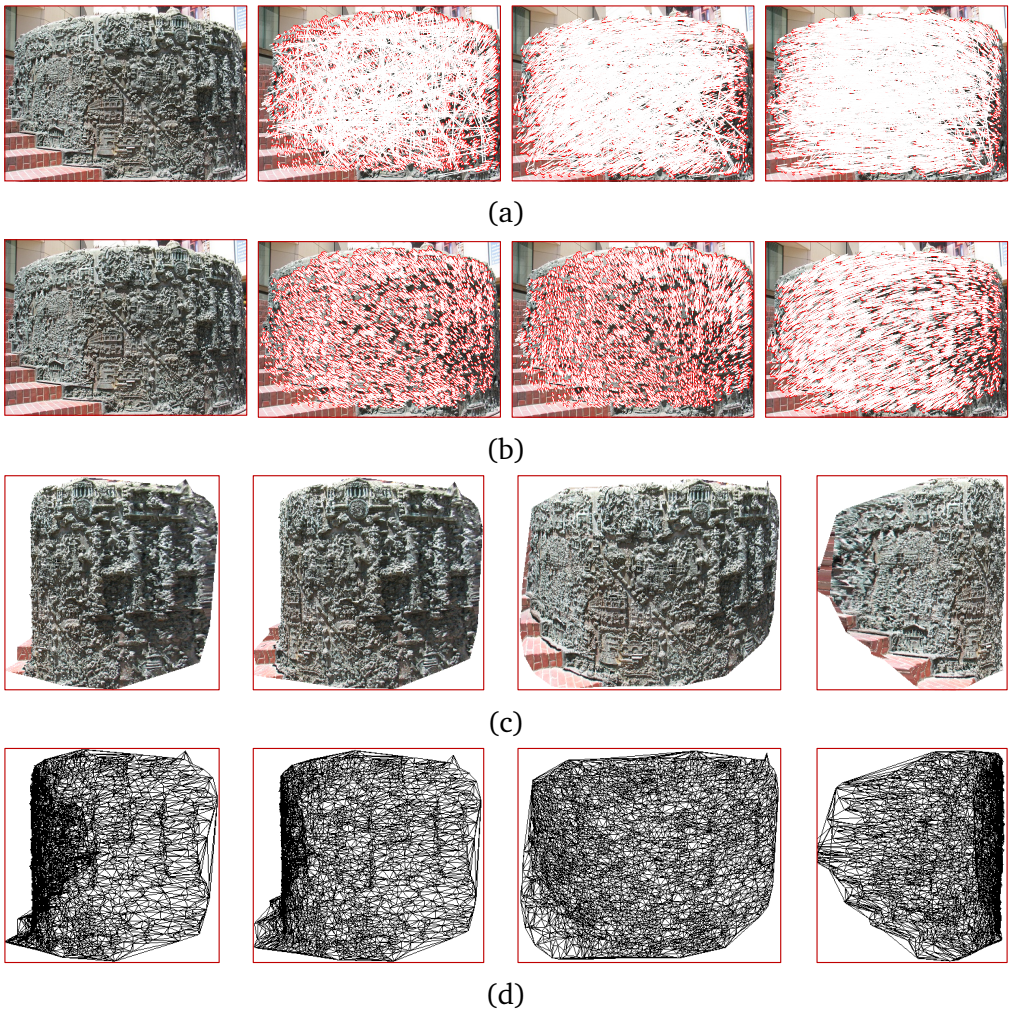
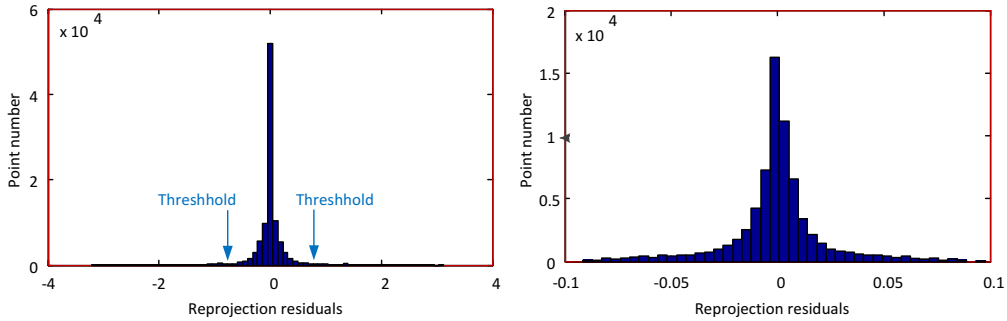


Figure 7.6: Reconstruction results of the fountain base sequence. (a) Four frames from the sequence with the tracking data and 5% outliers overlaid on three images; (b) the correctly detected inlying tracking data; (c) the reconstructed VRML model shown from different viewpoints; and (d) the corresponding triangulated wireframe of the model.




(a) (b)

Figure 7.7: The histogram distribution of the residual matrix of the fountain base sequence before (a) and after (b) outlier rejection.

## 7.7 Evaluations on Real Sequences

The method was tested on many real image sequences. The results on two data sets are reported here. The resolution of each image is $1024 \times 768$, and the feature correspondences are established by the feature tracking system based on SIFT and epipolar constraints [80].

The first test is on a fountain base sequence captured at downtown San Francisco. The sequence consists of 10 images and 5648 features were tracked across the sequence. Since the images have a large portion of homogeneous and repetitive textures, it is thus hard to track accurately for this type of scene, but visually only a few features were obviously mismatched. In order to test the algorithm, an additional 5% outliers were added to the tracking data, as shown in Figure 7.6 (a). After rejecting the outliers by the proposed approach, the remaining inlying features with disparities to the first image are shown in Figure 7.6 (b), the outliers were successfully rejected. Then, the alternative weighted algorithm was employed to recover the motion and structure matrices. Finally, the solution was upgraded to the Euclidean space. As shown in Figure 7.6 (c) and (d), the structure looks realistic and most details are correctly recovered.

The histogram distribution of the reprojection residual matrix with outliers is shown in Figure 7.7 (a). The residuals are largely conform to the assumption of normal distribution. It can be seen from the distribution that the outliers can be obviously distinguished from inliers, the computed threshold is shown in the figure. The histogram distribution of the residuals of the detected inlying data is shown in Figure 7.7 (b). Obviously, the residual error was reduced significantly after rejecting the outliers. The final reprojection error is 0.426, as a comparison, the reprojection errors by the algorithms of 'Huynh' and 'Ke' are 0.736 and 0.597,

respectively. The proposed scheme yields the lowest reprojection error.

The second sequence is a corner of the Hearst Gym at UC Berkeley. There are 12 images in the sequence, and 1890 features were tracked in total. The tracking data with 5% outliers are shown in Figure 7.8 (a); and the correctly detected inlying features are shown in Figure 7.8 (b). The proposed robust algorithm was used to recover the Euclidean structure of the scene. Figure 7.8 (c) and (d) show the reconstructed VRML model with texture mapping and the corresponding triangulated wireframes from different viewpoints. The model is correctly recovered by the proposed method. The reprojection errors obtained from 'Huynh', 'Ke', and the proposed algorithm are 0.742, 0.635, and 0.508, respectively. The proposed approach outperforms other robust algorithms.

## 7.8 Conclusion

The chapter first proposed an augmented rank-4 factorization algorithm which has been proved to be more accurate and more widely applicable than classical rank-3 affine factorization, especially in the case when the centroid of the feature points could not be reliably recovered due to the presence of missing and outlying data. Then, an alternatively weighted factorization algorithm was presented to alleviate the influence of large image noise. Finally, a robust factorization scheme was designed to deal with corrupted data containing outliers and missing points. The proposed technique requires no prior information of the error distribution in the tracking data. Extensive tests and evaluations demonstrated its advantages over previous methods. Although the proposed approach was demonstrated by factorization of rigid objects and static scenes, it can be applied directly to handle nonrigid factorization, as will be discussed in Chapter 8.

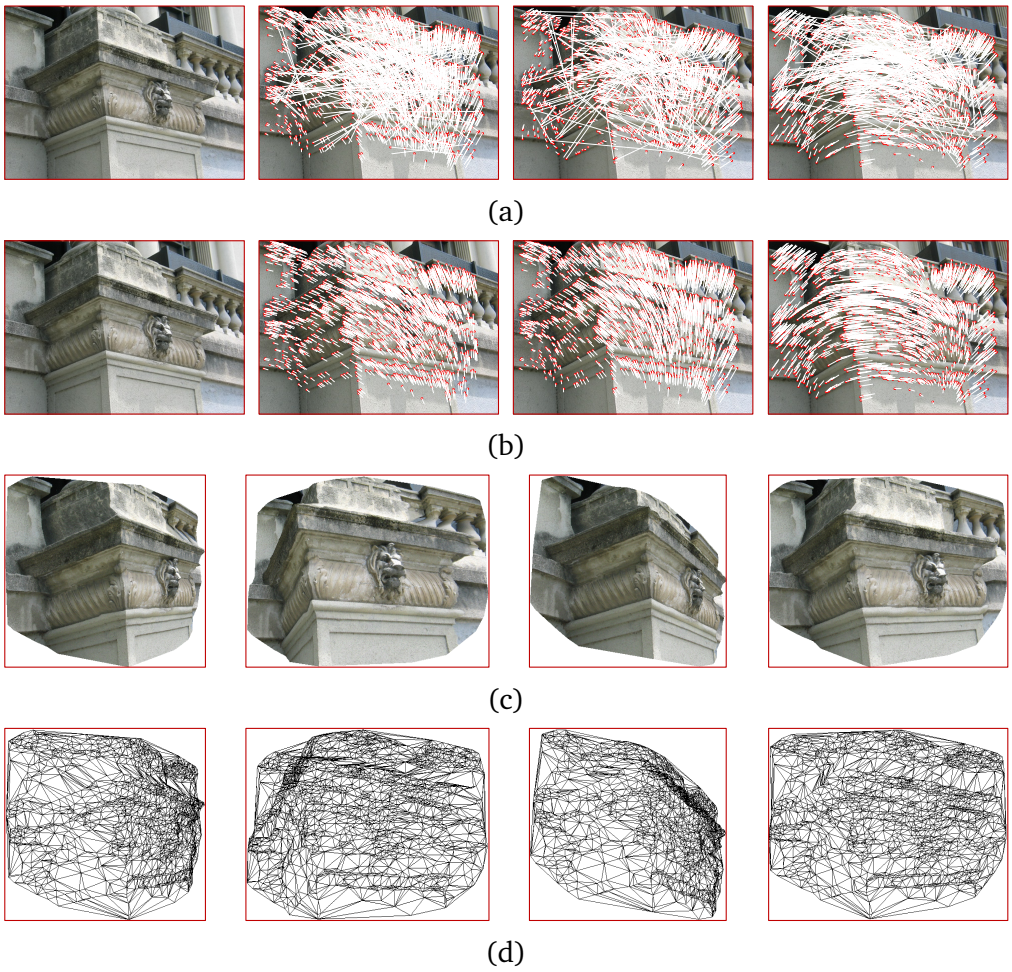


Figure 7.8: Reconstruction results of the Hearst Gym sequence. Four frames from the sequence with the tracking data and 5% outliers overlaid on three images; (b) the correctly detected inlying tracking data; (c) the reconstructed VRML model shown from different viewpoints; and (d) the corresponding triangulated wireframe of the model.

# Chapter 8

# Robust SfM of Nonrigid Objects

This chapter is an extension of the robust algorithm of Chapter 7 to nonrigid scenarios, and it is presented in a similar style. An augmented nonrigid factorization algorithm is first proposed to circumvent the difficulty of image alignment for imperfect data, followed by a robust scheme to handle outlying and missing data in nonrigid structure and motion recovery. The contributions of this chapter are as follows: (i) as a new addition to previous nonrigid structure from motion, the proposed factorization algorithm can perfectly handle imperfect tracking data; (ii) it is demonstrated that the outliers can be detected directly from image reprojection residuals of nonrigid factorization; and (iii) the robust factorization scheme is proved empirically to be more efficient and more accurate than other robust algorithms. The proposed approach is validated and evaluated by extensive experiments on synthetic data and real image sequences.

## 8.1 Introduction

The factorization algorithm was extended to nonrigid SfM by assuming that the 3D shape of a nonrigid object can be modeled as a weighted linear combination of a set of shape bases [11]. Thus, the shape bases and camera motions are factorized simultaneously for all time instants under a rank-3$k$ constraint of the tracking matrix. The method has been extensively investigated and developed in [9], [75], and [66].

In recent years, the problem of robust factorization has received a lot of attention [50] [103] [100]. Aguitar and Moura [3] proposed a scalar-weighted SVD algorithm that minimizes the weighted square errors. Gruber and Weiss [29] formulated the problem as a factor analysis and derived an Expectation Maximization (EM) algorithm to enhance the robustness to missing data and

uncertainties. Huynh *et al.* [39] proposed an iterative approach to correct the outliers with 'pseudo' observations. Ke and Kanade [41] designed a robust algorithm to handle outliers by minimizing a $L1$-norm of the reprojection errors. Wang *et al.* [89] proposed a spatial-and-temporal-weighted factorization approach to handle significant noise in the measurement. The problem is further studied and different robust algorithms have been proposed [23] [55] [99] [92] [60].

Most of the above robust algorithms are initially designed for SfM of rigid objects. To the best of our knowledge, few studies have been carried out for nonrigid scenarios. In this chapter, we will extend the robust factorization approach of Chapter 7 to the nonrigid scenarios.

This chapter first proposes an augmented rank-$(3k+1)$ factorization algorithm to circumvent the difficulty of image registration for imperfect data, followed by an alternative weighted factorization algorithm to handle the missing features and image uncertainty. Then, the outlying data are detected from a new viewpoint via image reprojection residuals by exploring the fact that the reprojection residuals are largely proportional to the measurement errors. Finally, a robust factorization scheme is proposed to deal with outliers. A preliminary work on this research was published at the CRV conference [90].

## 8.2 Background of Nonrigid Factorization

Under the affine projection model, a 3D point $\bar{\mathbf{X}}_j$ is projected onto $\bar{\mathbf{x}}_{ij}$ in frame $i$ according to the imaging equation

$$\bar{\mathbf{x}}_{ij} = \mathbf{A}_i \bar{\mathbf{X}}_j + \mathbf{c}_i \tag{8.1}$$

where $\mathbf{A}_i$ is a $2\times3$ affine projection matrix; and the translation term $\mathbf{c}_i$ is the image of the centroid of all space points. Let $\mathbf{S}_i = [\bar{\mathbf{X}}_{i1}, \cdots, \bar{\mathbf{X}}_{in}]$ be the 3D structure associated with frame $i$, the structure may be different at different instants. In nonrigid SfM, we usually follow Bregler's assumption that $\mathbf{S}_i = \sum_{l=1}^{k} \omega_{il}\mathbf{B}_l$, where the nonrigid structure is assumed to be a linear combination of a set of rigid shape bases $\mathbf{B}_l$ [11]. Under this assumption, the imaging process of one image can be modeled as

$$\begin{aligned}\mathbf{W}_i &= [\bar{\mathbf{x}}_{i1}, \cdots, \bar{\mathbf{x}}_{in}] = \mathbf{A}_i\mathbf{S}_i + [\mathbf{c}_i, \cdots, \mathbf{c}_i] \\ &= \left[\omega_{i1}\mathbf{A}_i, \cdots, \omega_{ik}\mathbf{A}_i\right] \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \end{bmatrix} + [\mathbf{c}_i, \cdots, \mathbf{c}_i].\end{aligned}$$

It is easy to verify that if all image points in each frame are registered to the centroid and relative image coordinates are employed, the translation term

vanishes, i.e., $\mathbf{c}_i = \mathbf{0}$. Consequently, the nonrigid factorization under affine camera model is expressed as

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \omega_{11}\mathbf{A}_1 & \cdots & \omega_{1k}\mathbf{A}_1 \\ \vdots & \ddots & \vdots \\ \omega_{m1}\mathbf{A}_m & \cdots & \omega_{mk}\mathbf{A}_m \end{bmatrix}}_{\mathbf{M}_{2m\times 3k}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \end{bmatrix}}_{\bar{\mathbf{B}}_{3k\times n}}. \tag{8.2}$$

Structure from motion is an ill-posed problem. Suppose the tracking matrix $\mathbf{W}$ is available, our purpose is to recover the camera motion parameters and the 3D structure from the shape matrix. It is obvious from (8.2) that the rank of the tracking matrix $\mathbf{W}$ is at most $3k$. Previous studies on nonrigid SfM are based on the rank-$3k$ constraint due to its simplicity, and the factorization can be easily obtained via SVD decomposition by truncating its rank to $3k$.

## 8.3 Augmented Affine Factorization

One critical condition for Equation (8.2) is that all image measurements are registered to the corresponding centroid of each frame. When the tracking matrix contains outliers and/or missing data, it is impossible to reliably retrieve the centroid. As will be shown in the experiments, the miscalculation of the centroids will cause significant errors to the final solutions. Previous studies were either ignoring this problem or hallucinating the missing points with pseudo observations, which may lead to a biased estimation. In this section, an augmented affine factorization algorithm based on rank-$(3k+1)$ constraint is proposed to solve this problem.

### 8.3.1 Rank-$(3k+1)$ affine factorization

Let us formulate the affine imaging process (8.1) in the following form

$$\mathbf{x}_{ij} = \left[\mathbf{A}_i | \mathbf{c}_i\right] \mathbf{X}_j \tag{8.3}$$

where $\mathbf{X}_j = [\bar{\mathbf{X}}_j^T, t_j]^T$ is a 4-dimensional homogeneous expression of $\bar{\mathbf{X}}_j$. Then, the imaging process of frame $i$ can be written as

$$\begin{aligned} \mathbf{W}_i = [\bar{\mathbf{x}}_{i1}, \cdots, \bar{\mathbf{x}}_{in}] &= \left[\mathbf{A}_i | \mathbf{c}_i\right] \begin{bmatrix} \sum_{l=1}^{k} \omega_{il}\mathbf{B}_l \\ \mathbf{t}_i^T \end{bmatrix} \\ &= \left[\omega_{i1}\mathbf{A}_i, \cdots, \omega_{ik}\mathbf{A}_i, \mathbf{c}_i\right] \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \\ \mathbf{t}_i^T \end{bmatrix}. \end{aligned}$$

Thus, the structure and motion factorization for the entire sequence is formulated as follows.

$$\underbrace{\begin{bmatrix} \bar{\mathbf{x}}_{11} & \cdots & \bar{\mathbf{x}}_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}_{m1} & \cdots & \bar{\mathbf{x}}_{mn} \end{bmatrix}}_{\mathbf{W}_{2m\times n}} = \underbrace{\begin{bmatrix} \omega_{11}\mathbf{A}_1 & \cdots & \omega_{1k}\mathbf{A}_1 & \mathbf{c}_1 \\ \vdots & \ddots & \vdots & \vdots \\ \omega_{m1}\mathbf{A}_m & \cdots & \omega_{mk}\mathbf{A}_m & \mathbf{c}_m \end{bmatrix}}_{\mathbf{M}_{2m\times(3k+1)}} \underbrace{\begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_k \\ \mathbf{t}_i^T \end{bmatrix}}_{\mathbf{B}_{(3k+1)\times n}}. \tag{8.4}$$

Obviously, the rank of the tracking matrix becomes $3k+1$ in this case. Given the tracking matrix, the factorization can be easily obtained via SVD decomposition with rank-$(3k+1)$ constraint. Unlike the affine factorization (8.2), the augmented expression (8.4) does not require any image registration. Thus, it can work with outlying and missing data.

Both factorization algorithms (8.2) and (8.4) can be equivalently denoted as the following minimization scheme.

$$f(\mathbf{M},\mathbf{S}) = \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W}-\mathbf{M}\mathbf{S}\|_F^2. \tag{8.5}$$

By enforcing different rank constraints, the Frobenius norm of (8.5) corresponding to the algorithms (8.2) and (8.4) would be

$$E_{3k} = \sum_{i=3k+1}^{N} \sigma_i^2, \;\; E_{3k+1} = \sum_{i=3k+2}^{N} \sigma_i^2 \tag{8.6}$$

where $\sigma_i, i = 1,\cdots,N$ are singular values of the tracking matrix in descending order, and $N = \min(2m,n)$. Clearly, the error difference by the two algorithm is $\sigma_{3k+1}^2$. For noise free data, if all image points are registered to the centroid, then, $\sigma_i = 0, \forall i > 3k$, the equations (8.2) and (8.4) are actually equivalent. However, in the presence of outlying and missing data, the image centroid could not be accurately recovered, the rank-$3k$ algorithm (8.2) will yield a big error since $\sigma_{3k+1}$ does not approach to zero in this situation.

### 8.3.2 Euclidean upgrading matrix

Suppose $\mathbf{W} = \hat{\mathbf{M}}\hat{\mathbf{B}}$ is a set of factorization result of (8.4). Obviously, the decomposition is not unique since it is only defined up to a nonsingular linear $\mathbf{H} \in \mathbb{R}^{(3k+1)\times(3k+1)}$ as $\mathbf{M} = \hat{\mathbf{M}}\mathbf{H}$ and $\mathbf{S} = \mathbf{H}^{-1}\hat{\mathbf{S}}$. The recovery of the upgrading matrix is different with that in the rank-$3k$ factorization.

Let us write the $(3k+1)\times(3k+1)$ upgrading matrix in the following form.

$$\mathbf{H} = [\mathbf{H}_1,\cdots,\mathbf{H}_k|\mathbf{h}_{3k+1}] \tag{8.7}$$

where $\mathbf{H}_l \in \mathbb{R}^{(3k+1)\times 3}(l = 1, \cdots, k)$ denotes the $l$-th triple columns of $\mathbf{H}$, and $\mathbf{h}_{3k+1}$ denotes the last column of $\mathbf{H}$. Suppose $\hat{\mathbf{M}}_i$ is the $i$-th two-row submatrix of $\hat{\mathbf{M}}$, then the upgraded motion matrix can be written as

$$\mathbf{M}_i = \hat{\mathbf{M}}_i\mathbf{H} = [\hat{\mathbf{M}}_i\mathbf{H}_1, \cdots, \hat{\mathbf{M}}_i\mathbf{H}_k|\hat{\mathbf{M}}_i\mathbf{h}_{3k+1}]. \tag{8.8}$$

Comparing the above equation with (8.4), we have

$$\hat{\mathbf{M}}_i\mathbf{H}_l = \omega_{il}\mathbf{A}_i = \omega_{il}f_i \begin{bmatrix} \mathbf{r}_{i1}^T \\ \mathbf{r}_{i2}^T \end{bmatrix} \tag{8.9}$$

where $f_i$ is the focal length of the camera, $\mathbf{r}_{i1}^T$ and $\mathbf{r}_{i2}^T$ are the first two rows of the rotation matrix. Denote $\mathbf{Q}_l = \mathbf{H}_l\mathbf{H}_l^T$, then, $\mathbf{Q}_l$ can be constrained from (8.9) as

$$\hat{\mathbf{M}}_i\mathbf{Q}_l\hat{\mathbf{M}}_i^T = \left(\hat{\mathbf{M}}_i\mathbf{H}_l\right)\left(\hat{\mathbf{M}}_i\mathbf{H}_l\right)^T = \omega_{il}^2 f_i^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{8.10}$$

The matrix $\mathbf{Q}_l$ has $9k$ degree-of-freedom since it is a $(3k+1)\times(3k+1)$ positive semidefinite symmetric matrix defined up to a scale. The above equation provides two constraints to $\mathbf{Q}_l$, thus, it can be linearly solved via least squares by stacking the constraints (8.10) frame by frame. Furthermore, the submatrix $\mathbf{H}_l$ can be decomposed from $\mathbf{Q}_l$ via extended Cholesky decomposition [86].

From equations (8.4) and (8.8), it is easy to prove that the last column of the upgrading matrix $\mathbf{h}_{3k+1}$ only influences the translation from the world coordinate system to the image system. Under a given coordinate system, different values of $\mathbf{h}_{3k+1}$ will only alter the origin of the world system, however, it does not change the Euclidean structure of the reconstructed points. Therefore, $\mathbf{h}_{3k+1}$ can be set freely as any $(3k+1)$-dimensional vector that is independent of the columns of $\{\mathbf{H}_l, l = 1, \cdots, k\}$ such that the resulted upgrading matrix is nonsingular.

After recovering the Euclidean upgrading matrix, the camera parameters, motions, shape bases, and deformation weights can be easily decomposed from the upgraded motion matrix $\hat{\mathbf{M}}\mathbf{H}$ and shape matrix $\mathbf{H}^{-1}\hat{\mathbf{B}}$.

### 8.3.3 Alternative factorization with missing data

SVD decomposition could not directly work with missing data. If some features are missing in the tracking data, we can adopt a similar alternative factorization algorithm as in Chapter 7. The basic idea is to minimize the cost function (8.5) over $\mathbf{S}$ and $\mathbf{M}$ alternatively until convergence, while leaving the other one fixed at each step, i.e.,

$$f(\mathbf{S}) = \underset{\mathbf{S}}{\operatorname{argmin}} \|\mathbf{W} - \mathbf{M}\mathbf{S}\|_F^2 \tag{8.11}$$

$$f(\mathbf{M}) = \underset{\mathbf{M}}{\operatorname{argmin}} \|\mathbf{W} - \mathbf{MS}\|_F^2. \tag{8.12}$$

As discussed in Chapter 7, each cost function of the algorithm is indeed a convex function thus a global minimum can be found. The algorithm converges very fast if the tracking matrix is close to rank-$(3k+1)$ even with a random initialization.

In the alternative factorization, the minimization process of each cost function is carried out by least squares. Thus, in case of missing elements, one can simply eliminate those entries in the equation and still obtain a least-square solution of the motion and structure matrices.

### 8.3.4 Alternative weighted factorization

When the uncertainty of image measurement is available, a weighted factorization strategy can be adopted to alleviate the influence of unreliable features. The weighted factorization is formulated as follows.

$$f(\mathbf{M}, \mathbf{S}) = \underset{\mathbf{M},\mathbf{S}}{\operatorname{argmin}} \|\boldsymbol{\Sigma} \otimes (\mathbf{W} - \mathbf{MS})\|_F^2 \tag{8.13}$$

where $'\otimes'$ denotes the Hadamard product of element-by-element multiplication; and $\boldsymbol{\Sigma} = \{\sigma_{ij}\}$ is the weight matrix whose entries are derived from the confidence of the image measurements. Similar to Chapter 7, the solution of (8.13) is obtained using the alternative factorization algorithm.

$$f(\mathbf{S}) = \underset{\mathbf{s}_j}{\operatorname{argmin}} \left\|\boldsymbol{\Sigma}_j \otimes (\mathbf{w}_j - \mathbf{M}\mathbf{s}_j)\right\|_F^2 \tag{8.14}$$

$$f(\mathbf{M}) = \underset{\mathbf{m}_i^T}{\operatorname{argmin}} \left\|\boldsymbol{\Sigma}_i^T \otimes (\mathbf{w}_i^T - \mathbf{m}_i^T \mathbf{S})\right\|_F^2 \tag{8.15}$$

where $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\Sigma}_i^T$ denote the $j$-th column and $i$-th row of $\boldsymbol{\Sigma}$, respectively. The algorithm alternatively updates the shape and motion matrices using least squares until convenience. In case of missing data, one can simply delete the corresponding elements in the equations.

## 8.4 Outlier Detection and Robust Factorization

Based on the foregoing proposed factorization algorithm, a fast and practical scheme for outlier detection is discussed in this section.

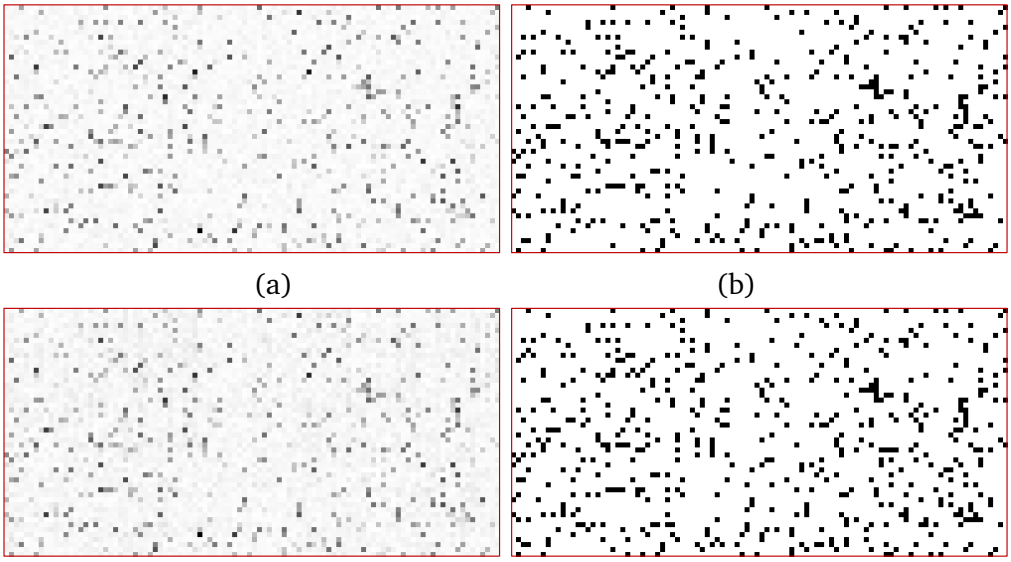


Figure 8.1: (a) Illustration of the normalized absolute value of the added Gaussian noise, where the intensity of each pixel corresponds the error magnitude at that point; (b) the distribution of the real added outliers; (c) the normalized value of the reprojection errors; and (d) the outliers segmented from the reprojection errors by applying a single threshold. Only 50 frames of 100 points are showed in the image.

### 8.4.1 Outlier detection scheme

The best fit model of the factorization algorithm is obtained by minimizing the sum of squared residuals between the observed data and the fitted values provided by the model. Extensive empirical studies show that the least-square solutions are usually reasonable even in the presence of certain amount of outliers, and the reprojection residuals of the outlying data are significantly larger than those associated with inliers.

Suppose $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ are a set of initial solutions of the motion and structure matrices, the reprojection residuals can be computed by reprojecting the solutions back onto all images. Let us define a residual matrix as:

$$\mathbf{E} = \mathbf{W} - \hat{\mathbf{M}}\hat{\mathbf{S}} = \begin{bmatrix} \mathbf{e}_{11} & \cdots & \mathbf{e}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{e}_{m1} & \cdots & \mathbf{e}_{mn} \end{bmatrix}_{2m \times n} \tag{8.16}$$

where

$$\mathbf{e}_{ij} = \bar{\mathbf{x}}_{ij} - \hat{\mathbf{M}}_i \hat{\mathbf{s}}_j = \begin{bmatrix} \Delta u_{ij} \\ \Delta v_{ij} \end{bmatrix} \tag{8.17}$$

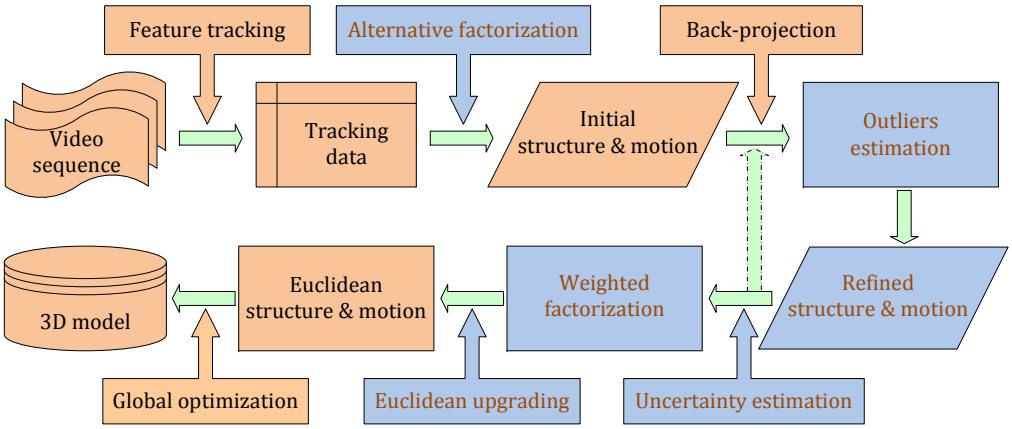



Figure 8.2: The outline and flowchart of the proposed approach

is the residual of point $(i, j)$ at both image directions. The reprojection error of a point is defined as $\|\mathbf{e}_{ij}\|$, which is the Euclidean norm of the residual at that point.

Below is an example of the residual matrix and reprojection errors. Using the synthetic data in Section 8.5, we added 3-unit Gaussian noise and 10% outliers (defined by significant noise greater than 5-unit) to the images. Then, the reprojection error is estimated via back-projection of the solutions from rank-$(3k+1)$ factorization.

The real added noise and the reprojection error are illustrated in Figure 8.1 as grayscale images, where the gray level of each pixel corresponds to the inverse magnitude of the error on that point, lower gray level (black points) stands for larger error. It is evident that the reprojection error and the added noise have similar distribution. The ground truth of the real added outliers is depicted as a binary image in Figure 8.1 (b). Figure 8.1 (d) shows the binarized image of the reprojection errors by simply applying a global threshold. Surprisingly, almost all outliers are successfully detected by a single threshold.

### 8.4.2 Implementation details

Inspired by the above observation, an efficient outlier detection and robust factorization scheme is developed based on the reprojection residuals. The outline of the proposed scheme is shown in Fig. 8.2, where the blue boxes indicate the contributions of this chapter. The computational details of the proposed scheme are as follows.

**Robust Factorization Algorithm**

**Input:** Tracking matrix of the sequence

1. Perform rank-$(3k+1)$ factorization to obtain an initial solutions of $\hat{\mathbf{M}}$ and $\hat{\mathbf{B}}$.
2. Estimate the reprojection residuals (8.16) from initial solutions.
3. Determine an outlier threshold and eliminate the outliers.
4. Refine the solutions from the inliers via the alternative factorization algorithm.
5. Estimate the weight matrix $\boldsymbol{\Sigma}$ from the refined solutions.
6. Perform the weighted factorization using the inliers.
7. Recover the upgrading matrix $\mathbf{H}$ and upgrade the solutions to the Euclidean space.
8. Recover the Euclidean structure and motion parameters corresponding to each frame.

**Output:** 3D structure and camera motion parameters

The alternative factorization algorithm is employed in steps 4 and 6 to handle missing data, while the initial values are obtained from the previous steps. Although the alternative factorization can work with random initialization, a reliable initial values can speed up its convergence.

Two important parameters are required in the robust algorithm: one is the outlier threshold, the other is the weight matrix. The following will discuss how to recover these parameters.

### 8.4.3 Parameter estimation

Assuming Gaussian image noise, it is easy to prove that the reprojection residuals also follow the same distribution as that of the image noise. Figure 8.4 shows an example from the synthetic data in Section 8.5. 3-unit Gaussian noise and 10% outliers were added to the synthetic images, and the residual matrix was calculated from (8.16). As shown in Figure 8.4, the residuals are obviously follow Gaussian distribution. Thus, the outlier threshold can be determined from the distribution of the residuals.

Let $\mathcal{V}(\mathbf{E})$ be a $2mn$-vector formed by the residual matrix $\mathbf{E}$, suppose $\mu$ and $\sigma$ are the mean and standard deviation of $\mathcal{V}(\mathbf{E})$, then the outlier threshold can be chosen as follows.

$$\theta = \kappa\,\sigma \tag{8.18}$$

where $\kappa$ is a parameter which is usually set from 3.0 to 5.0. Register $\mathcal{V}(\mathbf{E})$ with respect to its mean $\mu$, then the outliers are classified via the following criteria.

$$|\Delta u_{ij} - \mu| > \theta \text{ or } |\Delta v_{ij} - \mu| > \theta \text{ or } \left\| \begin{matrix} \Delta u_{ij} - \mu \\ \Delta v_{ij} - \mu \end{matrix} \right\| > \theta. \tag{8.19}$$

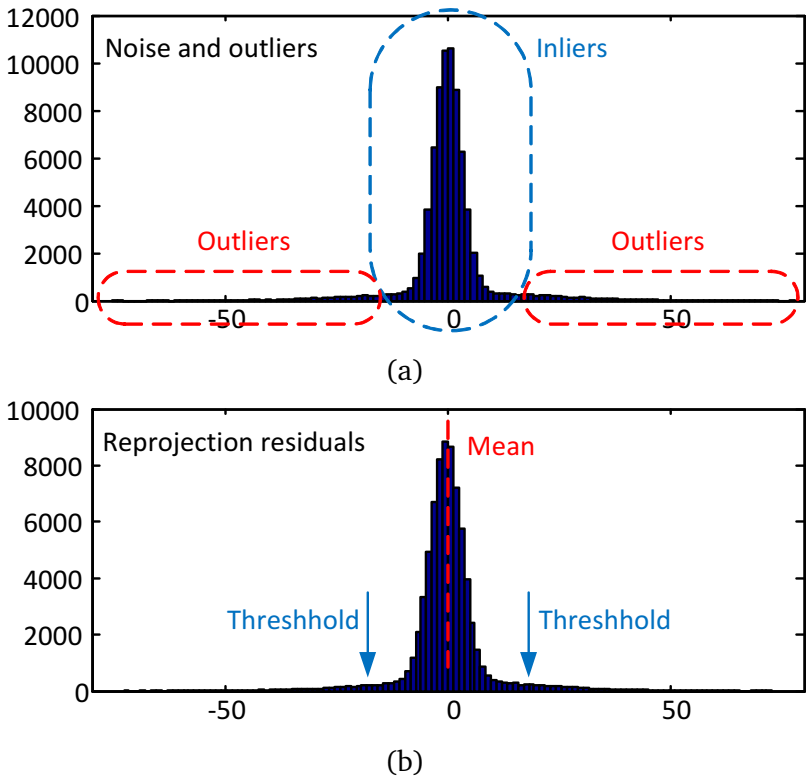



Figure 8.3: (a) Histogram distribution of the real added noise and outliers; (b) histogram distribution of the reprojection residuals.

Since the outliers have heavily influence to the estimation of the mean and standard deviation due to their large deviations, the mean is practically estimated from the data that are less than the median value of $|\mathcal{V}(\mathbf{E})|$.

$$\mu = \text{mean}\{\mathcal{V}'(\mathbf{E}) | \mathcal{V}'(\mathbf{E}) < \text{median}(|\mathcal{V}(\mathbf{E})|)\} \tag{8.20}$$

while the standard deviation is estimated from the median absolute deviation (MAD)

$$\sigma = 1.4826\ \text{median}(|\mathcal{V}(\mathbf{E}) - \text{median}(\mathcal{V}(\mathbf{E}))|). \tag{8.21}$$

The above computation is resistant to outliers and thus guarantees a robust estimation of the mean and standard deviation of the residuals.

The weight matrix is determined from the uncertainty of each feature based on the information such as sharpness and intensity contrast around its neighborhood [3][74]. The uncertainty is usually estimated during the process of feature detection and tracking or given as prior information. Nonetheless, this information is unavailable in many applications. In our early study [89], it was

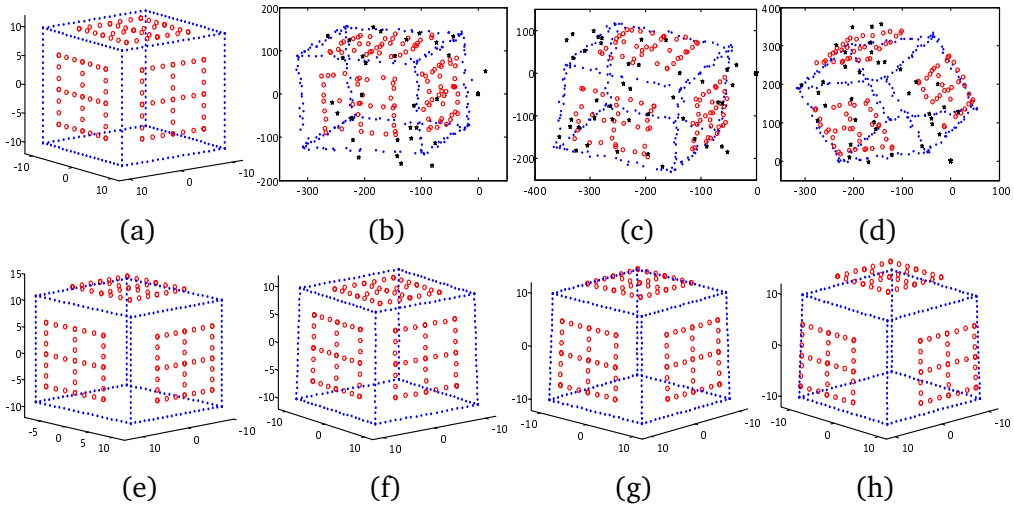


Figure 8.4: (a) (e) Two simulated space cubes with three sets of moving points; (b) (c) (d) three synthetic images with noise and outliers (shown as black stars); (f) (g) (h) the reconstructed 3D structures corresponding to the three images of (b), (c), and (d).

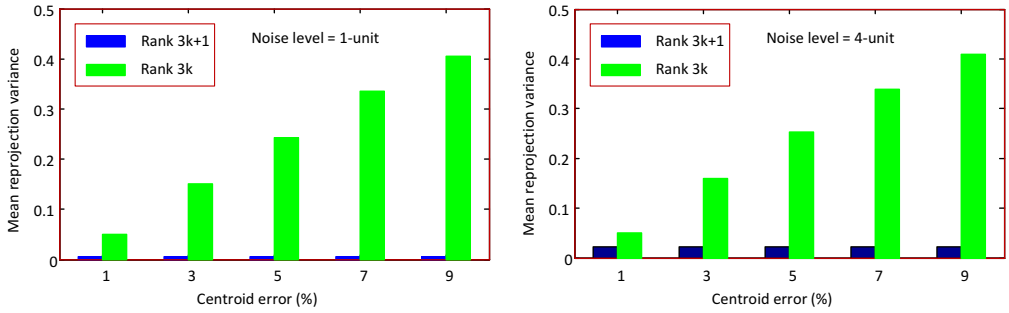
demonstrated that image uncertainty is generally proportional to the magnitude of reprojection residuals. The points with larger residuals have higher uncertainties, and vice versa. Based on this fact, the weight of each point is estimated directly from the residual matrix as:

$$\omega_{ij} = \frac{1}{\mathcal{N}} \exp\Big(-\frac{\mathbf{E}_{ij}^2}{2\sigma^2}\Big) \tag{8.22}$$

where the standard deviation $\sigma$ is estimated from the median absolute deviation (8.21) using the data after eliminating the outliers, $\mathbf{E}_{ij}$ is the $(i,j)$-th element of the residual matrix (8.16), and $\mathcal{N}$ is a normalization scale. Clearly, the weight of a point is directional, it may have different values at different coordinate directions based on its residual. The points with higher residuals get lower weights, and the weights of missing data and outliers are set at zeros.

## 8.5 Evaluations on Synthetic Data

The proposed technique was tested and evaluated extensively on synthetic data. During the simulation, we generated a deformable space cube, which was composed of 21 evenly distributed rigid points on each side and three sets of dynamic points ($33 \times 3$ points) on the adjacent surfaces of the cube that were moving




(a) (b)

Figure 8.5: The mean reprojection variance with respect to different centroid deviations at the noise level of (a) 1-unit and (b) 4-unit.

outward. There are 252 space points in total as shown in Figure 8.4. Using the synthetic cube, 100 images were generated by the affine projection with randomly selected camera parameter. Each image corresponds to a different 3D structure. The image resolution is $800 \times 800$ units and Gaussian white noise is added to the synthetic images.

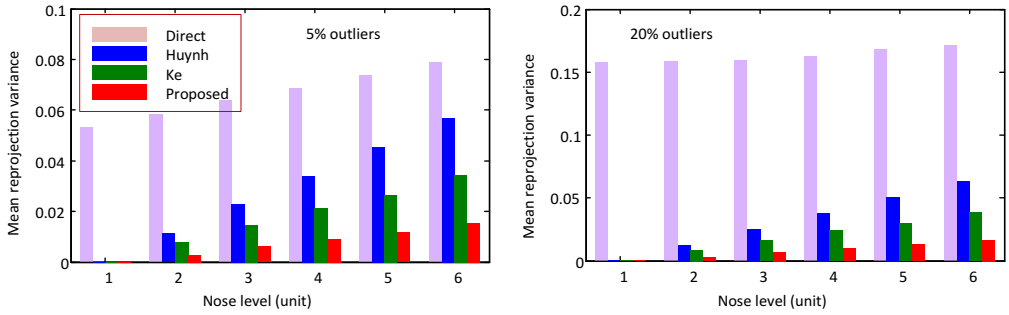
### 8.5.1 Influence of image centroid

The influence of the centroid was evaluated in this test. We deliberately deviated the centroid of the features in each image from 1% to 9% in steps of 2%, then registered all image points to the deviated centroid. This is a simulation of the situation that the centroid could not be reliably recovered due to missing and outlying data.

Using the misaligned data, the motion and shape matrices were recovered using the augmented factorization algorithm and its rank-3$k$ counterpart. The performance of different algorithms were evaluated and compared by means of the below defined mean reprojection variance.

$$E_{rv} = \frac{1}{mn}\|\mathbf{W}_0 - \hat{\mathbf{M}}\hat{\mathbf{S}}\|_F^2 \tag{8.23}$$

where $\mathbf{W}_0$ is the noise-free tracking matrix; $\hat{\mathbf{M}}$ and $\hat{\mathbf{S}}$ are the estimated motion and shape matrices, respectively. In order to obtain a statistically meaningful comparison, 100 independent tests were performed at each noise level. The mean reprojection variance with respect to different centroid deviations at different noise levels is shown in Fig. 8.5, where the noise level is defined as the standard deviation of the Gaussian noise.




(a) (b)

Figure 8.6: The mean reprojection variance with respect to different noise levels and outliers. (a) 5% outliers; (b) 20% outliers.

As shown in Fig. 8.5, the miscalculated centroid has no influence to the proposed augmented algorithm, but has an extremely large impact to the performance of rank-3$k$ based approach. The test indicates that the influence caused by the centroid error is far more significant than that caused by the image noise. Thus, it is better to choose the augmented affine factorization in practice, especially in cases that the centroid cannot be reliably recovered due to the presence of missing data and/or outliers.

### 8.5.2 Performance of the robust algorithm

For the above simulated image sequence, Gaussian noise was added to each image point (i.e. the points in the tracking matrix) and the noise level was varied from 1-unit to 5-unit in steps of 1. In the mean time, 10% outliers were added to the tracking matrix. Using the contaminated data, the foregoing proposed robust algorithm was employed to recover the motion and shape matrices.

Figure 8.4 shows three noise and outlier corrupted images and the corresponding 3D structures recovered by the proposed approach. It is evident that the deformable cube structures are correctly retrieved.

As a comparison, the direct nonrigid factorization algorithm without outlier rejection [75] and two successful robust algorithms in the literature were implemented as well, one is an outlier correction scheme proposed by Huynh *et al.* [39], the other one is proposed by Ke and Kanade [41] based on minimization of the $L1$ norm. The two robust algorithms were employed to recover the nonrigid structure using the same data in the test. The mean reprojection variance at different noise levels and outliers ratios is shown in Figure 8.6.

The results in Figure 8.6 were evaluated from 100 independent tests, and the reprojection variance was estimated only using the original inlying data so

Table 8.1: Real computation time of different algorithms (second)

| Frame no. | 50 | 100 | 150 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|
| Huynh | 3.62 | 12.25 | 23.18 | 41.93 | 70.54 | 99.27 |
| Ke | 5.48 | 32.61 | 90.17 | 176.42 | 303.26 | 527.98 |
| Proposed | 15.76 | 56.08 | 79.86 | 136.14 | 212.39 | 303.46 |

as to provide a fair comparison. Obviously, the proposed scheme outperforms other algorithms in terms of accuracy. The direct factorization algorithm yields significantly large errors due to the influence of outliers, and the error increases with the increase of the amount of outliers. The experiment also shows that all three robust algorithms are resilient to outliers, as can be seen in Fig. 8.6, the ratio of outliers has little influence to the reprojection variance of the three robust algorithms.

The complexity of different approaches was compared in the above test. We implemented all algorithms using Matlab on a Lenovo T500 laptop with 2,26GHz Intel Core Duo CPU. The frame number was varied from 50 to 300 in steps of 50 so as to generate different sizes of the tracking data, and 10% outliers were added to the data.

Table 8.1 shows the real computation time of different algorithms. Obviously, the complexity of the proposed scheme lies in between of 'Huynh' and 'Ke', but it yields the best accuracy. The minimization of the $L1$ norm in 'Ke' is computationally more intensive than the alternative factorization algorithm. Huynh's method does not include the step of weighted factorization, this is why it is relatively fast but it yields the lowest accuracy among the three algorithms.

## 8.6 Evaluations on Real Sequences

The method was tested on several real image sequences. The results on two data sets are reported here.

The first test was on a dinosaur sequence from paper [5]. The sequence consists of 231 images with various movement and deformation of a dinosaur model. The image resolution is $570 \times 338$ pixel and 49 features were tracked across the sequence. In order to test the robustness of the algorithm, an additional 8% outliers were added to the tracking data as shown in Figure 8.7.

Using the proposed approach, all outliers were successfully detected, however, a few tracked features were also eliminated due to large tracking errors. The proposed approach was employed to remove the outliers and recover the motion and structure matrices. Then, the solutions were upgraded to the Euclidean

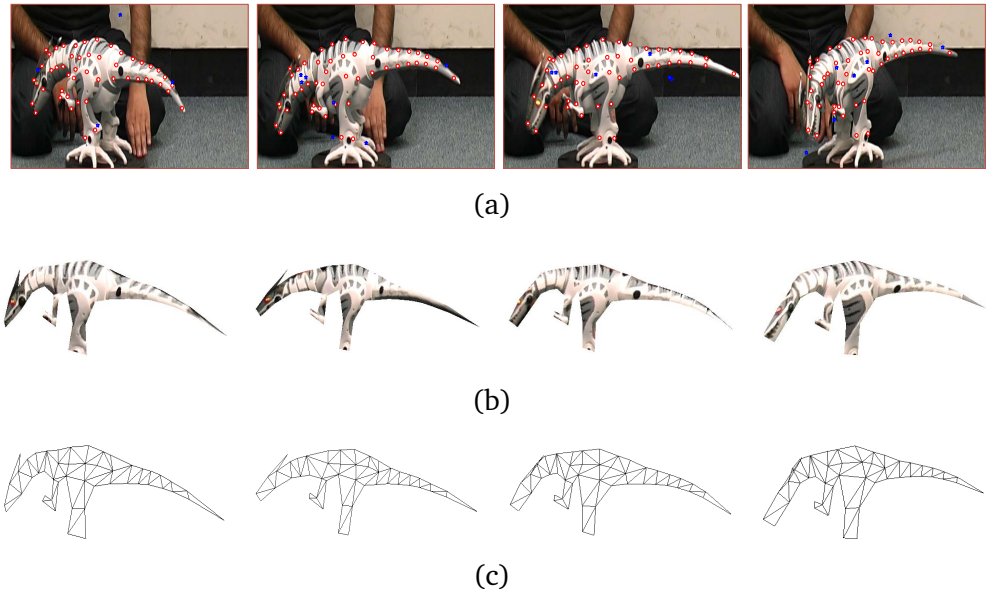
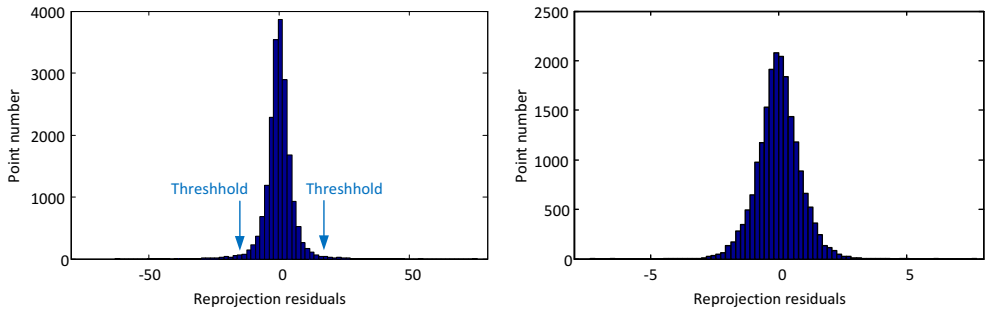


Figure 8.7: Test results of the dinosaur sequence. (a) Four frames from the sequence overlaid with the tracked features (red circles) and added outliers (blue stars); (b) the corresponding 3D VRML models from different viewpoints; and (c) the associated wireframes of the reconstructed models.

space. Figure 8.7 shows the reconstructed structure and wireframe at different viewpoints. It can be seen from the results that the deformed structure has been correctly recovered from the corrupted data, and the reconstructed VRML model is visually realistic.

The histogram distribution of the reprojection residual matrix (8.16) with outliers is shown in Figure 8.8. The residuals largely conform to the assumption of normal distribution. As can be seen from the histogram, the outliers are obviously distinguished from inliers, the computed threshold is shown in the figure. After rejecting outliers, the histogram distribution of the residuals produced by the final solutions is also shown in Figure 8.8. Obviously, the residual error is reduced significantly by the proposed approach. The final mean reprojection error given by the proposed approach is 0.597. In comparison, the reprojection errors by the algorithms of 'Huynh' and 'Ke' are 0.926 and 0.733, respectively. The proposed scheme outperforms other approaches.

The second test was on a face sequence with different facial expressions. The sequence was downloaded from FGnet, and 200 images from the sequence were used in the test. The image resolution is $720 \times 576$ with 68 automatically tracked feature points using the active appearance model (AAM). For test purpose, 8%




(a) (b)

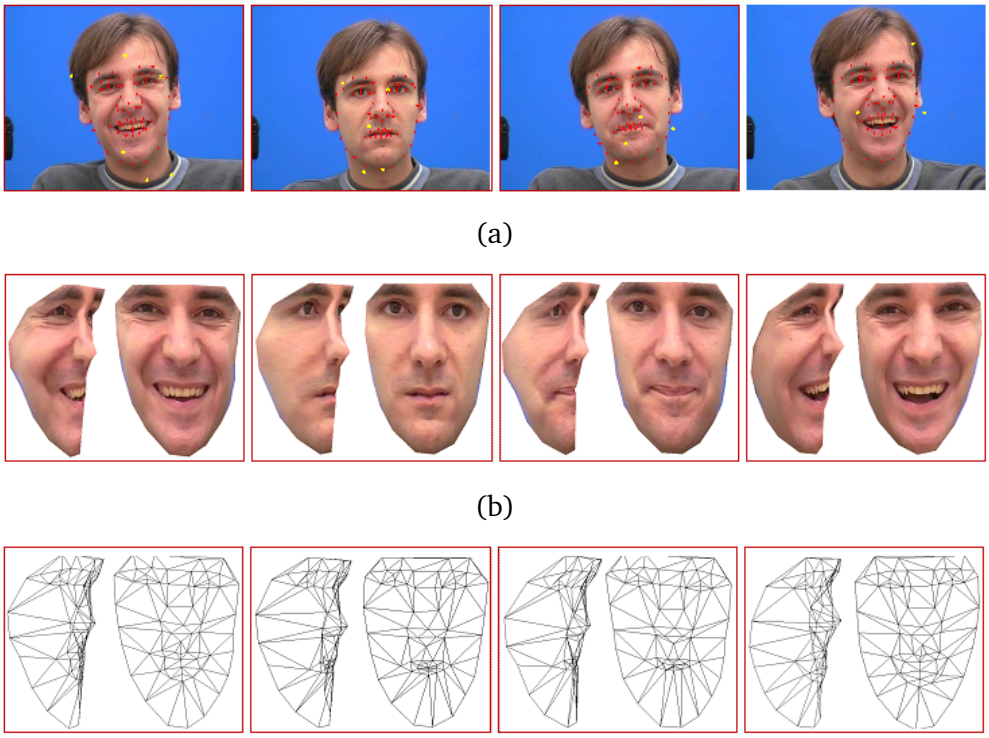
Figure 8.8: The histogram distribution of the residual matrix of the dinosaur sequence. (a) Before outlier rejection; (b) after outlier rejection.

outliers were added to the tracking data as shown in Figure 8.9.

The proposed robust algorithm was used to recover the Euclidean structure of the face. Figure 8.9 shows the reconstructed VRML models of four frames and the corresponding wireframes from different viewpoints. As demonstrated in the results, different facial expressions have been correctly recovered by the proposed approach. The reprojection errors obtained from 'Huynh', 'Ke', and the proposed algorithms are 0,697, 0.581, and 0.453, respectively. The proposed scheme yields the lowest reprojection error.

## 8.7 Conclusion

This chapter first proposed an augmented factorization algorithm which has been proved to be more accurate and more widely applicable than classical rank-3$k$ based nonrigid factorization, especially in the case that the image centroids could not be reliably recovered due to the presence of missing and outlying data. Then, an alternatively weighted factorization algorithm was presented to reduce the influence of large image noise. Finally, a robust factorization scheme was designed to deal with corrupted data containing outliers and missing points. The proposed technique requires no prior information of the error distribution in the tracking data. Extensive tests and evaluations demonstrated its advantages over the previous methods.



(a)

(b)

(c)

Figure 8.9: Test results of the face sequence. (a) Four frames from the sequence overlaid with the tracked features (red circles) and added outliers (yellow stars); (b) the corresponding 3D VRML models from different viewpoints; and (c) the associated wireframes of the reconstructed models.

# Chapter 9

# Conclusion and Future Work

This thesis started with an introduction to the state-of-the-art techniques of structure from motion and some challenges the field was facing. Then, we proposed several new algorithms to increase the accuracy, efficiency, and robustness of the previous approaches. This chapter revisits the contributions and conclusion of the thesis, followed by a discussion of potential research topics for future study.

## 9.1 Contributions and Conclusion

This thesis focused on the accuracy and robustness for the problem of structure from motion. The major contributions and conclusion are as follows.

The first part of the thesis discussed the accuracy and efficiency of structure from motion. First, a quasi-perspective projection model was proposed in Chapter 3 under the assumption that the camera was far away from the object with small lateral rotations. As a trade-off for the simplified affine assumption and the complicated full perspective projection model, the quasi-perspective model filled the gap between the two popular models. The proposed model was demonstrated theoretically and experimentally to be more accurate than the affine model.

More special geometric properties of the model were investigated in Chapter 4 in the context of one- and two-view geometry, including the quasi-fundamental matrix, plane induced homography, and quasi-perspective 3D reconstruction. Theoretical analysis and experimental study showed that the quasi-perspective model was a good trade-off between the simplicity and the accuracy of previous projection models.

Finally, the quasi-perspective model was applied to the problem of SfM, and a structure and motion factorization framework of both rigid and nonrigid objects was presented in Chapter 5. The new framework avoided the difficulty of com-

puting projective depths under perspective factorization. It was computationally efficient with better accuracy than affine approximation.

The proposed model is suitable for structure and motion factorization of a short sequence with small camera motions. It should be noted that the small rotation assumption of the proposed model is not a limiting factor and is usually satisfied in many real-world applications. For a long image sequence taken around an object, we can simply divide the sequence into several subsequences with small movements, then register and merge the results of each subsequence to reconstruct the structure of the entire scene.

The second part of the thesis focused on the robustness of SfM algorithms. First, a spatial-and-temporal-weighted factorization algorithm was proposed in Chapter 6 to handle significant noise contained in the tracking data, where the uncertainty of image measurement was estimated directly from the reprojection residuals. Unlike other existing weighted factorization algorithms, the proposed technique didn't require any prior information of the error distributions of the data, and the errors could be easily modeled both spatially and temporally to cope with different measurement uncertainties.

In Chapter 7, an augmented rank-4 affine factorization algorithm was proposed to handle outlying and missing data in rigid SfM. As a new addition to the previous affine factorization family, the new formulation is more accurate and more widely applicable than classical rank-3 affine factorization, since it avoids the difficulty of image alignment for erroneous data. Based on the distribution of reprojection residuals, a robust factorization scheme was designed to deal with corrupted data with outliers.

The robust algorithm of Chapter 7 was further extended to nonrigid scenarios in Chapter 8. A new augmented nonrigid factorization algorithm was first proposed to rout the difficulty in image alignment. Then, it was demonstrated that the outliers could be detected directly from the image reprojection residuals of nonrigid factorization. Thus, a similar robust scheme as rigid factorization was designed to handle outlying and missing data in nonrigid structure and motion recovery. The proposed technique was demonstrated empirically to be more efficient and more accurate than other robust algorithms.

The study in this thesis is not only academically significant, but also urgently needed by automotive and robotic industries for autonomous navigation and environment modeling. The study provides a new perspective for accurate and robust modeling of both rigid and nonrigid objects.

It should be noted that the proposed robust algorithms are based on the affine camera model. However, if the camera undergoes small lateral movements, we can adopt the quasi-perspective model in these robust algorithms. As studied in Chapter 5, the quasi-perspective model is more accurate than the affine model.

Thus, more accurate solutions are expected by assuming the quasi-perspective assumption.

## 9.2 Future Work

3D environment modeling and motion estimation using visual sensors are two main functions of autonomous robots. As a central theme in computer vision, the problem of 3D structure and motion recovery from image sequences has been widely studied during the past three decades and considerable progress has been made in theory and applications. Unfortunately, the problem remains far from being solved. One critical challenge is how to recover the 3D structure of dynamic natural scenes with rigid, nonrigid, articulated, and moving objects. The problem is very complex and has, thus, seldom been addressed in the literature. Some potential research topics in this field are as follows.

1. Structure and motion recovery of dynamic scenes

   Previous study on structure from motion dealt only with simple objects or scenes, however, a natural environment is more complex; it may be coupled with rigid, nonrigid, moving, and articulated objects. The 3D reconstruction of such a dynamic environment is still an open problem in computer vision. A possible approach is to solve this problem in trajectory space by introducing some constraint to the 3D trajectory of the scene.

2. Visual SLAM in dynamic environments

   The SfM of dynamic scenes has many potential applications. One direct application is the vision based simultaneous localization and mapping (SLAM) of mobile robots in dynamic environments. Another application is 3D reconstruction in video surveillance; the 3D information is very helpful for objects recognition and event detection from a dynamic surveillance video. If the problem of dynamic modeling is solved, we can directly use all information in the video, including moving and nonrigid objects. Thus, a faster and more reliable result is expected.

3. Robust algorithm for structure from motion

   Robustness is a well-known bottleneck of many computer vision applications, because outliers and missing data are inevitable during the process of feature tracking. This thesis provides a partial solution to the problem by virtue of the reprojection residuals, however, the problem has not been completely solved, especially for 3D modeling of dynamic environments. More efforts have to be addressed to this research.

# Bibliography

The numbers at the end of each entry list pages where the reference was cited. In the electronic version, they are clickable links to the pages.


[1] H. Aanæs, R. Fisker, K. Åström, and J. M. Carstensen. Robust factorization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(9):1215–1225, 2002. 12, 95

[2] A. K. Agrawal and R. Chellappa. Robust ego-motion estimation and 3-D model refinement using surface parallax. *IEEE Transactions on Image Processing*, 15(5):1215–1225, 2006. 108

[3] P. M. Q. Aguiar and J. M. F. Moura. Rank 1 weighted factorization for 3D structure recovery: Algorithms and performance analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1134–1049, 2003. 12, 83, 85, 86, 95, 108, 114, 119, 129, 138

[4] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, pages 41–48, 2008. 24

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(7):1442–1456, 2011. 24, 25, 142

[6] J. Y. Aloimonos. Perspective approximation. *Image and Vision Computing*, 8(3):177–192, 1990. 27

[7] P. Anandan and M. Irani. Factorization with uncertainty. *International Journal of Computer Vision*, 49(2-3):101–116, 2002. 12, 83, 85, 86, 114, 119

[8] B. Bascle and A. Blake. Separability of pose and expression in facial tracing and animation. In *Proc. of the International Conference on Computer Vision*, pages 323–328, 1998. 10

[9] M. Brand. Morphable 3D models from video. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 456–463, 2001. 11, 19, 20, 21, 129

[10] M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2)*, volume 2, pages 122–128, 2005. 11, 22, 62

[11] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 690–696, 2000. 10, 11, 17, 19, 24, 62, 63, 129, 130

[12] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2)*, pages 316–322, 2005. 12, 84

[13] C.-M. Cheng and S.-H. Lai. A consensus sampling technique for fast and robust model fitting. *Pattern Recognition*, 42(7):1318–1329, 2009. 42

[14] S. Choi, T. Kim, and W. Yu. Performance evaluation of ransac family. In *BMVC*, 2009. 11, 115

[15] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(11):1098–1104, 1996. 9, 15, 27

[16] J. Costeira and T. Kanade. A multibody factorization method for independent moving objects. 29(3):159–179, September 1998. 10, 62

[17] D. J. Crandall, A. Owens, N. Snavely, and D. P. Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2841–2853, 2013. 1

[18] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using nonparametric tracking and non-linear optimization. In *IEEE Workshop in Articulated and Nonrigid Motion ANM04, held in conjunction with CVPR2004*, pages 8–15, June 2004. 11

[19] A. Del Bue. Deformable 3-D modelling from uncalibrated video sequences. *PhD Thesis, Queen Mary, University of London*, 2007. 20, 21

[20] A. Del Bue and L. Agapito. Non-rigid stereo factorization. *International Journal on Computer Vision*, 66(2):193–207, 2006. 10

[21] A. Del Bue, X. Lladó, and L. de Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (1)*, volume 1, pages 1191–1198, 2006. 11, 62

[22] F. Dellaert, S. M. Seitz, C. E. Thorpe, and S. Thrun. Structure from motion without correspondence. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2557–2564, 2000. 42

[23] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In *CVPR*, pages 771–778, 2010. 12, 108, 130

[24] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993. 27

[25] O. Faugeras. Stratification of 3-D vision: projective, affine, and metric representations. *Journal of the Optical Society of America A*, 12:46548–4, 1995. 52

[26] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 11, 41, 50, 115

[27] P. F. U. Gotardo and A. M. Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):2051–2065, 2011. 25

[28] P. F. U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *ICCV*, pages 802–809, 2011. 25

[29] A. Gruber and Y. Weiss. Multibody factorization with uncertainty and missing data using the EM algorithm. In *CVPR (1)*, pages 707–714, 2004. 11, 12, 84, 107, 108, 129

[30] N. Guilbert, A. Bartoli, and A. Heyden. Affine approximation for direct batch recovery of Euclidian structure and motion from sparse data. *International Journal of Computer Vision*, 69(3):317–333, 2006. 42

[31] M. Han and T. Kanade. Creating 3D models with uncalibrated cameras. In *proc. of IEEE Computer Society Workshop on the Application of Computer Vision (WACV2000)*, December 2000. 23, 65, 76, 77

[32] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 11, 15, 27, 29, 48, 52, 53, 71, 107

[33] R. Hartley and F. Schaffalizky. Powerfactorization: 3D reconstruction with missing or uncertain data. In *proc. of Australia-Japan Advanced Workshop on Computer Vision*, 2003. 11, 72, 107, 112, 113

[34] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *Proc. of European Conference on Computer Vision (ECCV) (1)*, volume 5302 of *Lecture Notes in Computer Science*, pages 276–289. Springer, 2008. 9, 11, 62

[35] R. Hartley. In defense of the eight-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):580–593, 1997. 41, 48

[36] M. He, H. Li, and Y. Dai. A simple prior-free method for non-rigid structure-from-motion factorization. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 0:2018–2025, 2012. 11

[37] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image Vision Comput.*, 17(13):981–991, 1999. 10, 16

[38] M. Hu, K. McMenemy, S. Ferguson, G. Dodds, and B. Yuan. Epipolar geometry estimation based on evolutionary agents. *Pattern Recognition*, 41(2):575–591, 2008. 41

[39] D. Q. Huynh, R. Hartley, and A. Heyden. Outlier correction in image sequences for the affine camera. In *ICCV*, pages 585–590, 2003. 108, 113, 122, 123, 130, 141

[40] Y. Sugaya K. Kanatani and H. Ackermann. Uncalibrated factorization using a variable symmetric affine camera. *IEICE Transactions on Information and Systems*, E90-D(5):851–858, 2007. 28

[41] Q. Ke and T. Kanade. Robust $L1$ norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR (1)*, pages 739–746, 2005. 12, 107, 108, 113, 115, 122, 123, 130, 141

[42] B. Klingner, D. Martin, and J. Roseborough. Street view motion-from-structure-from-motion. In *Proc. of IEEE International Conference on Computer Vision*, 2013. 1

[43] S. Lehmann, A. P. Bradley, I. Vaughan L. Clarkson, J. Williams, and P. J. Kootsookos. Correspondence-free determination of the affine fundamental matrix. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):82–97, 2007. 42

[44] T. Li, V. Kallem, D. Singaraju, and R. Vidal. Projective factorization of multiple rigid-body motions. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 10, 62

[45] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157, 1999. 3

[46] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision (ijcai). In *Proc. of International Joint Conference on Artificial Intelligence*, pages 674–679. 3

[47] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 430–437, 2000. 10, 16, 23, 62

[48] P. R. S. Mendonça and R. Cipolla. Analysis and computation of an affine trifocal tensor. In *Proc. British Machine Vision Conference*, pages 125–133, Southampton, UK, 1998. 42

[49] D. D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proc. of IEEE International Conference on Computer Vision*, pages 696–702, 1998. 11, 86, 119

[50] P. Moulon, P. Monasse, and R. Marlet. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *Proc. of IEEE International Conference on Computer Vision*, pages 3248–3255, 2013. 129

[51] J. L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, 1992. 27, 42

[52] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Mach. Vis. Appl.*, 16(5):321–329, 2005. 11

[53] Y.I. Ohta, K. Maenobu, and T. Sakai. Obtaining surface orientation from texels under perspective projection. In *In Proc. International Joint Conferences on Artificial Intelligence*, pages 746–751, 1981. 31

[54] T. Okatani and K. Deguchi. On the Wiberg algorithm for matrix factorization in the presence of missing components. *International Journal of Computer Vision*, 72(3):329–337, 2007. 12

[55] T. Okatani, T. Yoshida, and K. Deguchi. Efficient algorithm for low-rank matrix factorization with missing components and performance comparison of latest algorithms. In *ICCV*, pages 842–849, 2011. 12, 108, 130

[56] J. Oliensis and R. Hartley. Iterative extensions of the Sturm/Triggs algorithm: Convergence and nonconvergence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2217–2233, 2007. 10, 27, 62

[57] K. E. Ozden, K. Schindler, and L. Van Gool. Multibody structure-from-motion in practice. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(6):1134–1141, 2010. 1

[58] D. A. Ross, D. Tarlow, and R. S. Zemel. Learning Articulated Structure and Motion. *International Journal of Computer Vision*, 88(2):214–237, 2010. 10

[59] A. Zisserman P. H. S. Torr and S. J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding: CVIU*, 71(3):312–333, 1998. 42, 79

[60] M. Paladini, A. Del Bue, J. M. F. Xavier, L. de Agapito, M. Stosic, and M. Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *International Journal of Computer Vision*, 96(2):252–276, 2012. 12, 108, 130

[61] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *ECCV (3)*, pages 158–171, 2010. 25

[62] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern and Analysis and Machine Intelligence*, 19(3):206 – 218, March 1997. 9, 31, 61, 76

[63] G. Qian, R. Chellappa, and Q. Zheng. Bayesian algorithms for simultaneous structure from motion estimation of multiple independently moving objects. *IEEE Transactions on Image Processing*, 14(1):94–109, 2005. 108

[64] L. Quan and T. Kanade. A factorization method for affine structure from line correspondences. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 803–808, June 1996. 10

[65] L. Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–105, 1996. 1, 9, 61, 65

[66] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 11, 62, 129

[67] D. Scaramuzza. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. 95(1):74–85, October 2011. 11, 107

[68] L. S. Shapiro, A. Zisserman, and M. Brady. 3D motion recovery via affine epipolar geometry. *International Journal of Computer Vision*, 16(2):147–182, 1995. 27, 42

[69] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, 1994. 3

[70] I. Shimshoni, R. Basri, and E. Rivlin. A geometric interpretation of weak-perspective motion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(3):252–257, 1999. 42

[71] P. F. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proc. of European Conference on Computer Vision (ECCV) (2)*, pages 709–720, 1996. 9, 15, 61, 71, 118

[72] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, ISBN-10: 1848829345, 2011. 1

[73] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, pages 2761–2768, 2010. 10

[74] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992. 1, 9, 19, 20, 61, 86, 119, 138

[75] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):878–892, 2008. 11, 62, 129, 141

[76] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proc. of IEEE Conference*

*on Computer Vision and Pattern Recognition*, volume 1, pages 493–500, 2001. 11, 19, 20

[77] B. Triggs. Factorization methods for projective structure and motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 845–851, 1996. 1, 9, 15, 61

[78] R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *Proc. of European Conference on Computer Vision (ECCV) (2)*, volume 3952 of *Lecture Notes in Computer Science*, pages 205–218. Springer, 2006. 11

[79] R. Vidal and R. Hartley. Motion segmentation with missing data using powerfactorization and GPCA. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (2)*, volume 2, pages 310–316, 2004. 11

[80] G. Wang. A hybrid system for feature matching based on sift and epipolar constraints. *Tech. Rep. Department of ECE, University of Windsor*, 2006. 52, 57, 79, 101, 125

[81] G. Wang, H.-T. Tsui, and Z. Hu. Structure and motion of nonrigid object under perspective projection. *Pattern Recognition Letters*, 28(4):507–515, 2007. 62

[82] G. Wang, H.-T. Tsui, and J. Wu. Rotation constrained power factorization for structure from motion of nonrigid objects. *Pattern Recogn. Lett.*, 29(1):72–80, 2008. 11, 21, 62

[83] G. Wang, and J. Wu. Quasi-perspective projection with applications to 3D factorization from uncalibrated image sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 28, 59, 62

[84] G. Wang, and J. Wu. Stratification approach for 3-D euclidean reconstruction of nonrigid objects from uncalibrated image sequences. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(1):90–101, 2008. 72

[85] G. Wang, and J. Wu. The quasi-perspective model: Geometric properties and 3D reconstruction. *Pattern Recognition*, 43(5):1932–1942, 2010. 28, 40, 42

[86] G. Wang, and J. Wu. Quasi-perspective projection model: Theory and application to structure and motion factorization from uncalibrated image sequences. *International Journal of Computer Vision*, 87(3):213–234, 2010. 28, 40, 42, 62, 111, 133

[87] G. Wang, and J. Wu. *Guide to Three Dimensional Structure and Motion Factorization*. Springer-Verlag, ISBN: 978-0-85729-045-8, 2011. 95, 102

[88] G. Wang, and J. Wu. Perspective 3D euclidean reconstruction with varying camera parameters. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(12):1793–1803, 2009. 58, 62, 72

[89] G. Wang, J. Zelek, and J. Wu. Structure and motion recovery based on spatial-and-temporal-weighted factorization. *IEEE Trans. Circuits Syst. Video Techn.*, 22(11):1590–1603, 2012. 84, 108, 119, 130, 138

[90] G. Wang, J. Zelek, and J. Wu. Robust structure from motion of nonrigid objects in the presence of outlying and missing data. In *CRV*, pages 159–166, 2013. 130

[91] G. Wang, J. Zelek, J. Wu, and R. Bajcsy. Robust rank-4 affine factorization for structure from motion. In *WACV*, pages 180–185, 2013. 108

[92] H. Wang, T.-J. Chin, and D. Suter. Simultaneously fitting and segmenting multiple-structure data with outliers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(6):1177–1192, 2012. 12, 108, 130

[93] J. Y. Weng, T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(5):451–476, 1997. 48

[94] L. Wolf and A. Shashua. Affine 3-D reconstruction from two projective images of independently translating planes. In *Proc. of the International Conference on Computer Vision*, pages 238–244, 2001. 42

[95] J. Xiao, J. X. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, April 2006. 11, 19, 62, 78

[96] J. Xiao, and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Proc. of the International Conference on Computer Vision*, volume 2, pages 1075–1082, 2005. 11, 21, 22, 23, 62, 64, 74, 78

[97] J. Yan and M. Pollefeys. A factorization-based approach to articulated motion recovery. In *IEEE Conference on Computer Vision and Pattern Recognition (2)*, pages 815–821, 2005. 10, 62

[98] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(5):865–877, 2008. 10, 62

[99] J. Yu, T.-J. Chin, and D. Suter. A global optimization approach to robust multi-model fitting. In *CVPR*, pages 2041–2048, 2011. 12, 108, 130

[100] A. Zaharescu and R. Horaud. Robust factorization methods using a Gaussian/uniform mixture model. *International Journal of Computer Vision*, 81(3):240–258, 2009. 12, 83, 84, 108, 114, 129

[101] L. Zelnik-Manor and M. Irani. Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2:287, 2003. 10

[102] L. Zelnik-Manor and M. Irani. On single-sequence and multi-sequence factorizations. *International Journal of Computer Vision*, 67(3):313–326, 2006. 10

[103] L. Zelnik-Manor, M. Machline, and M. Irani. Multi-body factorization with uncertainty: Revisiting motion consistency. *International Journal of Computer Vision*, 68(1):27–41, 2006. 12, 84, 108, 129

[104] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. 57

[105] Z. Zhang, P. Anandan, and H.-Y. Shum. What can be determined from a full and a weak perspective image? In *Proc. of International Conference on Computer Vision*, pages 680–687, 1999. 42

[106] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78(1-2):87–119, 1995. 3

[107] Z. Zhang and T. Kanade. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998. 41, 47, 48

[108] Z. Y. Zhang and G. Xu. A general expression of the fundamental matrix for both projective and affine cameras. In *Proc. of International Joint Conference on Artificial Intelligence*, pages 1502–1507, 1997. 42