3D Non-Rigid Reconstruction with Prior Shape Constraints

by

Lili Tao
陶立力

A thesis submitted in partial fulfilment for the requirements for the degree of

Doctor of Philosophy

at

the University of Central Lancashire

May 2014

Student Declaration

Concurrent registration for two or more academic awards

Either *I declare that while registered as a candidate for the research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution

or ~~*I declare that while registered for the research degree, I was with the University's specific permission, a *registered candidate/*enrolled student for the following award:~~

_____

Material submitted for another award

Either *****I declare that no material contained in the thesis has been used in any other submission for an academic award and is solely my own work

or ~~*I declare that the following material contained in the thesis formed part of a submission for the award of~~

_____

(state award and awarding body and list he material below):

* *delete as appropriate*

Collaboration

Where a candidate's research programme is part of a collaborative project, the thesis must indicate in addition clearly the candidate's individual contribution and the extent of the collaboration. Please state below:

Signature of Candidate _____**Lili Tao**_____

Type of Award _____**Doctor of Philosophy**_____

School ___**School of Computing, Engineering and Physical Sciences**__

Abstract

3D non-rigid shape recovery from a single uncalibrated camera is a challenging, under-constrained problem in computer vision. Although tremendous progress has been achieved towards solving the problem, two main limitations still exist in most previous solutions. First, current methods focus on non-incremental solutions, that is, the algorithms require collection of all the measurement data before the reconstruction takes place. This methodology is inherently unsuitable for applications requiring real-time solutions. At the same time, most of the existing approaches assume that 3D shapes can be accurately modelled in a linear subspace. These methods are simple and have been proven effective for reconstructions of objects with relatively small deformations, but have considerable limitations when the deformations are large or complex. The non-linear deformations are often observed in highly flexible objects for which the use of the linear model is impractical.

Note that specific types of shape variation might be governed by only a small number of parameters and therefore can be well-represented in a low dimensional manifold. The methods proposed in this thesis aim to estimate the non-rigid shapes and the corresponding camera trajectories, based on both the observations and the prior learned manifold.

Firstly, an incremental approach is proposed for estimating the deformable objects. An important advantage of this method is the ability to reconstruct the 3D shape from a newly observed image and update the parameters in 3D shape space. However, this recursive method assumes the deformable shapes only have small variations from a mean shape, thus is still not feasible for objects subject to large scale deformations. To address this problem, a series of approaches are proposed, all based on non-linear manifold learning techniques. Such manifold is used as a shape prior, with the reconstructed shapes constrained to lie within the manifold. Those non-linear manifold based approaches significantly improve the quality of reconstructed results and are well-adapted to different types of shapes undergoing significant and complex deformations.

Throughout the thesis, methods are validated quantitatively on 2D points sequences projected from the 3D motion capture data for a ground truth comparison, and are qualitatively demonstrated on real example of 2D video sequences. Comparisons are made for the proposed methods against several state-of-the-art techniques, with results shown for a variety of challenging deformable objects. Extensive experiments also demonstrate the robustness of the proposed algorithms with respect to measurement noise and missing data.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deeply-felt thanks to my director of studies, Dr. Bogdan Matuszewski, for providing me the constant support and every bit of guidance, assistance, and expertise over the last three years. He not only guided me in the area of computer vision, but also inspired me with his insights, enthusiasm and hard working attitude to research. He gave me a lot of encouragement during the good times and also endless patience during the hard times. I always feel very lucky that I have a supervisor like him. I wish the achievement I have got and I will obtain can make him proud.

I would like to express my gratitude to my second supervisor Stephen Mein for comments and helpful feedback on papers and the thesis. Thanks must also go to other staff in the school of CEPS and all the members of the ADSIP research centre at UCLan for all the great times that we have shared. I am particularly thankful to my colleagues Bartek, Wei and Pedro, for helpful advices whenever I asked.

I would like to thank my friends from inside/outside of the university, even outside of the country for sharing my happiness and patiently listening to my complaint about the research. Especially thanks my good friend, Wen, for giving help on presentation design. The slides always look much better after took her suggestions. Her helps made it possible for me to show my work in a fun and professional way.

Most of all, I will forever be thankful to my family, especially my mum who has scarified a lot only for me. I owe her everything and wish I could show how much I appreciate her. Her love and immense support provided me inspiration and motivation. This thesis would not have been written without her.

Last three years was the most memorable and pleasant time I have ever had. It was the time that I grew to be a researcher and felt the beauty of science. I also would like to thank myself (sounds a bit strange) for focusing on research and not giving up at all times. I believe all those special memories will give me the confidence to face anything the future may holds.

Acronyms

2D	Two-dimensional
3D	Three-dimensional
DCT	Discrete Cosine Transform
dof	degree of freedom
GSVD	Generalised Singular Value Decomposition
IBR	Image Based Rendering
IPCA	Incremental Principal Component Analysis
KLT	Kanade-Lucas-Tomasi feature tracker
LLE	Locally Linear Embedding
MDS	Multidimensional Scaling
MoCap	Motion Capture
NRSfM	Non-Rigid Structure from Motion
PCA	Principal Component Analysis
PPCA	Probabilistic Principal Components Analysis
RIK	Rotation Invariant Kernels
RMS	Root Mean Square
SfM	Structure from Motion
SLAM	Simultaneous Localisation and Mapping
SSD	Sum of Squared Differences
SVD	Singular Value Decomposition

Related publications

- Lili Tao, Stephen J. Mein, Wei Quan and Bogdan J. Matuszewski, **Recursive non-rigid structure from motion with online learned shape prior**, *Computer Vision and Image Understanding 117*, 2013.

- Lili Tao and Bogdan J. Matuszewski , **3D deformable shape reconstruction with diffusion maps**, *24th British Machine Vision Conference (BMVC 2013)*, 2013.

- Lili Tao and Bogdan J. Matuszewski, **Deformable shape reconstruction from monocular video with manifold forests**, *15th International Conference on Computer Analysis of Images and Patterns (CAIP 2013)*, 2013.

- Lili Tao and Bogdan J. Matuszewski, **Non-rigid structure from motion with diffusion maps prior**, *26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, 2013.

- Lili Tao, Bogdan J. Matuszewski and Stephen J. Mein, **Non-rigid structure from motion with incremental shape prior**, *19th IEEE International Conference on Image Processing (ICIP 2012)*, 2012.

- Lili Tao, Bogdan J. Matuszewski and Stephen J. Mein, **Model constraints for non-rigid structure from motion**, *British Machine Vision Conference (BMVC 2011) PhD Workshop*, 2011.

Chapter 1

Introduction

This chapter provides a brief review of the 3D shape and motion recovery problem. Several successful applications based on solutions of this problem are detailed and the limitations of these existing approaches discussed. Finally, a summary is presented of the original contributions made in this thesis.

1.1 Background

The main task of computer vision is to analyse, process and understand the world from images captured by visual sensors. Shape and motion estimation is one of the most fundamental problems in computer vision, which has made remarkable progress over the last two decades. Solutions for this problem have a wide range of applications in many different areas. They have been successfully used in object recognition, robot navigation, augmented reality, biomedical engineering, human-computer interaction and entertainment. Among these, 3D reconstruction employs a variety of techniques, but the complexity of the task differs widely under different conditions. The first section provides an overview of existing techniques of recovery using a single camera, in the wider context of reconstruction problems.

The problem can be seen as simultaneous recovery of object's 3D structure and its relative camera motion. Obtaining information about the geometry of 3D shapes and the corresponding camera information from a set of images is a challenging problem, also known as Structure from Motion (SfM). The task is difficult because the image formation is not invertible [113]. Given only 2D projected position of scene points in a camera plane, it is impossible to recover their distance from the camera. To address this, additional information is needed to solve the reconstruction problem. The idea was first introduced by Ullman in [137] who provides the original proof of existence of the solution under orthographic camera model. A solution to the perspective model was formulated in [112]. After these seminal works, numerous methods have been presented in this field under the assumption of scene rigidity, which means the geometry of object is fixed, the only motion included in the model is camera motion.

One of the most influential solutions of rigid SfM was proposed by Tomasi and Kanade [129]. They demonstrated a factorisation algorithm based on the singular value decomposition (SVD) for reconstruction of rigid objects by making an assumption that the camera operates under an orthographic projection model. The factorisation algorithm can be extended to deal with weak perspective projection [106, 104, 71] and perspective camera models [58, 68, 134, 105, 69]. Since then, techniques for rigid shape recovery via point-based SfM have achieved maturity [125, 47]. With more and more challenging SfM applications, the rigid model became insufficient to represent a scene. The algorithms have been developed to deal with multiple rigid objects [31, 59, 8] and articulated rigid objects [132].

By the 2000s, the reconstruction of rigid objects became a well-established process. However, in real environments many objects of interest are non-rigid as they deformation over time, e.g. human body due to movement [3, 53], face due to articulation [19, 150] and other objects of interest [46]. Therefore the research has expanded

into deformable object reconstruction. In contrast to rigid object reconstruction, deformable shape reconstruction is still challenging, mainly because it is a severely under-constrained problem. This is particularly true for the articulated deformable objects or the object which contains large and complex deformations. Such time varying shape recovery problem is known as Non-Rigid Structure from Motion (NRSfM).

Bregler et al. [19] were the first to adopt the factorisation algorithm to deformable 3D structures by introducing a low rank shape model to represent deformable shapes. As a time-varying object usually cannot arbitrarily deform, the idea of this model is to describe a deformable shape as a linear combination of a small number of basis shapes. Due to its simplicity, this shape model has been widely used to tackle the NRSfM [9, 151, 5]. Departing from the low rank shape model, a model based on point trajectory information was proposed in [5] by Akhter et al. who described a duality theorem in 3D structure representation which models independent 3D point trajectories. The main advantage of this representation is that the basis trajectories can be predefined, thus removing a large number of unknowns from the estimation [53, 52].

Considering that the inherently high number of degrees of freedom and motion degeneracy, depend only on the 2D measurements, such methods may fail to provide meaningful reconstruction. To counter this effect, it is common to introduce prior information to define additional constraints into minimisation of the 2D re-projection error. **Statistic priors** and **physical priors** are the most commonly used constraints in the NRSfM problem and both approaches have been proven to produce high quality reconstructions.

Both low rank shape model and trajectory space bases model are regarded as statistic prior. Other statistical priors include Probabilistic Principal Components Analysis (PPCA), which was firstly applied to NRSfM problem in [130] as a hierarchical Bayesian

prior [49]. Bartoli et al. [9] introduced another type of statistic prior based on coarse-to-fine model. In that method, the basis shapes are ordered starting from a mean shape and deformation modes are iteratively added. Recently Zhou et al. [156] proposed a method operating in the presence of nonlinear motion and non-Gaussian distribution using the Markov chain Monte Carlo technique which is applied to minimise the residuals of the estimated shapes. An alternative approach to SfM is bundle adjustment demonstrated by Del Bue, in which rigid shape prior was introduced [37].

Since shapes do not deform in an arbitrary way, physical prior can help to force the object moving in a specific way. The methods using physical prior include inextensible surface [141, 103], smooth constraint on deformation [23], piecewise planar [139, 126, 45] or partially rigid/non-rigid model [38, 79]..

Linear techniques perform well only if the deformations are relatively small or simple, but fail to deal with more complex deformable shape. To move away from the linear representation of deformable shapes, Rabaud and Belongie [109] integrated the Locally Smooth Manifold Learning algorithm to regularise the NRSfM problem. However, there is no guarantee that the manifold is planar or isometric to a plane. Despite the manifold learning techniques becoming increasingly popular and having been successfully used in different applications including medical image analysis [148], object classification [86] and segmentation [42], these techniques have not been widely applied in the NRSfM problem.

1.2 Applications

3D reconstruction technology has been successfully used in many different areas, ranging from medical imaging and biometrics to computer gaming, animation and film production. The third dimension (depth information) plays a significant role in under-

standing and analysing static or dynamic objects and environments.

Many new applications require reliable depth data in order to improve performance. It is particularly important in many medical areas. A typical example is the minimally invasive surgery. Although the image guided surgery might be able to meet the requirement, 3D shapes would bring more information than only using 2D images. The study in [142] evaluated the effect of 2D and 3D visualisation on robotic surgery and proved that loss of 3D vision significantly increased perceived difficulty and slowed down the progress of the task. NRSfM can be used to help assess the size and shape of organs. This is a rapidly growing application area and it is anticipated that, within the next few years, the medical industry will launch affordable 3D vision systems.

Real time rigid structure from motion techniques have often been applied to Augmented Reality (AR) systems. AR technology can be seen as inserting artificial objects in a video. In advanced AR tasks, interaction with real world needs to be considered as well. It is obviously impossible to get realistic insertion which appears consistent with the background video if the scene in the video and camera motion is unknown. For more challenging cases, the augmented objects may be inserted in a dynamic scene, which makes it even more difficult to build a comprehensive map of the scene in real time since the shape of the object in the environment changes over time.

This reconstruction technique has recently become very popular in the entertainment industry. For example, in the movie "Avatar", a multi-camera system was used to track and reproduce an actor's skeletal motion from a 3D point cloud. The point cloud was created using 2D data collected from the cameras during a performance. It was then re-targeted on the 3D animated characters in post-processing, which greatly reduced the animators' workload. Technology used in movie industry usually relies on the Motion Capture (MoCap) system. The system is composed of 6-12 synchronised infrared cameras. To capture something, reflected markers are placed on the surface,

and each marker has to be captured by at least three cameras in order to get precise 3D data. Basically, the MoCap system uses 3D optical marker-based technology and is able to track and analyse movement. The system can handle many complex motion capture problems and has been used for engineering, entertainment and life sciences.

Although the use of infrared markers together with a multi-camera system to track and reconstruct the body or objects has been employed successfully, to use them in some cases is still unrealistic; for example, in the previous mentioned medical imaging applications for robotic surgery, human computer interaction and surveillance applications. Markerless reconstruction seems especially useful for these situations. Visual surveillance systems are employed for observation and protection of public and private areas. Since the subjects are observed unknowingly by the camera(s), the marker-based systems are not applicable for such surveillance applications. Most existing systems are primarily based on 2D information; a comprehensive review of current 2D surveillance system is provided in [61]. But when using 2D techniques it is very hard to handle the occlusion problem and track multiple people, whereas 3D data can resolve those problems.

Microsoft Kinect is an example of successful application of the 3D sensor in the gaming industry for real-time human pose detection and recognition [121]. With the availability of using RGB-D sensor, where "D" is the depth map produced by a sensor, it has become very popular recently, especially for tasks which have traditionally been difficult to solve. Unlike the MoCap system, the RGB-D cameras are not expensive and do not require a complicated calibration process. These types of sensors opened up a new area in 3D computer vision.

1.3 Motivation and Aims

The aim of structure from motion research is to jointly reconstruct 3D deformable shapes and estimate the corresponding camera motion trajectory based on observations from a set of images. The original formulation of the problem uses a moving monocular camera as the only sensor. But with advances in technology, alternative sensors and multiple camera system have been used to achieve the goal. One example mentioned previously is the infrared cameras in the Motion Capture system. In comparison with a single camera, the system setup and synchronisation of multiple cameras is rather complex, despite the fact that different visual sensors would bring more accurate reconstructed results. The main difficulties with the MoCap system are the need for markers and the requirement that the cameras need to be kept in a fixed position in relation to each other. Since handheld cameras are more portable, do not relay on reflected markers and are not restricted to specific types of objects, the input data used in our reconstruction techniques are only considered to be obtained from monocular video sequences.

The problem of reconstruction on rigid objects or static scene is well-understood. Current implementation of rigid SfM is able to handle the case of missing data in the measurements, large-scaled scene and has the ability to process data on real time, while most non-rigid shape reconstruction systems are still extremely restricted. Most extant works in structure from motion for deformable objects focus on non-incremental solutions. These batch type algorithms require collecting all the measurement data before the reconstruction takes place. This methodology is inherently unsuitable for applications that require real-time. An ideal online system should be capable of incrementally learning the model, and updating the model by using current measurements. Estimation of 3D structure and camera information needs to be done when the corresponding

frame arrives. On the other hand, most of these batch approaches only perform well when the deformations are relatively small or simple, but fail when more complex deformations need to be recovered. The main limitation of the current linear representations of shapes is that they overlook the problem that non-linear deformations are often observed during the reconstruction process.

Generally speaking, the work reported in this thesis focuses on the recovery of non-rigid 3D shapes from 2D observations acquired with a single camera. More specifically, it explores the recovery of highly deformable shapes through integration of the learned shape prior manifold into the NRSfM solver. The purpose of this work is motivated by the current general progress in the NRSfM area, but concentrates mostly on the following three aspects:

- Bridging the gap between batch and real time methods;
- Proposing non-linear manifold methods to recover large and complex deformations;
- Allowing methods extension to handle the case with missing data in the measurements, e.g. due to occlusion or feature track loss.

The work conducted was targeted on "feature-based" method throughout the thesis, in which the feature points are detected and tracked in the images before the reconstruction process.

1.4 Contributions

This thesis presents a series of novel approaches for non-rigid shape and motion recovery, especially for complex deformations; for example articulated human motion movements and highly deformable surfaces. The main contributions of this thesis are summarised as follows:

- A new approach to estimate shape of deforming object using prior learned 3D defor-

mation shape model is proposed. The method has developed several extensions for this prototype algorithm. The proposed extensions include two aspects: constraints imposed on the basis shapes, the basic "building blocks" from which shapes are reconstructed; as well as constraints imposed on the mixing coefficients in the form of their probability distribution, which improves performance of the optimisation process.

- Building on this method, an incremental approach is presented for recursively recovering shape and motion. An important advantage of this method is the ability to reconstruct the 3D shape from a newly observed image and update the parameters in 3D shape space. This is motivated by the incremental principal component analysis (IPCA). The main novelty in our method is to propose an adaptive algorithm for construction of shape constraints improving stability of the on-line reconstructed shapes. Then the recursive algorithm is extended with additional step solving to the missing data problem (caused by self occlusion or tracking failure). The extended algorithm can efficiently handle the case of missing data in the measurements for both batch and incremental mode.

- Most of the existing approaches, including ours, assumed that 3D shapes can be accurately modelled in a linear subspace. The non-linear deformations are often observed in highly flexible objects for which the use of the linear model is impractical. The approach is proposed based on a non-linear manifold learning technique, called diffusion maps. Such manifold is used as a shape prior, with the reconstructed shapes constrained to lie in the manifold. This method achieves good results when dealing with objects undergoing significant and complex deformations. In the case of articulated deformations, e.g., full-body movement, rather than performing an initial segmentation stage on different body parts, the whole data are considered

as a single entity without the need for recognising different body parts. Instead, it learns a corresponding low dimensional manifold from the training examples. Such techniques have rarely been applied in the context of non-rigid shape reconstruction. Our approach integrates the learned non-linear shape prior manifold into the NRSfM solver. The advantage of our method is that it can be adopted for reconstruction of highly deformable, complex objects.

- Additional modification on the affinity model construction in manifold learning is made to use random forest clustering. The main advantage of using manifold forest compared to standard diffusion maps is the fact that in the manifold forest the neighbourhood topology is learned from the data itself, rather than being defined by the Euclidean distance.

- Although the manifold based approach significantly improves the reconstruction quality and is well-adapted to large deformation of complex objects, building a dense representation of the manifold requires a large amount of training data which is not feasible in many real applications. The manifold based method can be improved with the algorithm modifications, enabling reconstructions when only a small number of training samples are available and the measurements matrix is incomplete. The problem is addressed by grouping shapes into evolving clusters, with the shapes in each cluster represented in the linear subspace, estimated based on the observations and the prior learned manifold.

1.5 Thesis Outline

The remainder of the thesis is organised as follows. Several dominant approaches for 3D shape reconstruction are presented in Chapter 2, which provides a comprehensive review of current research. Chapter 3 gives a detailed description of the proposed linear

method, which uses standard PCA to obtain constraints on the basis shapes, as well as constrain on the values of the weighting coefficients. Inspired by this model, Chapter 4 presents a methodology which bridges the gap between current batch mode NRSfM and online NRSfM. A new method is proposed to update the model with regards to prior probability of the shape coefficients by applying IPCA. This is an incremental approach of estimating the deformable objects. Chapter 5 describes a non-linear manifold based reconstruction algorithm. We focus on using diffusion maps as a dimensionality reduction method to learn a non-linear shape prior. In Chapter 6 two improved versions of the algorithm described in Chapter 5 are proposed. The first is a new approach to build non-linear manifold by using random decision forests. The second includes modifications to the algorithm, which enable reconstructions when only small number of training samples is available and measurements matrix is incomplete. Performance analysis and discussion of the practical implementation issues of the proposed reconstruction algorithms is covered in Chapter 7. Chapter 8 provides comparison for all the proposed methods. Chapter 9 concludes the thesis, discusses potential additional improvements and gives suggestions as to the future directions of this research.

Chapter 2

Current Approaches to 3D Reconstruction

This chapter focuses on existing algorithms for 3D shape reconstruction. We start with a single view methodology, and then briefly introduce multiple views reconstruction research in the earlier research. We provide details of existing rigid factorisation frameworks, including probably the most successful, the Tomasi-Kanade factorisation algorithm and other commonly used approaches. Then, we present a number of algorithms which have been developed for reconstruction of deformable objects, including: low rank shape model, smooth trajectory model, manifold learning methods and other alternative methods. We also provide the literature on solving the missing data problem and sequential approaches.

2.1 Shape-from-X

Objects observed in a 2D image can be seen as a projection of the objects in the 3D world. One significant task in computer vision is to recover the depth information of the objects from single or multiple images. The study of how the shape of the objects can be inferred from several cues is known as "shape-from-X", where X represents different cues

including motion, shading, photometric, texture, blurring etc. [25]. Reconstruction can also be classified according to the number of images used for reconstruction.

2.1.1 Single view reconstruction

Shape reconstruction from a single image is possible, but cannot be done without prior knowledge related to the image scene [65]. The prior may involve camera information [32], or the geometric scene information, such as parallel lines [146] or vanishing points [26, 143]. The performance of shape from single-image cues can be improved by adding more constraints, e.g. applying shading or texture to infer shapes.

When using "shading" as a cue for shape reconstruction, estimating a 3D shape of a surface can be achieved using only a single image [70]. A comprehensive survey of shape from shading techniques was presented by Zhang et al., in which they compared four main different approaches and claimed that finding a unique solution to the problem is difficult, thus additional constraints are required [154]. Shape from texture can be understood as a problem of estimating the shape of the observed surface from a given image of a textured surface [136]. Moreover, different cues can be used together to produce an accurate geometric reconstruction [28]. For example, as demonstrated in [145], reconstruction from a single view using a combination of shading and texture by producing a normal estimate can minimises the error between the texture and the shading estimate.

2.1.2 Reconstruction from multiple views

Obviously using more images will bring more information for reconstruction; photometric stereo is a long studied technique which is based on the shading cue, but requires more than one image. The idea of reconstructing shapes by using three or more images taken under changing illumination conditions was originally introduced in [149].

Most previous work on photometric was developed for rigid objects [10, 67], while a non-rigid photometric stereo was presented recently in [66]. This algorithm is able to acquire, track and reconstruct the detailed deformable 3D shapes from video sequences of untextured data.

Using "motion" as a cue requires a sequence of images, and with an assumptions that the disparity between consecutive frames is small, otherwise it needs to be considered a "stereo-like" problem [136]. Our research focuses specifically on "motion", where the recovery of the 3D geometry is obtained from the spatial and temporal changes in an image sequence.

Our research aims to recover deformable objects from multiple images. We used optical camera as the only sensor with all the images captured by an uncalibrated single camera. Neither the shape of the objects nor the camera information is known in advance. The literature on estimation of 3D shape and motion is immense. In recent decades, a large number of algorithms and techniques have been proposed to solve this problem. We present the description of some of these algorithms.

2.2 Rigid Structure from Motion

Structure and motion recovery from image sequences is an active area of research in the computer vision community. It usually requires certain assumptions on the camera and scene in order to simplify the problem. Most of the work focused on static scene or rigid objects, which implies that the shape of the object is not changed or deformed, thus the reconstructed results can be gradually refined.

The projective reconstruction problem

When a scene is observed by human eyes, the distance objects appear smaller than the objects which are close to the eyes.This is known as perspective. Perspective camera model is the most common geometric model of a camera. For example, parallel lines in an image may not be seen as parallel, instead they are distorted by a projective transformation.

Perspective reconstruction has been successfully applied where the object model was assumed rigid. The reconstruction process consists of two main steps: projective reconstruction and Euclidean reconstruction. The first step is to recover the projective shape and motion from the measurement data only; and the second step usually imposes the rank constraints to obtain Euclidean structure.

After the seminal work of self-calibration [44], Sturm and Triggs [124, 133] described a non-iterative factorisation method for uncalibrated cameras. According to the pairwise constraints among images, this approach uses only epipoles and a set of fundamental matrices to estimate the scaled image measurements. But the result of this algorithm strongly depends on the accuracy of the epipolar geometry. An error in the estimation of fundamental matrix and epipoles would affect the reconstruction results. Han and Kanade [59] presented an alternative method using bilinear projective factorisation algorithm; this iteratively improves the depth information, eliminating the need for calculation of fundamental matrices. Mahamud and Hebert [83] also proposed an iterative method which concurrently recovered the projective depths, together with structure and motion.

To recover the Euclidean shape from the projective reconstruction, Hartley [64] presented a global optimisation by assuming the camera is intrinsic parameters were unchanged throughout the sequence. Although this method has shown to directly

recover structure and camera parameters, the complicated non-linear optimisation process requires a reliable initial estimation. To improve this, the method was further studied in [68, 134, 105, 69], where different additional constraints on either the camera or the scene are needed. The first complete theoretical convergence analysis for the iterative algorithms was provided by Oliensis and Hartley [98], where they proved that the previous methods may not converge to useful minima, and also proposed an iterative extension of [124] which effectively avoids this problem.

An investigation of different camera models is presented in [63]. Using the full perspective camera model can indeed help to obtain a correct 3D reconstruction of the object, but too many unknown variables lead to an under constrained problem. However, in some cases perspective projection model is unnecessary if the range of object depth is relatively small compared to the distance between camera centre and the object.

The affine reconstruction problem

In certain cases when the depth variation of an object is much smaller than the distance between the object and the camera, the perspective camera model can be approximated as an affine camera [65]. Affine camera model includes orthographic, weak perspective and paraperspective projections. Most factorisation based SfM techniques begin with the assumption of an affine camera model. Using an orthographic projection model can greatly simplify the problem since recovery of the camera intrinsic parameters is no longer required.

A direct solution for recovery of both motion and structure of the object is the classical algorithm of point based SfM with factorisation. Tomasi and Kanade [129] first proposed a factorisation algorithm based on the Singular Value Decomposition (SVD), which was used for the reconstruction of a rigid object under an orthographic

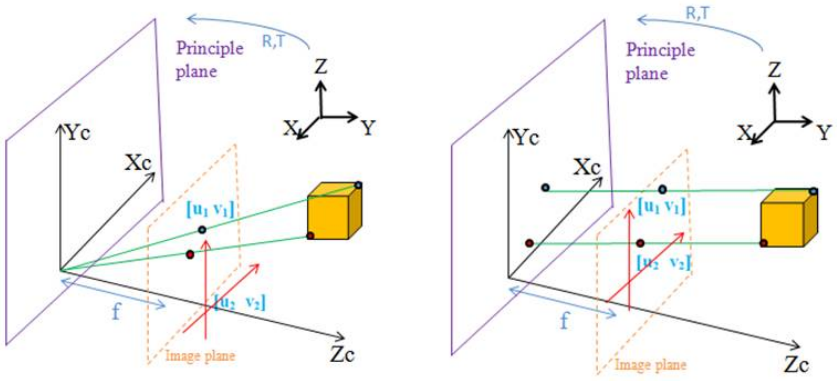


Figure 2.1: Two camera models. (Left) Perspective projection model. (Right) Orthographic projection model.

camera model. In this, the algorithm factorises the measurement matrix into shape and rotation matrices under a rank constraint. Since then, techniques for rigid shape recovery via point based SfM have achieved maturity over the following decades [47, 84, 125]. Subsequent work focused on a factorisation approach applied to multiple rigid objects [60].

For dealing with dynamic scenes, Costeira and Kanade first presented a method for separating and recovering the motion and shape of multiple independently unknown number of moving objects in a sequence of images [31]. Han and Kanade followed the idea but with consideration of degenerate cases [59], and assumed that objects are moving linearly with constant speed. The method in [8] did not require constraints on moving speed, but assumed that the object has to move along a line.

Rigid objects may be linked by joints, such as the human body, hand gesture etc. [151]. The factorisation method was first extended to the case of articulated object reconstruction in [132]. Unlike multiple moving objects, the relative motion of articulated objects is interlinked, thus the dependency can be seen as articulated constraints which should be incorporated from the beginning.

2.2.1 Problem formulation

Given a point in a world coordinate system, denoted as $\mathbf{s}_p = [x_p, y_p, z_p]^T$ and transformed into the t^{th} image coordinate system through rotation $\mathbf{R}_t$ and translation $\mathbf{t}_t$, its orthographic projection $\mathbf{x}_{tp}$ onto t^{th} image, is given by:

$$\mathbf{x}_{tp} = \begin{bmatrix} u_{tp} \\ v_{tp} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t | \mathbf{t}_t \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s}_p \\ 1 \end{bmatrix} = \begin{bmatrix} r_{t1} & r_{t2} & r_{t3} & t_{xt} \\ r_{t4} & r_{t5} & r_{t6} & t_{yt} \end{bmatrix} \cdot \begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix}^T \tag{2.1}$$

where $\mathbf{x}_{tp}$ represents the p^{th} 3D point $\mathbf{s}_p$ projected onto t^{th} image; the orthographic camera matrix $\mathbf{R}_t$ only encodes the first two rows of rotation matrix with rotation constraint $\mathbf{R}_t\mathbf{R}_t^T = \mathbf{I}$. It can be seen that when $\mathbf{x}_{tp}$ are given with respect to the origin at the centre of gravity calculated for all projected points in the t^{th} frame, $\mathbf{t}_t = [t_{xt}\ t_{yt}]^T = 0$.

Considering P feature points tracked in F video frames, the $2F \times P$ observation matrix can be expressed as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \mathbf{x}_{tp} & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_P \end{bmatrix} = \mathbf{MS} \tag{2.2}$$

where $\mathbf{M}$ is a stack of motion (rotation) matrices representing camera orientation for each frame and $\mathbf{S}$ represents all 3D feature points on reconstructed objects concatenated into a single matrix. Each of the columns in measurements $\mathbf{W}$ represents the trajectory of a single feature point across all frames and each of the rows represents all of the feature points in a single frame. Now the problem can be summarised as to estimate appropriate shape $\mathbf{S}$ and motion $\mathbf{M}$ only from a set of 2D image trajectories $\mathbf{W}$. The two most often used approaches are factorisation algorithm and bundle adjustment.

2.2.2 Tomasi-Kanade factorisation algorithm

To reconstruct a rigid object or a static scene, factorisation is a long-standing and well-known algorithm. Kanatani and Sugaya provided comprehensive descriptions and complete derivation of this technique [74]. Given its simplicity this is widely exploited in many applications and also frequently used as a first step in an optimization procedure designed to reconstruct time-varying shape structure.

One of the best known approaches for rigid object based on factorisation technique has been developed by Tomasi and Kanade [129] in the early 90s. They factorised measurement matrix into two factors, shape and motion matrix, under the rank theorem described in [129]. These two factors can be described as a bilinear model which has lower dimensionality if compared with data space. Tomasi and Kanade's factorisation method is sometimes misunderstood as reconstructing 3D by matrix factorisation using SVD. In reality, it is only an affine approximation to the camera and shape matrix and the real resulting matrices are obtained by imposing orthonormality of the rotation matrices. Factorisation by SVD is nothing but a means for numerically computing the least-squares solution [74].

Suppose first two rows of camera rotation at time t can be represented as a pair of unit vectors, $\mathbf{i}_t$ and $\mathbf{j}_t$, pointing the orientations of the horizontal and vertical camera reference axes throughout the images, then the motion $\mathbf{M}$ in Equation 2.2 can be written as,

$$\mathbf{M} = [\mathbf{i}_1 \cdots \mathbf{i}_F, \mathbf{j}_1 \cdots \mathbf{j}_F]^T \tag{2.3}$$

According to the rank theorem [129], $\mathbf{M}$ is a $2F \times 3$ matrix and the size of shape $\mathbf{S}$ in Equation 2.2 is $3 \times P$ which implies that the measurements $\mathbf{W}$ is at most rank 3 with absence of noise. Because $\mathbf{i}_t$ and $\mathbf{j}_t$ are mutually orthogonal unit vectors, so they

must satisfy the constraints with,

$$|\mathbf{i}_t| = |\mathbf{j}_t| = 1, \text{ and } \mathbf{i}_t^T \mathbf{j}_t = 0 \tag{2.4}$$

To keep the rotation matrix unique, the first camera reference system is aligned with the world reference system, therefore the unit vectors $\mathbf{i}$ and $\mathbf{j}$ in the first frame can be written as $\mathbf{i}_1 = (1, 0, 0)^T$ and $\mathbf{j}_1 = (0, 1, 0)^T$. By applying the rank constraint, the measurement matrix $\mathbf{W}$ is initially decomposed into two terms, affine motion $\hat{\mathbf{M}}$ and affine shape $\hat{\mathbf{S}}$, using rank-3 truncated SVD decomposition,

$$\text{SVD} : \mathbf{W} \approx \mathbf{U}^{2F\times 3}\mathbf{D}^{3\times 3}\mathbf{V}^{3\times P} = (\mathbf{U}\mathbf{D}^{1/2})(\mathbf{D}^{1/2}\mathbf{V}) = \hat{\mathbf{M}}\hat{\mathbf{S}} \tag{2.5}$$

The affine motion $\hat{\mathbf{M}}$ and affine shape $\hat{\mathbf{S}}$ have the same size as desired motion $\mathbf{M}$ and shape $\mathbf{S}$. In fact, the affine solution is a linear transformation of desired solution, and therefore the decomposition is not unique, any 3×3 invertible matrix $\mathbf{Q}$ can satisfy the following equation,

$$\hat{\mathbf{M}}\hat{\mathbf{S}} = (\hat{\mathbf{M}}\mathbf{Q})(\mathbf{Q}^{-1}\hat{\mathbf{S}}) = \mathbf{M}\mathbf{S} \tag{2.6}$$

To solve the inherent ambiguity in the factorisation, metric constraints are imposed to find a unique $\mathbf{Q}$. Based on the constraints in Equation 2.4, it is possible to calculate $\mathbf{Q}$ by solving the following over-constrained, non-linear data fitting problem,

$$\begin{aligned} \mathbf{i}_t^T \mathbf{Q}\mathbf{Q}^T \mathbf{i}_t - \mathbf{j}_t^T \mathbf{Q}\mathbf{Q}^T \mathbf{j}_t = 0, \\ \mathbf{i}_t^T \mathbf{Q}\mathbf{Q}^T \mathbf{j}_t = 0. \end{aligned} \tag{2.7}$$

Once $\mathbf{Q}$ has been determined, the desired motion and shape can be easily computed as,

$$\mathbf{M} = \hat{\mathbf{M}}\mathbf{Q} \text{ and } \mathbf{S} = \mathbf{Q}^{-1}\hat{\mathbf{S}} \tag{2.8}$$

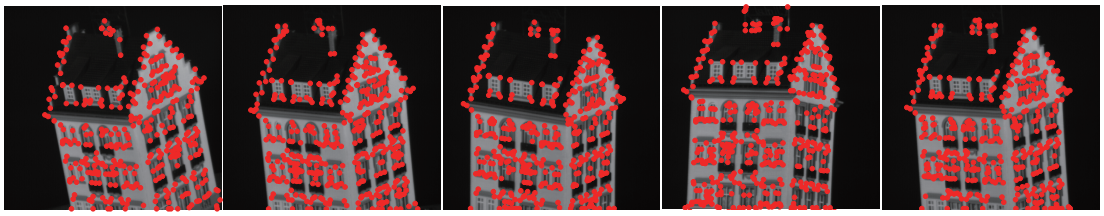


Figure 2.2: Extracted frames with tracked feature points from "Hotel" sequence.

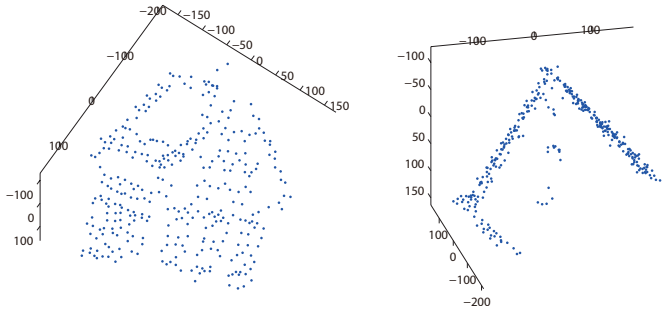


Figure 2.3: (Left) Front view of the output rigid 3D reconstructed shape. (Right) Top view of the output rigid 3D reconstructed shape

As the solution is determined up to a rotation of the reference system, the fist frame should be aligned with the world reference system.

Experimental results

We reproduce the experiment which was originally presented in [129]. The "Hotel" sequence is obtained from CMU database [1]. The feature points are extracted and tracked using the Kanade-Lucas-Tomasi (KLT) feature tracker [82, 128]. Figure 2.2 shows the extracted frames with the tracked feature points from the sequence of images. Figure 2.3 shows both front view and top view of 3D reconstructed results obtained by the factorisation algorithm.

2.2.3 Projective factorisation

Affine model can be seen as a special case of perspective projective model. When the camera is close to the observed object or the scene has significant depth, the orthographic or weak-perspective projection model no longer approximates the problem. The perspective effect will lead the existing methods to produce distorted reconstruction results.

Under the perspective projection, a 3×4 camera matrix at time t is defined as:

$$\mathbf{P}_t = \begin{bmatrix} f_x & \alpha & u_c \\ 0 & f_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_t & \mathbf{t}_t \end{bmatrix} \tag{2.9}$$

where f_x, f_y are focal length in width and height, $[u_c, v_c]^T$ are the coordinates of the cameras principal point.

Suppose $\mathbf{s}_p$ is an unknown homogeneous coordinate vectors of a 3D point, $\mathbf{P}_t$ is the unknown projection matrix. The image projection equation projects $\mathbf{s}_p$ onto the image at time t is,

$$\lambda_{tp}\mathbf{x}_{tp} = \mathbf{P}_t\mathbf{s}_p \tag{2.10}$$

where the unknown scaling factor λ_{tp} is projective depth. The complete set of all the points in all the perspective frames, together with their corresponding projective depth can be gathered into a single $3F \times P$ measurement matrix,

$$\mathbf{W} = \begin{bmatrix} \lambda_{11}\mathbf{x}_{11} & \cdots & \lambda_{1P}\mathbf{x}_{1P} \\ \vdots & \ddots & \vdots \\ \lambda_{F1}\mathbf{x}_{F1} & \cdots & \lambda_{FP}\mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_F \end{bmatrix} [\mathbf{S}_1 \cdots \mathbf{S}_P] = \mathbf{PS} \tag{2.11}$$

with the correct projective depths λ, the rescaled measurement matrix $\mathbf{W}$ has rank at most 4, since shape matrix $\mathbf{S}$ and motion matrix $\mathbf{P}$ are at most rank 4. If the true projective depths λ_{tp} are known, it is possible to follow the factorisation method, which is similar to the orthographic case. Sturm and Triggs [124] described a non-iterative factorisation method for uncalibrated camera. This method only needs to estimate the fundamental matrices $\mathbf{F}$ and epipole components e using pairwise images and then recursively chained together to calculate the equation with $\lambda_{1p} = 1$,

$$\lambda_{(t+1)p} = \frac{\left(e_{t(t+1)} \wedge \mathbf{x}_{(t+1)p}\right) \cdot \left(\mathbf{F}_{t(t+1)}\mathbf{x}_{tp}\right)}{\left\|e_{t(t+1)} \wedge \mathbf{x}_{(t+1)p}\right\|^2}\lambda_{tp} \tag{2.12}$$

Once the projective depths are obtained, it is possible to factorise projective shape and motion from rescaled measurement matrix by SVD. But unlike the orthographic projection, there are no further rotation constraints here for projection matrix $\mathbf{P}$. Thus to find linear transformation, the constraints can be either added to the projection matrix [59] or shape matrix [65].

2.3 Non-Rigid Structure from Motion

To extend the rigid SfM into the case of 3D non-rigid objects [2], the seminal work proposed by Bregler et al. in [19] was the first to represent shapes as a linear combination of a set of basis shapes which describes the main modes of deformation. Those basis shapes are unknown but fixed for each sequence. The 2D measurement matrix has been factorised into shape coefficients, a camera motion matrix and 3D basis shapes using SVD, which is similar to the method proposed in [129]. This low rank shape model has been widely used in the non-rigid and articulated object reconstructions. Representing the 3D deformable shape as the linear combination of 3D basis shapes is called 3D morphable model. Such model was successfully used to obtain full 3D

models of faces in [17], where the 3D model of the shape was built by using a large face database as a priori. Such model was originally inspired by the 2D active shape model which later extended to model human facial expressions of the same face. Following this shape model, factorisation for articulated NRSfM was proposed in [101], but small inaccuracies in the affine values obtained from the initial affine decomposition greatly affect the subsequent estimation process. Xiao et al. [150] proposed a closed-form solution and demonstrated an ambiguity in orthonormality constraints that using only orthonormality constraints is insufficient to provide unique solutions to estimated structures. They employed the traditional orthonormality constraints, but also introduced additional constraints to further determine shape basis, however this method does not cope well with noisy data. To overcome this, iterative optimisation methods based on bundle adjustment were introduced in [144] as a last step of reconstruction, in order to improve the quality of the estimation. Recent approaches have focused on solving problems related to the inherently large number of degrees of freedom, which together with motion degeneracy (very limited camera motion during data acquisition) may eventually result in worthless reconstructions.

2.3.1 Problem formulation

In the case of non-rigid objects, the 3D shapes deform over time, which makes the problem more difficult to solve. Considering a set of 2D images viewed by an orthographic camera, tracking P feature points in F video frames, the measurement matrix can be formed as,

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \mathbf{x}_{tp} & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_F \end{bmatrix} \begin{bmatrix} -\mathbf{S}_1- \\ \vdots \\ -\mathbf{S}_F- \end{bmatrix} = \mathbf{RS} \tag{2.13}$$

The problem consists of shapes $\{\mathbf{S}_1, \dots \mathbf{S}_F\}$ and the camera rotation $\{\mathbf{R}_1, \dots, \mathbf{R}_F\}$ recovery from the 2D observations $\mathbf{W}$, thus can be formulated as the following optimisation problem,

$$\arg\min_{\mathbf{R}_t, \mathbf{S}_t} \sum_{t=1}^{F} \|\mathbf{W}_t - \mathbf{R}_t \mathbf{S}_t\|^2 \tag{2.14}$$

where $\mathbf{W}_t$ represents the 3D points projected onto t^{th} image. The camera translation can be eliminated, by expressing 2D observations with respect to the data points centroid calculated in each observed image. It is obviously an under constrained problem since shape and motion are both changing with time. Describing the deformation using F independent shapes $\mathbf{S}_t = [\mathbf{s}_{t1} \cdots \mathbf{s}_{tP}]$, with $\mathbf{s}_{tP}$ representing coordinates of the n^{th} 3D feature point in frame t may entail more unknown variables $(3F + 3FP)$ than the number of observed input data $(2FP)$ from the observation. However, it is clear that motion is not random; feature points are highly correlated in time and space. Therefore, an object is unlikely to deform completely arbitrarily over time. To deal with this, low rank shape model and smooth trajectory model are two major approaches to determine a structure which lies in a lower dimensional subspace.

2.3.2 Low rank shape model

Using a low rank shape model to represent the non-rigid structure is one way of reducing dimensionality of the problem [19]. A linear combination of K basis shapes, $\mathbf{B}_d$, could be used to mathematically represent a morphable 3D model represented in each frame,

$$\mathbf{S}_t = \alpha_{t1}\mathbf{B}_1 + \alpha_{t2}\mathbf{B}_2 + \cdots + \alpha_{tK}\mathbf{B}_K = \sum_{d=1}^{K} \alpha_d \mathbf{B}_d \tag{2.15}$$

where basis shapes $\mathbf{B}_d$ are unknown but fixed, whilst deformation coefficients α_d are adjustable over time. Figure 2.4 illustrates an example of a deformable model. As shown "symbolically" in the figure, second basis shape provides a greater contribution

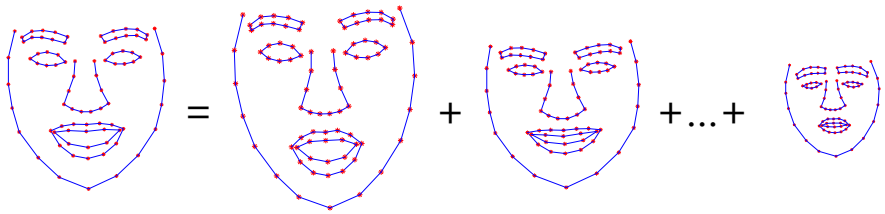


Figure 2.4: A graphical representation of the deformable shape model as a weighted superposition of several basic shapes (shown shapes do not represent a true appearance of the basic $\mathbf{B}_i$). The size of the shape visually encodes the corresponding shape's weight

than any other basic shape. The whole shape matrix $\mathbf{S}$ can be rearranged as:

$$\mathbf{S} = \begin{bmatrix} -\mathbf{S}_1- \\ \vdots \\ -\mathbf{S}_F- \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1K} \\ \vdots & \ddots & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FK} \end{bmatrix} \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} \tag{2.16}$$

To deal with the case of non-rigid shapes under orthographic camera model, a low rank shape model has proved a successful representation. The advantage of this approach is that it can tackle the problem without any prior information about the object or the scene, or any other multiple views and 3D input. The core of this method is to express the measurement matrix as a trilinear product of three matrices: pose, basic models and time varying coefficients. Given that,

$$\begin{aligned} \mathbf{W} &= \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_F \end{bmatrix} \left(\begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1K} \\ \vdots & \ddots & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FK} \end{bmatrix} \otimes \mathbf{I}_3 \right) \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} \\ &= \begin{bmatrix} \alpha_{11}\mathbf{R}_1 & \cdots & \alpha_{1K}\mathbf{R}_1 \\ \vdots & \ddots & \vdots \\ \alpha_{F1}\mathbf{R}_F & \cdots & \alpha_{FK}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} -\mathbf{B}_1- \\ \vdots \\ -\mathbf{B}_K- \end{bmatrix} = \mathbf{MB} \end{aligned} \tag{2.17}$$

where $\otimes$ is the Kronecker product, motion $\mathbf{M}$ is a $2F \times 3K$ matrix which contains rotations $\mathbf{R}_t$ with weighting factors α_{tp} and basis shapes $\mathbf{B}_d$ have size $3K \times P$, the rank of $\mathbf{W}$ must be at most $3K$ in the absence of noise. The variables in Equation 2.17 can be estimated by minimising the following reprojection error:

$$\underset{\alpha_{td},\mathbf{R}_t,\mathbf{B}_d}{\arg\min} \sum_{t=1}^{F} \left\| \mathbf{x}_t - \mathbf{R}_t \sum_{d=1}^{K} \alpha_{td} \mathbf{B}_d \right\|^2 \tag{2.18}$$

As K is usually a relatively small number, in this formulation the total $3F + FK + 3KP$ number of parameters is much smaller than given $2FP$ coordinates, which makes the problem under-constrained.

Non-rigid factorisation algorithm

The factorisation algorithm presented by Bregler et al. [19] was the first that can tackle the non-rigid object reconstruction problem without the use of prior information, multi-camera or other 3D input. They demonstrated how 3D deformable objects, such as human faces and animals, can be recovered from image streams taken by a single camera by solving multiple factorisation steps.

The first step is to compute shape bases $\mathbf{B}$ by factorising the measurements $\mathbf{W}$. Equation 2.17 shows that the measurement $\mathbf{W}$ has rank at most $3K$ and can be factorised into 2 matrices: Basis shape $\mathbf{B}$ and motion matrix $\mathbf{M}$ contain camera rotation $\mathbf{R}_t$ and deformable coefficients α_{tp}. Using SVD by only considering the first $3K$ singular vectors and its corresponding singular values, the factorisation can be done as,

$$\text{SVD} : \mathbf{W} \approx \mathbf{U}^{2F \times 3K} \mathbf{D}^{3K \times 3K} \mathbf{V}^{3K \times P} = \hat{\mathbf{M}} \hat{\mathbf{B}} \tag{2.19}$$

The solution is not unique and is defined up to an ambiguity matrix $\mathbf{G} \in \mathbb{R}^{3K \times 3K}$,

such as $\mathbf{W} = (\hat{\mathbf{M}}\mathbf{G})(\mathbf{G}^{-1}\hat{\mathbf{B}}) = \mathbf{MB}$.

The second step is to extract the rotation and the coefficient of each basis shapes from motion matrix $\mathbf{M}$. The row $(2t-1)$ and $2t$ in $\mathbf{M}$ are the two rows that correspond to the frame t, which can be rearranged and factorised in the following form,

$$\mathbf{m}_t = \begin{bmatrix} \alpha_{t1}r_{t1} & \alpha_{t1}r_{t2} & \cdots & \alpha_{t1}r_{t6} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{tK}r_{t1} & \alpha_{tK}r_{t2} & \cdots & \alpha_{tK}r_{t6} \end{bmatrix} = \begin{bmatrix} \alpha_{t1} \\ \vdots \\ \alpha_{tK} \end{bmatrix} \begin{bmatrix} r_{t1} & r_{t2} & r_{t3} & r_{t4} & r_{t5} & r_{t6} \end{bmatrix} \quad (2.20)$$

According to the metric constraint presented in Equation 2.7, the linear transformation $\mathbf{Q}$ can be found by enforcing orthonormality of all rotations.

The limitation of this approach is that the motion matrix is non-linear; when an inaccurate set of basis shapes have been chosen, it may not be possible to remove the affine ambiguity. Besides, the method is very sensitive to noise since it strongly relies on rank theorem, which leads to reconstruction fail for the object with large deformations. However, this method is still effective to provide initialisation solutions for other approaches [18, 131].

The original work of non-rigid factorisation [19] has utilised only the orthonormality constraints on camera rotations to solve the problem. However, enforcing only the rotation constraints may lead to ambiguity that the shape bases are not unique and cannot guarantee the desired solution. To improve this, Xiao et al. proposed a set of novel constraints by enforcing both the basis and the rotations [150]. According to their closed-form solution, substituting $\mathbf{G}_d = \mathbf{H}_d\mathbf{H}_d^T$, the constraints are written as,

$$\begin{cases} \hat{\mathrm{M}}_{2t-1}\mathrm{G}_d\hat{\mathrm{M}}_{2t-1}^T - \hat{\mathrm{M}}_{2t}\mathrm{G}_d\hat{\mathrm{M}}_{2t}^T = 0, & t = 1:F \\ \hat{\mathrm{M}}_{2t-1}\mathrm{G}_d\hat{\mathrm{M}}_{2t}^T = 0, & t = 1:F \end{cases} \quad (2.21)$$

$\mathbf{H}_d$ can be determined by using SVD or other decomposition algorithm, once $\mathbf{G}_d$ has been obtained.

Since NRSfM is an ill-posed problem, using additional constraints can help to solve the problem of upgrading the metric space. Some representative work such as Bartoli et al. [9] introduce prior information based on coarse-to-fine scheme and compose low-rank shape model with euclidean transformations. However Dai et al. argue that these additional constraints are not necessary and limit the practical applicability of the methods [34]. Thereby they proposed a simple method without assuming any extra prior constraints, by implementing semi-definite programming of trace minimisation problem. However inherent prior knowledge has still been used such as the method is based on low-rank shape model, and the number of basis shapes are still required for the metric upgrade step.

2.3.3 Smooth trajectory model

Although the majority of works use the low rank shape model and achieve successful results, an obvious drawback of this model is the shape basis are different in each sequence, thus needs to be estimated for every sequences. Besides, for more complex deformable shapes, such as inextensible surfaces or elastic objects, a large number of basis shapes are required to fit the model. Figure 2.5 illustrates representative shape and trajectory space.

Akhter et al.'s original work

According to the duality theorem, as described in [5], representing a non-rigid shape using the above shape basis model is dual to the trajectory basis model, in which each point trajectory is represented as a K dimensional point within an unknown linear trajectory space. The trajectory for each point is approximated by a linear combination

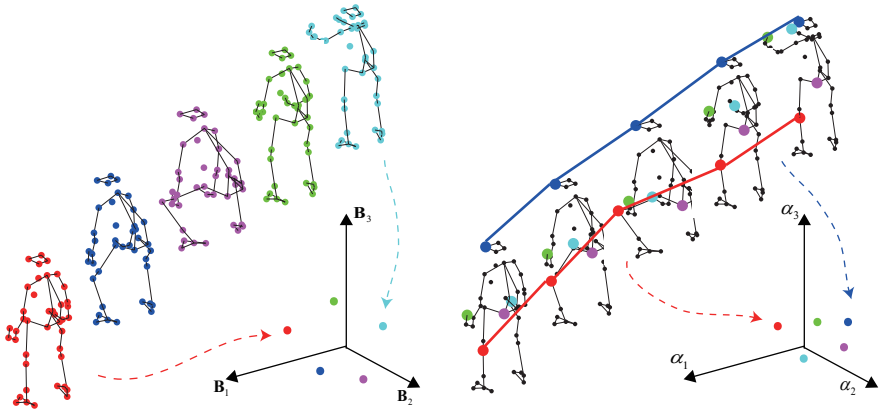


Figure 2.5: An example of shape and trajectory space. Left: Each point in shape basis space comes from independent shapes. Right: Each point in trajectory basis space comes from trajectory of each point over the whole sequence.

of a small number of basis trajectories. The basis trajectory can be predefined in an object independent way using K low-frequency Discrete Cosine Transform (DCT) basis and therefore avoid training for the bases.

Akhter et al.'s work proposed a factorisation approach but using the basis trajectory model. This allows the Equation 2.17 to be rewritten as, $\mathbf{W} = (\mathbf{D}\Theta)\mathbf{B} = \mathbf{MB}$, where $\mathbf{D}$ is a block-diagonal rotation matrix and $\mathbf{\Theta}$ contains basis vectors of the time-trajectory of 3D points,

$$\mathbf{D} = \begin{bmatrix} \mathbf{R}_1 & & \\ & \ddots & \\ & & \mathbf{R}_F \end{bmatrix}, \mathbf{\Theta} = \begin{bmatrix} \boldsymbol{\omega}_1^T & & & \\ & \boldsymbol{\omega}_1^T & & \\ & & \boldsymbol{\omega}_1^T & \\ \boldsymbol{\omega}_F^T & \vdots & & \\ & \boldsymbol{\omega}_F^T & & \\ & & \boldsymbol{\omega}_F^T & \end{bmatrix} \tag{2.22}$$

where the f^{th} column of $\boldsymbol{\omega}$ is the f^{th} frequency cosine wave with entries,

$$\omega_{tf} = \frac{\sigma_f}{\sqrt{F}} \cos\left(\frac{\pi(2t-1)(f-1)}{2F}\right), t = 1 \dots F, f = 1 \dots K \tag{2.23}$$

where $\sigma_1 = 1$ and, for $f \geq 2, \sigma_f = \sqrt{2}$. The model only needs to consider camera parameters and trajectory coefficients, thus requires less parameters than shape basis model, see section 5.3. In this work, the rotation matrix $\mathbf{D}$ is recovered first using Euclidean upgrade step. Once rotations are determined and basis trajectories are predefined in advance, the trajectory coefficient matrix $\mathbf{B}$ can be easily obtained.

However, because of the rank constraint (the measurement matrix has at most $3K$), the method cannot model high-frequency deformation. This may result in over-smoothed solutions, and therefore this method is restricted to a model with slow and smooth deformation.

Alternative trajectory model

Following Akhter et al.'s baseline algorithm, several alternative methods for computing the DCT coefficients in the model are presented. Gotardo and Matinez proposed an effective way of using higher-frequency DCT components without increasing the factorisation rank [53]. The method describes a smooth shape trajectories approach which models 3D shapes instead of independent 3D point trajectories of [5]. Thus the shape coefficient matrix containing α in Equation 2.17 can be rewritten as a linear combination of a small number (d in this case) of low-frequency DCT basis vectors,

$$\begin{bmatrix} \alpha_{11} & \cdots & \alpha_{1K} \\ \vdots & \ddots & \vdots \\ \alpha_{F1} & \cdots & \alpha_{FK} \end{bmatrix} = \mathbf{\Omega}_d \mathbf{X}, \ \mathbf{X} \in \mathbb{R}^{d \times K} \tag{2.24}$$

where $\mathbf{\Omega}_d$ is a DCT basis matrix with entries as in Equation 2.23. So the only unknown is $\mathbf{X}$ which describes the 3D shape trajectory in DCT domain. This method can also solve the rigid structure from motion problem by using the DCT basis to model camera trajectory.

2.3.4 Manifold learning approaches

Most of the existing methods are restricted by the fact that they try to explain complex deformations using a linear model. Recent methods have integrated the manifold learning algorithm to regularise the shape reconstruction problem, by constraining the shapes as to be well represented by the learned manifold. Rabaud and Belongie firstly claimed in their work [109] that the possible 3D shapes of an object may not lie on a linear low-dimensional manifold. Based on the low rank shape model, that work assumed that shapes lie on a d-dimensional manifold, and every neighbourhood of shape approximately lies on a d-dimensional linear subspace. In order to minimise the cost function which consists of the reprojection error and smoothing terms, the initial values are calculated by Rigid Shape Chain, in which sequences are clustered as several rigid shapes. After initialisation, the optimisation of the shapes is performed using two criteria: the cost function, and the shape manifold dimensionality constraint for which Locally Smooth Manifold Learning technique has been used. Later they proposed a method focusing on a globally linear manifold and used shape embedding as initialisation [110].

Other manifold based methods departed from the basis trajectory model. Gotardo and Martinez demonstrated the "kernel trick", which used for non-linear dimensionality reduction [119] can also be applied to standard NRSfM problem [52]. Recently Hamsici et al. [57] modelled the shape coefficients in a manifold feature space. This method has the ability to recover the shapes from a newly observed image. The mapping was learned from the corresponding 2D measurement data of upcoming reconstructed shapes, rather than a fixed set of trajectory bases. They introduced Rotation Invariant Kernels (RIK) to provide similarity measure for two 3D shapes based on their 2D projections which can eliminate the fact that two frames are taken from different points of view. But

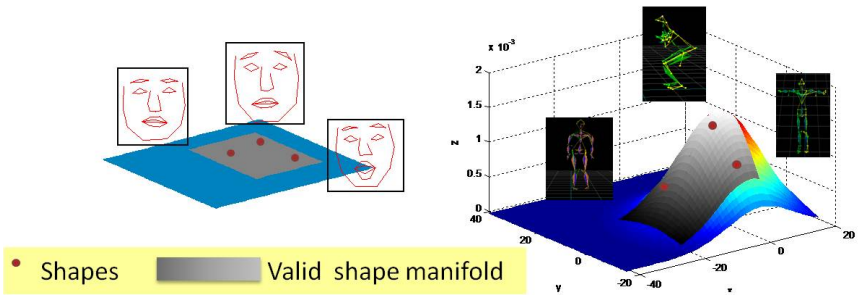


Figure 2.6: (left): Representation of the linear subspace shape model; (right): manifold interpretation of shapes with complex deformation

the problem is the 2D observations can be completely different when the images are taken from different angles of view. Meanwhile, because of different depths, similar 2D images may not represent similar 3D shapes. In comparison, [52] defines a non-linear model while [57] models 3D shapes in a linear space; [52] uses point trajectory bases as input data for building a kernel function, while [57] directly uses shapes from 2D images.

2.3.5 Other methods

Template based reconstruction is an alternative method which usually relies on a known reference frame and works well, especially for reconstruction of inextensible surfaces. Reconstruction is achieved from input images and a reference image, for which the corresponding 3D shape of the object is known. Since this is still an ill-conditioned problem [99], the most commonly used constraints in the reconstruction involve preserving either Euclidean or Geodesic distances as the surface deforms, thus it regularises the problem by solving either the convex optimisation problem [117, 23, 89] or in closed form sets of quadratic equations [118, 90]. This inextensibility prior of deformable surface has been extensively used for template based reconstruction and shown to be a sensible constraint for many shapes [141], including the human body and different types

of animals. However, the way to compute 3D template is difficult and sometimes even impossible.

Notice that the restriction of most existing NRSfM methods is that they try to explain the complex deformations using a global model. An alternative *piecewise model* has been recently developed [126, 139, 45]. Piecewise approaches mainly attempt to solve dense NRSfM problem. As a single shape can be approximated by a series of patches, they divided the surface into overlapping planer [139] or regular patches [45], then individually reconstructed them. This model is able to cope well with strongly deforming objects. However, necessity for dividing the surface into a set of overlapping patches (often preformed manually) is generally viewed as the severe drawback of this model.

The most recent dense NRSfM method is proposed by Garg et al. in [48], in which they provide robust dense 3D estimation for every pixel in the reference image of a deformable shape using only the original footage. The method departs from a trace norm minimisation approach similarly to [34], but using a multi-frame motion flow field as input.

2.4 Articulated object reconstruction

Articulated motion is one significant problem in structure from motion and has been studied since the last decade [132, 151, 101]. Sinclair et al. derived a direct constraint for recovery of Euclidean structure for articulated objects using perspective projection camera under the assumption that the objects were coupled by a hinge (two objects are coupled by one degree of freedom) [122]. For objects coupled by a universal joint (two objects are linked by two or three degrees of freedom, their rotations are independent), a direct extension of factorisation algorithm to the articulated object reconstruction

was proposed by Tresadern and Reid in [132] where they look at articulated objects that cannot be represented by a single statistical shape model. Their work shows how to segment the objects in order to group feature tracks and determine the type of coupling between two objects.

One particularly interesting problems in articulated motion is human motion analysis. The applications of estimating the 3D pose are completely different between biomechanical modelling, diagnosis and rehabilitation and to the human motion capture used in movies and video games. At the early stage of research on human motion reconstruction, different parts are approximated as a set of rigid articulated links [132, 101] in order to simplify the problem. Recently, research has increasingly moved to more difficult cases of this problem, when the objects are articulated while at the same time change shapes. Non-rigid articulated structure representation has also been formulated following the idea of probability model [114] and piecewise model [45]. For all of the methods, the most challenging part is recognition of the different parts of the articulated objects, for which the quality of segmentation directly leads to the reconstruction results. So, rather than having an initial segmentation stage to assign motion as a set of intersecting motion subspaces which may lead to unexpected errors, in our methods the whole data can be considered as a single entity without the need for body part recognition.

2.5 Reconstruction with missing data

Most algorithms assume that the input measurement matrix is complete, with all the feature points detected in all the images. This is unlikely to happen in practice, as some of the feature points will not be detected in all the images. This could be because of the feature point detection problems or because some parts of the 3D object may not be

visible from all camera positions. This means some of the entries in the measurement matrix may be unknown. This makes the shape reconstruction more challenging. The methods addressing this problem can be divided into three categories: imputation, alternation and non-linear optimisation.

Imputation algorithms attempt to fill in the missing entries using a complete subset of the data [129, 153]. The original method was presented in [129], where the authors believe that the information in partially filled measurements is sufficient to determine all the feature points and camera positions. The work in [153] shows how to impute missing data in non-rigid reconstruction problem. Their model is based on smooth trajectory assumption, which can handle various levels of missing observations. In practice, these methods are simple but cannot handle real data, which often tend to be very noisy. In spite of this, imputation algorithm is still sufficient to provide initial estimation for alternation and non-linear optimisation algorithms.

Alternation algorithms solve the problem based on closed-form solution, using a rank constraint imposed on the measurement matrix without estimating the missing values in advance [101, 84]. The algorithms relied on observation and required either motion or shapes to be known [24]. Most existing methods for this problem followed this idea by iteratively updating motion and shape in terms of observed measurements [52]. Note that optimising the complete matrix using only rank constraint is often not sufficient, but for these methods it is difficult to incorporate additional constraints [53]. Therefore a careful initialisation is needed, otherwise the results can easily drift into a local minima.

Non-linear optimisation is a direct solution for shape and motion recovery when measurement data are missing. By employing non-linear minimisation for cost function, the measurements can be gradually refined and produce jointly optimal 3D structures and camera motion. This problem is known as bundle adjustment and has been studied

for many years [135]. Even though the inherently high number of degrees of freedom may lead to failure of obtaining reliable 3D reconstructions, additional constraints can naturally be included in the cost function.

2.6 Sequential approaches

So far most non-rigid structure from motion methods only refer to batch approach, which implies that all the frames have to be processed at once after the measurement data has been collected. The off-line computations exclude theses methods from being used in many potential real-time applications. Real time tracking and scene estimation using a monocular camera as the only sensor has recently seen great progress [35, 76, 96, 36]. This problem is called real time SfM, or monocular simultaneous localisation and mapping (SLAM). From Davison's seminal work of sparse feature point based SLAM with a single moving camera [35] to live dense reconstruction of a scene [96], real time rigid SfM has already been well-studied and is now being considered in commercial applications.

The gap between batch algorithm and real time processing is that the batch methods used in the NRSfM problem usually are not able to deal with updating the new frame, thus making the on-line processing impossible. To fill in the gap and build the bridge between them, sequential mode can update the model by reformulating the problem in terms of new arriving frames. Morita and Kanade extended the traditional factorisation algorithm into the sequential case, by updating only the first three eigenvectors instead of re-calculation the singular value decomposition for all the data [91]. For the work in [43], the authors added a smoothing penalty on the camera trajectory, updating the structure accordingly as new views were added. Following that idea, the first work of deformable shape and motion recovery in the sequential domain was recently proposed

by Paladini et al. [100], in which they updated the current model by adding the new modes incrementally when the current one cannot model the current frame well enough. In addition, they also presented a 3D implicit low-rank shape model which departs from the classical explicit low-rank shape model. This work is inspired by the Klein and Murray's parallel tracking and mapping system described in [76], where they developed a real time system based on parallel threads - one dealing with robustly tracking erratic hand held motion, the other thread produces a 3D map.

2.7 Summary

This chapter introduced the preliminary knowledge on 3D reconstruction and provided a comprehensive review of existing approaches to 3D shape recovery from monocular sequences. Although a lot of research effort has focused on the development of efficient algorithms for recovery of deformable shapes, the following problems still remain in most existing systems:

The deformable shape reconstruction is rather challenging, mainly because of the inherent basis of ambiguity of the problem. Different structure and motion may be found if the measurements are factorised by enforcing constraints on the camera motion. In the next chapter, we proposed a linear method to solve the ambiguity. The main idea is based on the assumption that the shapes in a sequence can be treated as a set of basis shapes. By directly integrating the constraints to shape bases and their corresponding weighting coefficients, the algorithm avoids the ambiguity in the SVD-based methods, and the bundle adjustment can further optimise the results.

For most current approaches, especially concerning deformable shape recovery, real-time processing is still difficult. In Chapter 4, we proposed sequential approach as a trade-off between batch and real-time 3D reconstruction. Prior knowledge can be

learned and updated online with regards to probability of the shape coefficients.

Apart from the problem mentioned above, the biggest outstanding problem in previously reported research is the fact that 3D shapes may not be accurately modelled in a linear subspace. This is particularly true for articulated objects or an object which contains large and complex deformations. The non-linear manifold learning techniques can be applied in a reconstruction area and will be detailed in the rest of the thesis.

Chapter 3

Shape Recovery with Linear Constraints

3D reconstruction of non-rigid objects without using any prior models may lead to a local solution which correctly minimises the 2D re-projection error but fail to recover the depth information. To overcome this, using prior knowledge of the shape can improve accuracy and stability in the reconstruction process. In this chapter, we depart from the classical low rank shape model discussed in Chapter 2, then introduce the proposed shape model including estimate of the weight probability density function. We show comparative results with existing methods and also present successful reconstructed shapes on both synthetic and motion capture based data.

3.1 Introduction and related work

Structure and motion recovery from image sequences is one of the fundamental problems in computer vision. At the early stage of this research, it usually assumes a static scene or rigid objects, so the results can be gradually refined during the reconstructed process. To extend rigid SfM to the case of recovering 3D deformable objects, Bregler et al. [19] first described a low rank shape model to represent varying shapes. This constructive work not only provide an extension of Tomasi-Kanade's factorisation algorithm under

rigid assumption [129], but also inspired many other methods and models in the field. They factorise the 2D data matrix, using SVD, into object configuration weights, a camera motion matrix and 3D basis shapes used to represent the reconstructed object structure. But the accuracy of these methods strongly depends on the initial affine decomposition, small inaccuracies in the affine values greatly affect the subsequent estimation process. To eliminate the ambiguity, Xiao et al. [150] proposed a closed-form solution to focus on deformable structure from a sequence of images taken with an uncalibrated camera. They employ the traditional orthonormality constraints, but also introduce basis constraints to further determine shape basis, however this method does not cope well with noisy data. To overcome this, iterative optimisation methods [144], based on bundle adjustment [135], were subsequently introduced.

One of the fundamental issues when solving NRSfM problems is that the algorithms may result in meaningless reconstruction because of a high number of degrees of freedom and motion degeneracy. Del Bue demonstrated an alternative approach of bundle adjustment, which introduces object shape prior information [37]. This approach can improve performance for both rigid and non-rigid SfM, obtaining reliable 3D reconstructions when an appropriate initial guess is provided. But in practice, when only constrained by minimisation of the 2D re-projection error and a single basis shape, the optimisation of large number of variables, without a high quality initial guess, often results in convergence to a local minimum.

3.2 Contributions

The main contribution of this chapter is a novel approach for reconstruction of 3D deformable structures, such as articulating face, from 2D video sequences taken by an orthographic camera. We proposed to add specific constraints within the state-of-art

batch-processing scheme previously proposed by Del Bue [37].

Current methodologies apply a non-linear optimisation method to minimise image re-projection error for non-rigid object reconstruction and recovery of camera parameters. Although such methods are proven and widely adopted, their success strongly depends on the quality of the initial estimation. This initialisation oversensitivity can be reduced by the introduction of shape constraints, through integration of the prior information in the cost function. This inspired us to propose a new approach to estimate a shape-varying object using prior learned 3D deformation shape model. The advantage of this approach is that the proposed constraints reduce the likelihood of a non-linear optimisation procedure converging to a local minimum. Furthermore, the final results are not strongly dependent on the initial estimate used in the optimisation process, ensuring the system does not require complex initialisation.

3.3 Deformable Shape Model

As mentioned, the results obtained without using any prior information about shape and/or trajectory are sensitive to the level of noise present in the data and the algorithm initialisation. The greater number of degrees of freedom may lead to smaller re-projection error, but result in unrealistic reconstructed shapes. Appropriate prior shape information can help to augment the accuracy of motion and shape recovery. The key idea in our method is to use a learned shape space model.

3.3.1 PCA

Our method departs somewhat from the linear combination of weighted shape basis model presented in the preceding section. We propose to use standard Principal Component Analysis (PCA) to impose constraints on the basis shapes.

Principal Component Analysis is an effective statistical technique for dimensionality reduction. In the last two decades, it has been employed in a wide range of applications across many areas of computer vision. In this application the idea is to represent each of the shapes in the training dataset in a low dimensional shape space that reduces the large number of observed variables into a small number of principal components. Suppose a training dataset has N shapes and the set of points in i^{th} shape are represented by $\mathbf{X}_i$. The mean shape, $\bar{\mathbf{X}}$, of all the training dataset is given by: $\bar{\mathbf{X}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}_i$ and eigenshapes $\mathbf{E}_i$ and eigenvalues γ_i are obtained from the covariance matrix, defined as $\mathrm{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T$ Any of the shapes from the training dataset can be then approximated by:

$$\mathbf{X}_i \cong \bar{\mathbf{X}} + \boldsymbol{\gamma}_i\mathbf{E} = \bar{\mathbf{X}} + \begin{bmatrix}\gamma_{i1} & \dots & \gamma_{iK}\end{bmatrix}\begin{bmatrix}\mathbf{E}_1 \\ \vdots \\ \mathbf{E}_K\end{bmatrix} \tag{3.1}$$

K is the number of dimensions after reducing the dimensionality, γ_i describes the contribution of i^{th} eigenshape and is calculated using the inner product between $\mathbf{E}_i$ and $\mathbf{X}_i - \bar{\mathbf{X}}$. Every input data $\mathbf{X}_i$ projects into a point in the $K-1$ dimensional subspace, spanned by the selected eigenvectors [93].

Figure 3.1 shows a working example of PCA, where the left image gives a Gaussian distribution together with two principal components; the right image is a projection on the eigenvector and its corresponding largest eigenvalue. The transformation preserves most geometric information of the data.

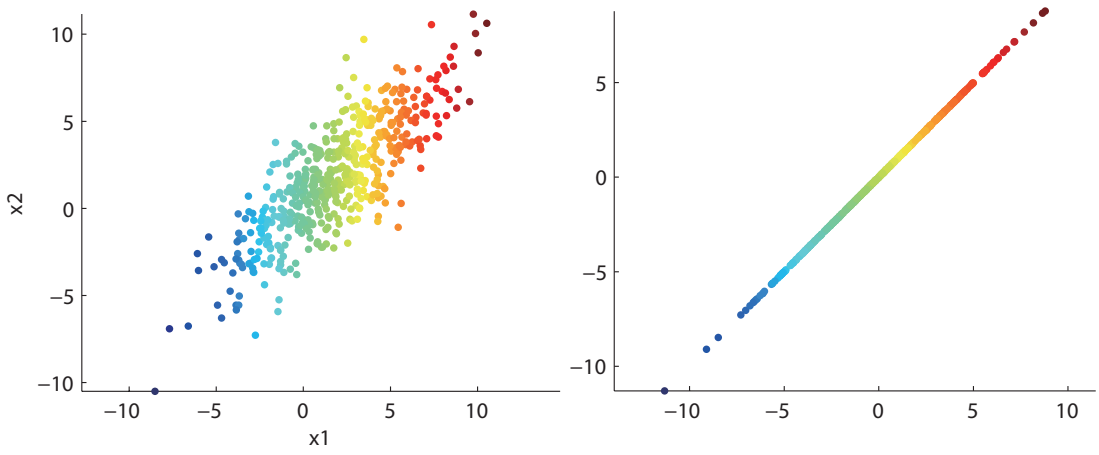


Figure 3.1: Left: Gaussian distribution; Right: Projection on the eigenvector corresponding to the largest eigenvalue.

3.3.2 Proposed shape model

Inspired by the idea of PCA and following the deformable shape representation described in Equation 2.16, our proposed shape model is given by:

$$\mathbf{S}_t = \mu\mathbf{B}_0 + \begin{bmatrix} \alpha_{t1} & \cdots & \alpha_{tK} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} \tag{3.2}$$

There are $K+1$ basis shapes, $\mathbf{B}_0$ as the first basis shape is similar to mean shape $\bar{\mathbf{X}}$ computed from all the faces in the training datasets. Therefore μ is a scaling factor for first basis shape which controls the overall size of the shape. The rest of the basis shapes, $\mathbf{B}_1$ to $\mathbf{B}_K$ are forced to be close to the corresponding eigenshapes. The basis shapes are only "encouraged" to be close to the mean shape and eigenshapes, instead of being forced to exactly match.

By stacking the shapes $\mathbf{S}_t$ for each time instant, then projecting them onto the 2D images using an orthographic projection model, equation 3.2 can be re-written in

compact matrix form:

$$\begin{aligned}\mathbf{W} &= \begin{bmatrix} \mu_1\mathbf{R}_1\mathbf{B}_0 \\ \vdots \\ \mu_F\mathbf{R}_F\mathbf{B}_0 \end{bmatrix} + \begin{bmatrix} \alpha_{11}\mathbf{R}_1 & \cdots & \alpha_{1K}\mathbf{R}_1 \\ \vdots & \ddots & \vdots \\ \alpha_{F1}\mathbf{R}_F & \cdots & \alpha_{FK}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} \\ &= \begin{bmatrix} \mu_1\mathbf{R}_1 & \alpha_{11}\mathbf{R}_1 & \cdots & \alpha_{1K}\mathbf{R}_1 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_F\mathbf{R}_F\alpha_{F1}\mathbf{R}_F & \cdots & \alpha_{FK}\mathbf{R}_F \end{bmatrix} \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} = \begin{bmatrix} -\mathbf{M}_1- \\ \vdots \\ -\mathbf{M}_F- \end{bmatrix} \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_K \end{bmatrix} = \mathbf{MB}\end{aligned} \tag{3.3}$$

3.4 Prior probability on shape coefficients

Given that deformation is not random, with prior knowledge it is possible to restrict the estimated deformation of the object; assuming it is known how the weighting coefficients α_{td} are distributed in K dimensional space. If the prior is not applied to constrain the weights, it may lead to the reconstructed shapes representing infeasible deformations.

To further constrain the reconstructed shapes, a prior probability on the values of the weighting coefficients is added to the model. The Parzen window density estimation [102] in the face-eigenspace was used for this purpose.

$$p(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h^2}\phi\left(\frac{\gamma_i - \alpha}{h}\right) \tag{3.4}$$

where N is the number of shapes used to estimate the probability density function, and $\phi(\centerdot)$ is a kernel function. For the isotropic Gaussian kernel function the estimate of the density function is given by:

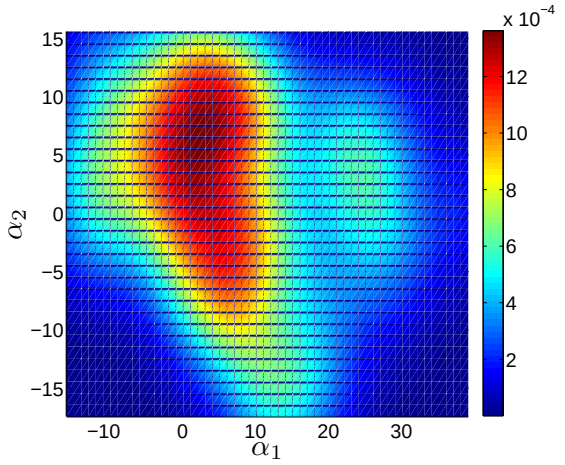


Figure 3.2: Probability distribution of configurations for first two basis shapes in 2D.

$$p(\alpha) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{\|\gamma_i - \alpha\|^2}{2\sigma^2}\right) \tag{3.5}$$

The dimensionality of this function is defined by the number of eigenshapes used in the approximation. As an example, shape coefficients probability distribution for 2D shape space is shown in Figure 3.2.

3.5 Non-linear refinement

As the information about shapes and weights probability distribution is learned in advance, the optimisation process comes down to minimising a cost function built as a superposition of four components.

The first component of the cost function measures the re-projection error between the feature points detected in the observed images and corresponding projection of 3D points in the estimated shapes. The re-projection error is given by:

$$\varepsilon_{re} = \sum_{t=1,p=1}^{F,P} \|\mathbf{w}_{tp} - \tilde{\mathbf{w}}_{tp}\|^2 \text{ with } \tilde{\mathbf{w}}_{tp} = \mathbf{R}_t \sum_{d=0}^{K} \alpha_{td}\mathbf{B}_{dp} \tag{3.6}$$

Assuming that the reconstructed object is viewed by an orthographic camera, rotation matrix $\mathbf{R}_t$ represents an orthographic camera matrix. The second component of the cost function enforces orthonormality of all $\mathbf{R}_t$ and is expressed as:

$$\varepsilon_{rot} = \sum_{t=1}^{F} \left\| \mathbf{R}_t {\mathbf{R}_t}^T - \mathbf{I} \right\|^2 \tag{3.7}$$

The prior on the shape basis given in Equation 3.8 is included as the third component of the cost function:

$$\varepsilon_{bs} = \left\| \mathbf{B}_0 - \bar{\mathbf{X}} \right\|^2 + \sum_{d=1}^{K} \left\| \mathbf{B}_d - \mathbf{E}_d \right\|^2 \tag{3.8}$$

Given that the reconstructed object is not part of the training dataset, we are much more concerned about recovering the 3D shapes, rather than having accurate basis shapes.

Last but not least, following from the discussion in Section 3.4, the fourth component of the cost function introduces constraints on the weighting coefficients. We restrict the search for optimal weights within the high probability region of the learned weights probability distribution by maximising $p(\alpha_t)$.

The overall proposed cost function combines minimisation of the re-projection error with efficient constraints for rotation matrices, shape basis, as well as weighting coefficients:

$$\min_{\mathrm{R}_t, \mathrm{B}_d, \alpha_t} \left(\varepsilon_{re} \left(\mathbf{R}_t, \mathbf{B}_d, \alpha_t \right) + \varphi_1 \varepsilon_{rot} \left(\mathbf{R}_t \right) + \varphi_2 \varepsilon_{bs} \left(\mathbf{B}_d \right) - \varphi_3 p(\alpha_t) \right) \tag{3.9}$$

where scalars φ_1, φ_2, φ_3 are the designed parameters controlling the importance of each constraint in the cost function. φ_2 is the importance factor for the constraint set on basis shapes. Consider that the reconstructed shapes are different from the training shapes, thus $\mathbf{B}_d$ is only forced to be close to the basis shape. $\mathbf{B}_d$ would be too similar

to the basis shapes if φ_2 is set too high. φ_3 is the parameter for the constraint on the weighting coefficient. Without the constraint, the weighting coefficients may go anywhere in the shape space which may lead to meaningless reconstructed shapes. The scalars φ_1, φ_2, φ_3 are selected experimentally as 1, 0.1 and 1, respectively in our case. The selection was based on a systematic search of the parameter space. A non-linear optimisation based on bundle adjustment using Levenberg-Marquardt algorithm was applied to minimise this compound cost function.

3.6 Initialisation

In the proposed method, rather than using the method described in Section 2.3.2 to initialise the data, the method proposed in [37] is implemented. The method uses the generalised SVD (GSVD) [16] followed by orthonormal decomposition [39]. The method formulates the problem as two bilinear models $\mathbf{W} = \mathbf{MB} = [\mathbf{M}_1|\mathbf{M}_2]\,[\mathbf{B}_1|\mathbf{B}_2]^T$, where the factors with subscript "1" subscript "2" are derived from single prior shape and image measurements, and subscript "2" refers to the remaining prior shapes. Thus an initial shape is given by a rigid shape which is computed from measured data and prior shape model. However, in our model, the mean shape and the eigenshapes have been trained in advance, thus we have more than one prior shape that means we can calculate the initial affine motion directly, as $\mathbf{M}_0 = \mathbf{WB}^\dagger$, where $\mathbf{B}$ is approximated as mean shape and eigenshapes and $\mathbf{B}^\dagger$ is the pseudo inverse of $\mathbf{B}$. Then to initialise rotation and weights, orthonormal decomposition [18] is applied to decompose the initial motion matrix $\mathbf{M}_0$.

Brand[18] proposed a method to factorise motion matrix for each frame using orthonormal decomposition into a rotation matrix and a shape coefficient vector. $\mathbf{M}_0$

can be written as $\mathbf{M}_0 = \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_F \end{bmatrix}$. Each motion matrix $\mathbf{M}_t$, where $t = 1 \dots F$, is a 2 row sub-block, see equation 3.3, which can be rearranged as,

$$\mathbf{M}_t \rightarrow \hat{\mathbf{M}}_t = \begin{bmatrix} \alpha_{t1}\mathbf{r}_t & \cdots & \alpha_{tK}\mathbf{r}_t \end{bmatrix} \tag{3.10}$$

where $\mathbf{r}_t = \begin{bmatrix} r_{t1} & \cdots & r_{t6} \end{bmatrix}^T$. Then the motion matrix is post-multiplied by a $K \times 1$ vector $\mathbf{c} = [1 \cdots 1]$,

$$\mathbf{a}_t = k\mathbf{r}_t = \hat{\mathbf{M}}_t\mathbf{c}, \text{ with } k = \alpha_{t1} + \ldots + \alpha_{tK} \tag{3.11}$$

The column vector $\mathbf{a}_t$ can be rearranged as a 2×3 matrix, as $\mathbf{a}_t \rightarrow \mathbf{A}_t = \begin{bmatrix} kr_{t1} & kr_{t2} & kr_{t3} \\ kr_{t4} & kr_{t5} & kr_{t6} \end{bmatrix}$. Consider rotation $\mathbf{R}_t$ is an orthonormal matrix, thus $\mathbf{A}_t\mathbf{R}_t^T = \sqrt{\mathbf{A}_t\mathbf{A}_t^T}$. The rotation can be computed as $\mathbf{R}_t^T = \sqrt{\mathbf{A}_t\mathbf{A}_t^T}/\mathbf{A}_t$.

Once rotation has been estimated, it is possible to get weighting coefficients from the rotation. Rearrange Equation 3.10 as $\hat{\mathbf{M}}_t \rightarrow \tilde{\mathbf{M}}_t = \begin{bmatrix} \alpha_{t1}\mathbf{r}_t^T & \cdots & \alpha_{tK}\mathbf{r}_t^T \end{bmatrix}$. The coefficients for each frame t can be derived as,

$$\tilde{\mathbf{M}}_t\mathbf{r}_t = \begin{bmatrix} \alpha_{t1}^T\mathbf{r}_t\mathbf{r}_t^T & \cdots & \alpha_{tK}^T\mathbf{r}_t\mathbf{r}_t^T \end{bmatrix} = 2\begin{bmatrix} \alpha_{t1}^T & \cdots & \alpha_{tK}^T \end{bmatrix}^T \tag{3.12}$$

3.7 Missing data

The two algorithms proposed above assume that the measurement matrix $\mathbf{W}$ is complete, with all the feature points detected in all the images. This is unlikely to happen in practice as some of the feature points will not be detected in all the images. This

could be because of the feature point detection problems or because some parts of the 3D object may not be visible from all the camera positions. This means some of the entries in the measurement matrix $\mathbf{W}$ may be unknown. This section describes a simple but efficient method for the solving missing data problem, by recovering the missing entries in measurement $\mathbf{W}$ before reconstruction of 3D shapes and camera motion.

If the input data is incomplete, instead of using more complex and time-consuming optimisation process to estimate the missing values [52, 41], we predict the 2D coordinates of these points only based on the current measurement and learned eigenshapes. Assuming the total P feature points are to be reconstructed, we can write $\mathbb{I} = \bar{\Pi}_t + \bar{\Pi}_t^*$, where $\mathbb{I}$ is an identity matrix and $\bar{\Pi}_t$ is a $P \times P$ diagonal matrix:

$$\bar{\Pi}_t(p,p) = \begin{cases} 0, \text{ if the point } p \text{ is missing in } t \text{ image} \\ 1, \text{ if the point } p \text{ presents in } t \text{ image} \end{cases} \tag{3.13}$$

According to Equation 2.17, measurement matrix can be factorised into motion $\mathbf{M}$ and shape basis $\mathbf{B}$ matrices. The incomplete measurements, which contain only detected points in t frame, can be represented as:

$$\hat{\mathbf{w}}_t = \mathbf{w}_t \Pi_t \tag{3.14}$$

and the missing measurements as:

$$\hat{\mathbf{w}}_t^* = \mathbf{w}_t \Pi_t^* \tag{3.15}$$

where matrix Π_t and Π_t^* are obtained from $\bar{\Pi}_t$ and $\bar{\Pi}_t^*$ by removing all columns for which entries are all zeros, $\mathbf{w}_t$ represents row $2t-1$ and $2t$ of the matrix $\mathbf{W}$. Substituting Equation 3.14 into Equation 2.17, the incomplete measurement can be written as:

$\hat{\mathbf{w}}_t = \mathbf{M}_t\mathbf{B}\Pi_t$.

We first compute the motion matrix $\mathbf{M}_t$ in terms of the visible points and its corresponding eigenshapes in t frame, $\mathbf{M}_t = \hat{\mathbf{w}}_t(\mathbf{E}\Pi_t)^\dagger$, where $(\mathbf{E}\Pi_t)^\dagger$ represents Moore-Penrose pseudoinverse of $\mathbf{E}\Pi_t$, with eigenshapes $\mathbf{E}$ used as basic shapes. Once the motion $\mathbf{M}_t$ is obtained, the missing values can be calculated as $\hat{\mathbf{w}}_t^* = \mathbf{M}_t\mathbf{E}\Pi_t^*$. Thus the completed measurement matrix is:

$$\mathbf{w}_t = \hat{\mathbf{w}}_t\Pi_t^T + \hat{\mathbf{w}}_t^*\Pi_t^{*T} \tag{3.16}$$

In the case of batch processing, the eigenshapes $\mathbf{E}$ are learned during off-line training and the whole measurement $\mathbf{W}$ is calculated before doing further reconstructions. In sequential mode, missing values in each frame have to be estimated when the new frame arrives. Note that the eigenshapes have been updated using incremental PCA, the eigenshapes used for calculating the missing values should consist of off-line eigenvectors $\mathbf{E}$ and online learned eigenvectors $\mathbf{U}$.

3.8 Experiments

The experiments to evaluate the proposed methodology were based on batch formulation of an articulating face and human motion. In the case of reconstruction of objects undergoing only small deformations, the estimated shape can be accurately represented using a model with a relatively small number of degrees of freedom, thereby allowing for linear deformations. We firstly introduce the training data and show the learned shape model. Then, to demonstrate the performance of the proposed methods, extensive experimental evaluation has been provided. We show qualitative and quantitative results on different datasets, and compare the proposed method with previous approaches. We have applied our approaches to the Hi4D-ADSIP [85] database, including video clips of

seven different facial expressions (*anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise* and *pain*) at three intensity levels (mild, middle and extreme) and videos showing people reading predefined phrases ("*talking* subjects"). Ground truth data of an articulating face was captured using Passive 3D scanner with 3D tracking of 83 feature points.The points were projected onto the image sequences under the orthographic camera model. The models and algorithms used for comparison are as follows:

SP: Factorisation with shape priors [37].

MP: The metric projection method [101].

BPCA: The proposed batch approach

3.8.1 Shape model

The off-line training datasets are taken from the BU-3DFE database [152]. A total number of 2400 with 83 feature points, rigidly co-registered using standard Procrustes Analysis [54], 3D face images of different subjects exhibiting different facial expressions were used for learning the shape model and the distribution of weights. The feature points tracked in the testing data have to be the same points extracted on the surface of the model from the training datasets. Consider that real measurements are noisy, to test the method in a use which reflects real reconstruction, the noisy measurement has been considered in later experiments in Section 3.8.2. In the case of real applications, the correspondence between points in different images must be found in advance. This has been discussed in Chapter 9. Figure 3.3 shows an example of the shapes built using the learned mean face with eigenfaces.

3.8.2 Evaluation

The performance of our proposed shape model with prior information based on batch type operation was evaluated in a number of experiments.

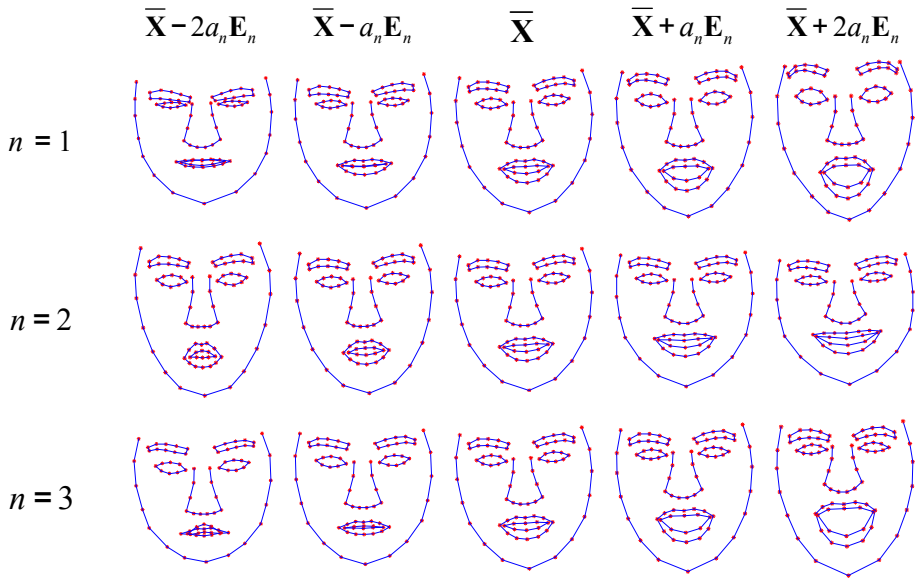


Figure 3.3: Learned shapes variability: the superposition of the mean shape with first three eigenshape with different weights coefficients.

The accuracy of 3D shape reconstruction is affected by the number of basis shapes. For the first set of experiments, we start with testing on four facial sequences, three for extreme level of facial expression (*happiness*, *sadness* and *surprise*) data and one for "*talking* subjects". The results are listed in Table 3.1, evaluated in terms of 3D reconstructed shape error and 2D re-projection error, for cases where differing number of basis shapes are used for reconstruction. The 3D error is measured by normalised mean error over all frames and all points:

$$e = \frac{1}{\Delta FP}\sum_{t=1}^{F}\sum_{p=1}^{P} e_{tp}, \qquad \Delta = \frac{1}{3F}\sum_{t=1}^{F}(\Delta_{tx}+\Delta_{ty}+\Delta_{tz}) \tag{3.17}$$

where Δ_{tx},Δ_{ty},Δ_{tz} are the standard deviations of x,y and z coordinates of ground truth shape at t^{th} frame and e_{tp} is the Euclidean distance between corresponding point p at frame t in the reconstructed and ground truth shapes.The 2D re-projection error is

#Basic shape		3	5	7	10	15
3D error	Happiness	0.1410	0.0776	0.0696	0.0704	0.0703
	Sadness	0.1411	0.0996	0.0916	0.0889	0.0898
	Surprise	0.1591	0.1193	0.1152	0.1172	0.1169
	Talking	0.1582	0.0856	0.0790	0.0674	0.0670
2D error	Happiness	0.0156	0.0081	0.0059	0.0054	0.0053
	Sadness	0.0169	0.0101	0.0068	0.0064	0.0060
	Surprise	0.0195	0.0082	0.0066	0.0061	0.0059
	Talking	0.0267	0.0121	0.0084	0.0076	0.0065

Table 3.1: The influence of the number of basis shapes. Reconstruction error with respect to the number of basis shapes for the selected facial expression sequences.

calculated using

$$\sum_{t=1}^{F}\sum_{p=1}^{P}\left(\mathbf{w}_{tp}-\mathbf{w}_{tp}^{'}\right)/\sigma FP \tag{3.18}$$

where σ is the standard deviation of the measurement data and $\mathbf{w}_{tp}^{'}$ represents re-projection 2D points getting from the projection of reconstructed shapes using recovered camera motion. As expected, more accurate results were obtained when increasing the number of basis shapes due to the greater number of trained eigenfaces used to constrain the reconstructed shapes. Without noise, the recovered shape is very similar to the true shape with the reconstruction error close to zero. With noise present in the measurements, reasonably accurate shapes are still obtainable, showing that the method is robust.

The results shown in Figure 3.4 are for tracking 83 points over a 259 frame sequence of anger. We present both front and side views of a selection of facial reconstructions extracted from the sequences.

In real image sequences, feature points often disappear and reappear from the image as the object deforms and camera moves. As a result, the measurement matrix is incomplete due to occlusions. To test the performance of the proposed methods in that case, we follow the evaluation procedure originally proposed in [131] to simulate the

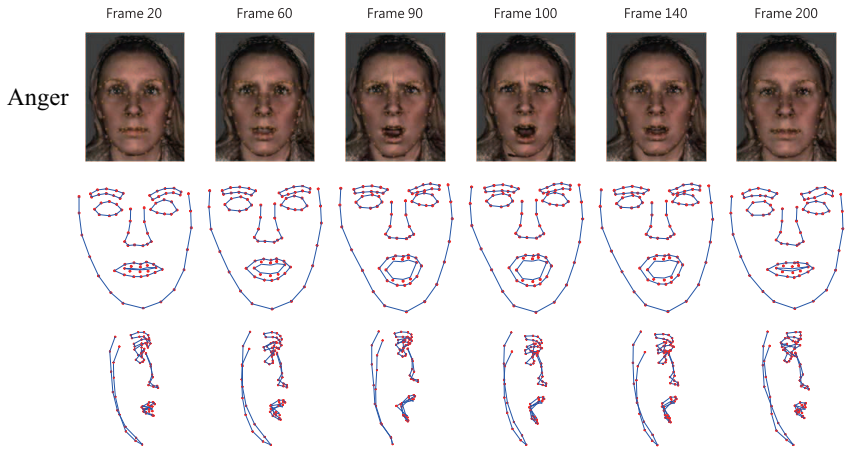
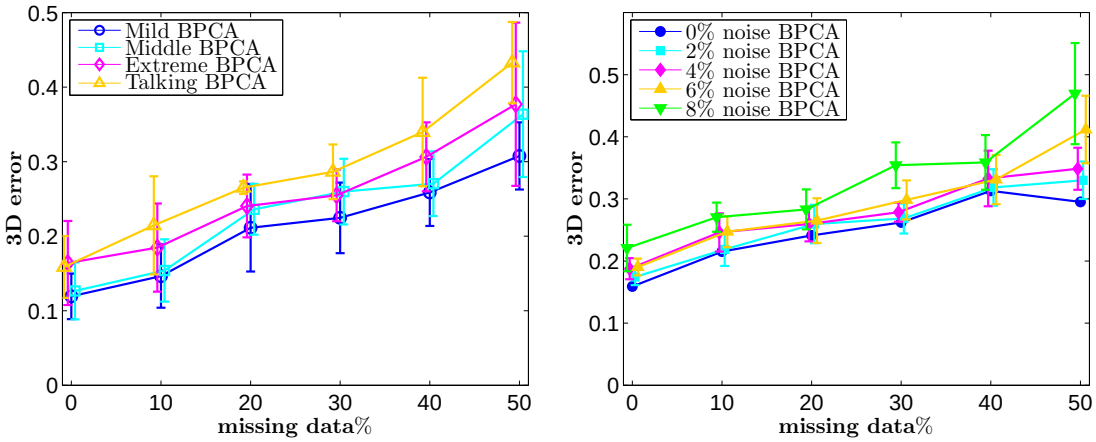


Figure 3.4: Results for anger facial expression sequences. First row: Input images tracked with feature points. Second and third row: Front and side views of the 3D reconstruction using the proposed method.

missing data by discarding 2D entries uniformly at random with 10%, 20%, 30%, 40% and 50% probability. To simplify results visualisation, all the sequences are separated into four groups: three for different intensities of facial expression and one for "*talking* subjects", with 10 sequences taken from different subjects per group. 3D reconstruction errors for BPCA and their corresponding standard deviation calculated for each group are shown in Figure 3.5(a).

In most cases, measurement noise usually appears when inaccurate tracking takes place, affecting the 2D observation data. The aim of the following experiment is to evaluate the performance with noise in measurement and different ratios of missing data. Gaussian noise with noise levels up to 8% was applied for extreme surprise facial expression sequence where the missing points were selected randomly with levels between 0% and 50%. The measurement $\mathbf{W}$ was perturbed by Gaussian noise according to the standard deviation of the measurement data with given level of noise. The experiments for each level of noise and each level of occlusion were repeated 10 times. Figure 3.5(b) shows the results of the proposed method.



(a) (b)

Figure 3.5: (a) Reconstruction results for all of the facial expression data with occlusion. The figure shows the dependence on increasing amount of missing data. (b) Reconstruction results for varying levels of missing data and 5 level of noise for extreme surprise facial expression sequence.

3.8.3 Comparison with previously proposed methods

For the comparative evaluation, performance of the proposed method is tested against two previously proposed approaches, namely: factorisation with shape prior (SP) [37]; and the metric projection method (MP) [101]. The experiments in this case were performed for all 30 facial expression sequences and 10 talking sequences. To better visualise the results the data was divided into the same groups (mild, middle, extreme and talking) as in Section 3.8.2. The average 3D error, maximum error and standard deviation of each group were calculated. Table 3.2 summarises results of these tests and indicates that the proposed BPCA produces better performance than the previous methods. Extreme level of facial expression and Talking sequences usually contain larger deformation than Mild and Middle level of expressions, which leads to higher reconstruction error using the proposed method.

Method		Mild	Middle	Extreme	Talking
SP	3D error	0.1741	0.2249	0.2867	0.2629
	Max error	0.2755	0.2878	0.3579	0.2930
	Std.dev.	0.0491	0.0376	0.0475	0.0422
MP	3D error	0.1063	0.1431	0.2467	0.1886
	Max error	0.1646	0.1916	0.4355	0.2238
	Std.dev.	0.0316	0.0342	0.1009	0.0399
BPCA	3D error	0.1193	0.1266	0.1641	0.1588
	Max. error	0.1862	0.1941	0.2956	0.2237
	Std. dev.	0.0306	0.0382	0.0564	0.0415

Table 3.2: Average 3D reconstruction error / Max 3D error / standard deviation for different approaches

3.9 Summary

We have developed several extensions for the recently proposed algorithm for recovering 3D deformable object and camera pose from a video sequence. The proposed extensions include use of learned shape model and distribution of the weights, in the cost function which improves performance of the optimisation process.

Although the method works well, the implicit assumption that the 2D points have to be the projections of the same 3D points on the surface is a limitation of the method. Furthermore the reconstruction can only be done after all the measurement data has been collected, which is obviously not suitable for any real-time applications. The recent progress on the algorithm of real-time 3D reconstruction system for deformable objects will be shown in the coming chapter.

Chapter 4

Incremental Approach with Online Learned Shape Prior

Most existing approaches to the non-rigid structure from motion problem use batch type algorithms, with all the data collected before 3D shape reconstruction takes place. Such a methodology is not suitable for real-time applications. Concurrent on-line estimation of the camera position and 3D structure, based only on the measurements up to that moment, is a much more challenging problem. In this chapter, a novel approach is proposed for recursive recovery of non-rigid structures from image sequences. The proposed, adaptively learned constraints have two aspects, consisting of constraints imposed on the basis shapes, the basis building blocks from which shapes are reconstructed, as well as constraints imposed on the mixing coefficients in a form of their probability distribution. The constraints are updated when the current model inadequately represents new shapes. This is achieved by means of Incremental Principal Component Analysis (IPCA). Results of the proposed method are shown on synthetic and real data of articulating face.

4.1 Introduction

Although tremendous progress has been made on SfM for both rigid and non-rigid shapes, the main limitation of most extant works is that they only refer to off-line (batch method) computations. The downside of batch methods is that the reconstruction can only start once all measurement data has been collected. To extend batch mode to the case of online (recursive) operation, Morita and Kanade [91] first presented a sequential factorisation method, by considering the feature positions as a vector time series and updating only the first three eigenvectors instead of computation of singular value decomposition. Subsequent research for sequential shape and motion recovery has been developed by Mouragnon et al. [92], who demonstrated a generic and incremental method by minimising an angular error between rays. Similarly, for the work in [43] the authors added a smoothing penalty on the camera trajectory, updating the structure accordingly as new views are added. Solutions to execute SfM in real-time can be classified as filter based framework [123, 40] or keyframe-based [77] optimization and have proven to be successful. These methods give motivation for real-time implementations, which nevertheless, have so far only dealt with rigid objects or static environment. As yet a limited number of works have been published covering online deformable structure recovery. Most recently, Paladini et al. [100] have made progress in this. They divided the NRSfM problem into two processes: model based tracking and model updating. Their work proposed a rank-growing system which updates the current shape model when the 2D re-projection error exceeds an expected value. This technique makes online NRSfM more tractable but whilst the higher number of degrees of freedom may lead to smaller re-projection error, this can result in unrealistic reconstructed shapes, unrepresentative of the true object. The method does not address the self-occlusion problem either, where measurements are

assumed to be complete, which is rarely valid.

4.2 Contributions related to previous work

The methodology proposed in this chapter is based on our previous work presented in Chapter 3, which utilises appropriate prior shape information. The key idea in this existing method was the use of a learned shape space model. The method departed from the linear combination of a set of shape bases presented in the preceding section, with standard PCA to obtain constraints on the basis shapes.

The idea was to represent each of the shapes in the training dataset in a low dimensional shape space that reduces the large number of observed variables into a small number of principal components. The overall shape model was similar to the one given in Equation 2.16, but with additional constraints imposed both on the basis shapes $\mathbf{B}_i$ and the deformation coefficients α_i. The $\mathbf{B}_0$ was constrained by the mean shape $\bar{\mathbf{X}}$ computed over all faces in the training datasets, with α_0 controlling the overall size of the shape. The rest of the basis shapes, $\mathbf{B}_1 \ldots \mathbf{B}_K$ are forced to be close to the corresponding eigen-shapes. Given that deformations are not random, additional constraint was applied to the deformation coefficients α_i through imposition of a prior probability on their distribution.

4.3 Recursive algorithm

Figure 4.1 is a flowchart of the proposed recursive method. Generally, the proposed algorithm contains two modules: the reconstruction module and the model update module.

For the recursive shape recovery, the shape is divided into off-line and online components: $\mathbf{S}_t = \mathbf{S}_t^{\text{off}} + \mathbf{S}_t^{\text{on}}$.The off-line part is mainly used to indicate the static overall shape

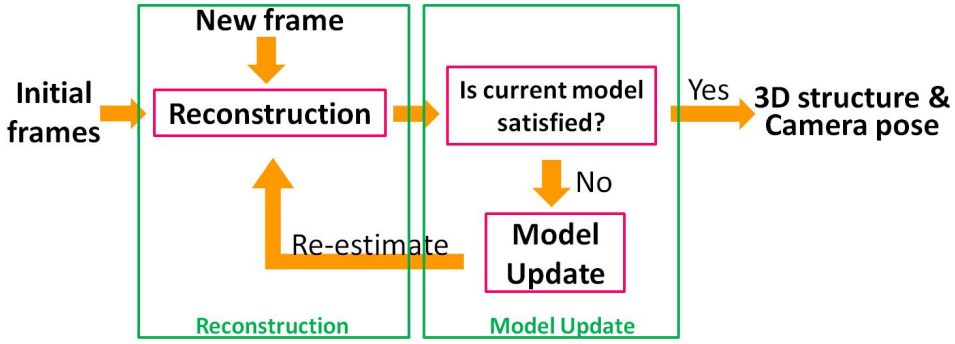


Figure 4.1: Flowchart for the proposed recursive method

and the online part is responsible for representing the dynamic shape changes. The method described in the preceding section was used to estimate the off-line shape $\mathbf{S}_t^{\text{off}}$ with the prior information about shapes and weights probability distribution learned in advance using standard PCA technique on a training database of co-registered shapes. The online (dynamic) shape $\mathbf{S}_t^{\text{on}}$ is modelled in a similar way as the off-line shape:

$$\mathbf{S}_t^{\text{on}} = \begin{bmatrix} \beta_{t1} & \cdots & \beta_{tM} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{B}}_1 \\ \vdots \\ \tilde{\mathbf{B}}_M \end{bmatrix} \tag{4.1}$$

As for the off-line model, the online shapes are represented by a linear combination of basis shapes $\tilde{\mathbf{B}}_i$ weighted by the shape coefficients β_i. The main difference between $\mathbf{S}_t^{\text{off}}$ and $\mathbf{S}_t^{\text{on}}$ is the way in which the shape and coefficient constraints are calculated. Whereas constraints for the off-line shapes are calculated using standard PCA, the constraints for the online shapes are learned recursively using the incremental PCA (IPCA) method.

4.3.1 Incremental PCA

Although standard PCA is allowed to optimise reconstruction of the training data by projecting the input data onto its principal axes, it is not suitable for online learning. It requires all the data in advance. Once each new sample arrives, the PCA is performed for all available data up to now. The idea of incremental PCA computation was introduced to overcome the drawback of batch method [55, 56, 27]. These algorithms have been developed and used in different areas in computer vision, such as learning and recognition [7, 155, 147]. The main advantage for incremental computation of PCA is that it enables estimation of the shape space based on partial observations. As an additional benefit, the original data can be removed once the eigenspace is updated, therefore reducing the data storage requirements.

The incremental approach requires updating the current model by taking into account a new input shape. Say when a new shape $\mathbf{S}_t$ arrives, assuming the mean shape $\bar{\mathbf{S}}_{t-1}$, a set of eigenvectors $\mathbf{U}_{t-1}$ and corresponding eigenvalues are obtained from already observed training dataset. The Algorithm 1 summarises the IPCA algorithm indicating how those inputs are updated.

Algorithm 1 Incremental PCA

Input: new shape $\mathbf{S}_t$, current eigenvectors $\mathbf{U}_{t-1}$, current projected vectors $\mathbf{A}_{t-1}$, current mean shape $\bar{\mathbf{S}}_{t-1}$

Output: updated eigenvectors $\mathbf{U}_t$, updated mean shape $\bar{\mathbf{S}}_t$, updated projected vectors $\mathbf{A}_t$.

1: Compute the projection of $\mathbf{S}_t$ on the shape space $\mathbf{a} = \mathbf{U}_{t-1}^T \left(\mathbf{S}_t - \bar{\mathbf{S}}_{t-1}\right)$
2: Get the orthogonal residual vector $\mathbf{r}_t = \mathbf{S}_t - \left(\mathbf{U}_{t-1}\mathbf{a} + \bar{\mathbf{S}}_{t-1}\right)$
3: Compute append eigenvector $\mathbf{U}' = \begin{bmatrix}\mathbf{U}_{t-1} & \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|}\end{bmatrix}$ and append projected vector $\mathbf{A}' = \begin{bmatrix}\mathbf{A}_{t-1} & \mathbf{a} \\ 0 & \|\mathbf{r}_t\|\end{bmatrix}$
4: Standard PCA on $\mathbf{A}'$ to get its mean shape $\bar{\mathbf{S}}''$ and eigenvectors $\mathbf{U}''$
5: Update projected vector $\mathbf{A}_t = \mathbf{U}' \left(\mathbf{A}' - \bar{\mathbf{S}}''\mathbf{1}_{1\times t+1}\right)$
6: Update eigenvectors $\mathbf{U}_t = \mathbf{U}'\mathbf{U}''$
7: Update mean shape $\bar{\mathbf{S}}_t = \bar{\mathbf{S}}_{t-1} + \mathbf{U}'\bar{\mathbf{S}}''$

4.3.2 On-line novelty detection

Neto and Nehmzow employed the traditional IPCA algorithm to perform on-line novelty detection [95]. They use the magnitude of the residual vector to check if the current model needs to be updated or not. The algorithm of IPCA with online novelty detection is summarised in Algorithm 2. As shown in the algorithm, the model does not need to be updated when the Root-Mean-Square (RMS) error between original data and the reconstruction of its projection onto the current eigenspace is smaller than the threshold, which implies that the current model is still able to describe the new data. The threshold is selected experimentally. Algorithm 2 is very similar to Algorithm 1, but with one more step for novelty detection.

Algorithm 2 On-line novelty detection

Input: new shape $\mathbf{S}_t$, current eigenvectors $\mathbf{U}_{t-1}$, current projected vectors $\mathbf{A}_{t-1}$, current mean shape $\bar{\mathbf{S}}_{t-1}$

Output: updated eigenvectors $\mathbf{U}_t$, updated mean shape $\bar{\mathbf{S}}_t$, updated projected vectors $\mathbf{A}_t$.

1: Compute the projection of $\mathbf{S}_t$ on the shape space $\mathbf{a} = \mathbf{U}_{t-1}^T \left(\mathbf{S}_t - \bar{\mathbf{S}}_{t-1}\right)$
2: Get the orthogonal residual vector $\mathbf{r}_t = \mathbf{S}_t - \left(\mathbf{U}_{t-1}\mathbf{a} + \bar{\mathbf{S}}_{t-1}\right)$
3: **if** $\|\mathbf{r}_t\| > r_T$ **then**
4: Compute append eigenvector $\mathbf{U}' = \left[\mathbf{U}_{t-1} \quad \frac{\mathbf{r}_t}{\|\mathbf{r}_t\|}\right]$ and append projected vector $\mathbf{A}' = \begin{bmatrix} \mathbf{A}_{t-1} & \mathbf{a} \\ 0 & \|\mathbf{r}_t\| \end{bmatrix}$
5: Standard PCA on $\mathbf{A}'$ to get its mean shape $\bar{\mathbf{S}}''$ and eigenvectors $\mathbf{U}''$
6: Update projected vector $\mathbf{A}_t = \mathbf{U}' \left(\mathbf{A}' - \bar{\mathbf{S}}''\mathbf{1}_{1\times t+1}\right)$
7: Update eigenvectors $\mathbf{U}_t = \mathbf{U}'\mathbf{U}''$
8: Update mean shape $\bar{\mathbf{S}}_t = \bar{\mathbf{S}}_{t-1} + \mathbf{U}'\bar{\mathbf{S}}''$
9: **else**
10: $\bar{\mathbf{S}}_t = \bar{\mathbf{S}}_{t-1}, \mathbf{U}_t = \mathbf{U}_{t-1}, \mathbf{A}_t = \mathbf{A}_{t-1}$
11: **end if**

4.3.3 A recursive approach to 3D reconstruction

A summary of the algorithm for recursive 3D reconstruction is given in Algorithm 3. Initial shapes are estimated from the first N frames (20 frames in the case of experiments described in section 3.8), in which the affine solution is estimated by using the initialisation described in the section 3.6. The initial shapes are obtained via a nonlinear optimisation with shape constraints, through integration of the prior information in the cost function following the method described in Section 3.5. For each new frame a local bundle adjustment is used over all frames in a sliding window of length l to optimise parameters for shape coefficients and basis shapes in order to reconstruct the current shape. Our approach for model updating is inspired by the work of Neto and Nehmzow [95]. They perform online novelty detection by comparing the magnitude of a residual vector which defines the error between reconstruction of a projection and its original data with a predefined threshold r_T (threshold r_T is set to 2 in our experiment). The model is updated only if the value exceeds the threshold or the magnitude of the two residual vectors between two reconstructed shapes is relatively large, which implies that the current model is unlikely to be able to recover the deformation in the subsequently arriving frame. Unlike the method presented in [100] where new basis shapes are added when the current model is unable to describe the shape, and considering that increasing the number of basis shape may lead to overfitting problem, the basis shapes in the proposed method are updated but the same number of basis are kept to avoid overfitting.

When all data has been updated in this stage, a re-estimation for the current frame ensures the model better fits the observation.

The optimisation during recursive computation is based on local bundle adjustment incorporating the proposed additional constraints. The online basis shapes $\tilde{\mathbf{B}}$ are

Algorithm 3 Outline of Recursive algorithm

Input: Stream of 2D correspondence points.

Output: 3D deformable shapes $\mathbf{S}_t$ and camera motion $\mathbf{R}_t$ for each frame.

1: Build matrix $\mathbf{W}_{t|t=N}$ where $\mathbf{W}$ is a $2N \times P$ matrix.
2: Using method described in Section 3.4 estimate:$\{\mathbf{R}_1, \cdots, \mathbf{R}_N\}$, $\{\alpha_1, \cdots, \alpha_N\}$, and $\mathbf{B} = \left[\mathrm{B}_0^T, \cdots, \mathrm{B}_K^T\right]^T$.
3: Calculate $\mathbf{S}_t^{\text{off}} = \alpha_t \mathbf{B}$; $\alpha_t = [\alpha_{t,0}, \cdots, \alpha_{\mathrm{t,K}}]$; $t = 1 \ldots N$.
4: For t=N+1, initialize model to mean shape $\bar{\mathbf{S}}_{t-1}$, eigenvectors $\mathbf{U}_{\text{t-1}}$, and projected vector A_{t-1} estimated via batch PCA for $\mathbf{S}^{\text{off}}$.
5: **loop**
6: Input new frame f_t with 2D correspondence points.
7: Build local measurement matrix $\mathbf{W}_t = \left[\mathbf{w}_{t-l+1}{}^T, \cdots, \mathbf{w}_t{}^T\right]^T$
8: Using Levenberg-Marquardt algorithm solve: $\left\{\hat{\beta}, \hat{\tilde{\mathbf{B}}}, \hat{\mathbf{R}}\right\} = \underset{\beta, \tilde{\mathbf{B}}, \mathbf{R}}{\arg\min}\,(\varepsilon)$ where ε is given by Equation 4.4, $\hat{\beta} = [\hat{\beta}_1, \cdots, \hat{\beta}_l]$, $\hat{\tilde{\mathbf{B}}} = [\hat{\tilde{\mathrm{B}}}_1^T, \cdots, \hat{\tilde{\mathrm{B}}}_M^T]^T$, $\hat{\mathbf{R}} = \{\hat{\mathrm{R}}_1^T, \cdots, \hat{\mathrm{R}}_l^T\}$.
9: Compute current shape:$\mathbf{S}_t = \mathbf{S}^{\text{off}} + \hat{\beta}_l \hat{\tilde{\mathbf{B}}}$ and rotation: $\mathrm{R}_t = \hat{\mathrm{R}}_l$.
10: Compute the projection of $\mathbf{S}_t$ on the shape space:$\mathrm{a} = \mathbf{U}_{t-1}{}^{\mathrm{T}}(\mathbf{S}_t - \bar{\mathbf{S}}_{t-1})$
11: Compute the residual vector $\mathbf{r}_t = \mathbf{S}_t - \left(\mathbf{U}_{t-1}\mathrm{a} + \bar{\mathbf{S}}_{t-1}\right)$
12: **if** $(\|\mathbf{r}\| > r_T)\, or\, (\|\mathbf{r}_t\| - \|\mathbf{r}_{t-1}\| > 0.1)$ **then**
13: Update the shape space and the corresponding shape vectors as defined in the Incremental PCA.
14: Re-estimate the current frame by nonlinear optimisation with new eigenvectors $\mathbf{U}_t$ $\mathbf{S}_t$ and $\mathbf{R}_t$
15: **end if**
16: Update mean shape $\bar{\mathbf{S}}_t = \bar{\mathbf{S}}_{t-1}(t-1)/t \; + \mathbf{S}_t/t$
17: go to next frame, $t \leftarrow t+1$.
18: **end loop**

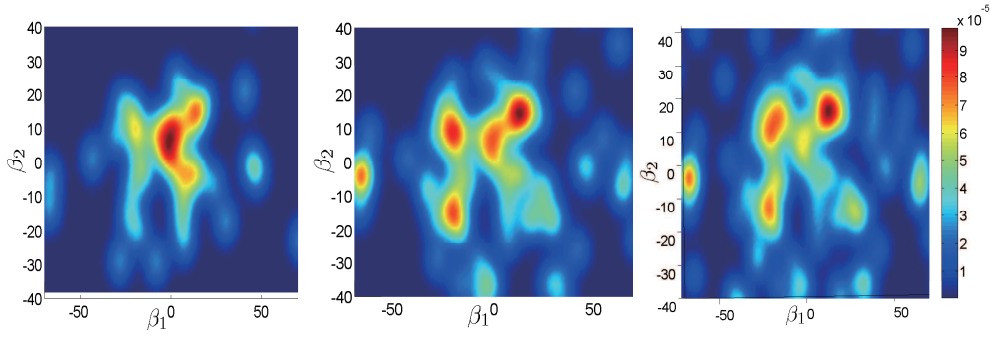


Figure 4.2: Shape coefficients probability distribution for the first two basis shapes estimated for frame 23, 38 and 58 respectively

forced to be close to learned eigenvectors $\mathbf{U}$, which are updated on each iteration. The constraint imposed on the basis shapes is given by:

$$\varepsilon_{bs} = \sum_{d=1}^{M} \left\| \tilde{\mathbf{B}}_d - \mathbf{U}_d \right\|^2 \tag{4.2}$$

According to Equation 3.4, the prior probability of the on-line shape coefficients β can be written as:

$$p(\beta) = \frac{1}{T} \sum_{i=1}^{T} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|\mathrm{A}_i - \beta\|^2}{2\sigma^2} \right) \tag{4.3}$$

Where A_i is the i_{th} shape vector. An example of the weight probability distribution for the first two weights is shown in Figure 4.2.

The cost function is built as:

$$\varepsilon = \sum_{t=f-l+1}^{f} \left\| \mathbf{w}_t - (\mathbf{S}^{\text{off}} + \mathbf{R}_t \sum_{d=1}^{\mathrm{M}} \beta_{td} \tilde{\mathbf{B}}_d) \right\|^2 + \varphi_1 \sum_{t=f-l+1}^{f} \left\| \mathbf{R}_t {\mathbf{R}_t}^T - \mathbf{I} \right\|^2 + \varphi_2 \varepsilon_{bs} - \varphi_3 p(\beta_t) \tag{4.4}$$

Minimising ε by the same method applied in Equation 3.9.

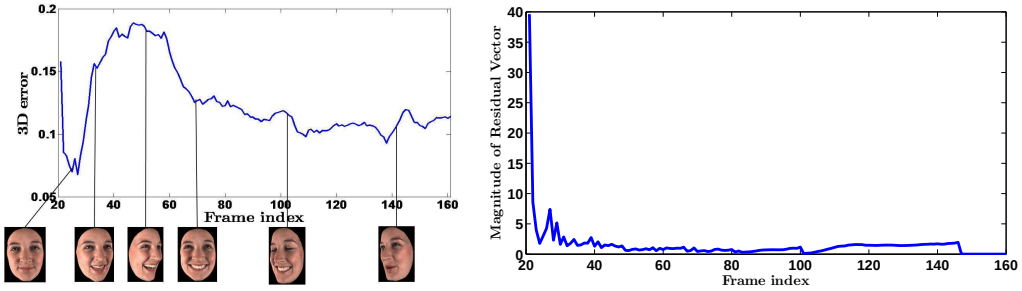
4.4 Experimental results

In this section, the experiments are designed to evaluate the performance of the proposed recursive approach. We also produce comparison results between the proposed sequential approach and the batch approach (described in Chapter 3) on different sequences. In all the experiments, the 3D errors are calculated by normalised mean 3D error over all frames and all points using Equation 3.17. The data we used for experiments was introduced in Section 3.8.

4.4.1 Evaluation

First we tested the proposed incremental approach on articulating facial expression sequence with ground truth data. Figure 4.3 shows representative sequential results. The top graph plots the 3D reconstruction error, with selected illustrative corresponding faces; the bottom shows the magnitude of residual vector of reconstructed shapes for each frame. The input is an image sequence with a facial expression of happiness. As expected, at first the error increases as each new frame arrives. Once the online adaptive learning algorithm has learned the shapes, the error decreases gradually. The error increases as new types of variations appear, but the algorithm can still learn quickly. As the new shapes occur, which is the case when the shape is a variation of a similar shape which has already been learned, the residual drops off to almost zero, an incidence of this is seen in the last 10-15 frames.

In the on-line reconstruction, some frames may be dropped when the calculations for the current frame have not finished before the next frame arrives. The following experiment was designed to test the sensitivity of our method with respect to percentage of missing frames. The simulated missing frames were selected randomly at 5%, 10%, 20% and 40% of the total number of frames. The results for the happiness facial



(a) (b)

Figure 4.3: (a) 3D reconstruction error for each frame with selected corresponding faces. (b) Magnitude of residual vector of reconstructed shapes for each frame.

expression sequence are shown in Table 4.1. The average 3D error and the standard deviation were estimated based on 10 trials. The 3D reconstruction error for this sequence without missing data is 0.0980. The experiment has been repeated twice, with respectively one and two frames missing at any given time. It should be noticed that the method is not very sensitive with respect to the number of missing frames. This can be explained by the fact that the proposed algorithm does not explicitly model temporal variations of the data and therefore the method is not too sensitive to missing frames, as long as the data can be well modelled by the online learned shape space

Frame Missing%		5%	10%	20%	40%
One frame	Mean 3D error	0.1022	0.1139	0.1156	0.1229
	Max error	0.1186	0.1254	0.1351	0.1319
	Std.dev.	0.0072	0.0087	0.0105	0.0065
Two frames	Mean 3D error	0.1071	0.1129	0.1223	0.1247
	Max error	0.1173	0.1251	0.1293	0.1385
	Std.dev.	0.0076	0.0089	0.0045	0.0096

Table 4.1: Average 3D reconstruction error / Max 3D error / standard deviation for missing frames.

4.4.2 Sequential mode vs. Batch mode

We compare the performance of our algorithms, the proposed incremental approach IPCA is tested against BPCA which was introduced in last Chapter. Similarity, to better visualise the results, the data were divided into the same groups as indicated in Section 3.8.2. Table 4.2 summarises results of these tests. Although the same data was used for training, as observed in the table, IPCA significantly improves the reconstructed results since the online adaptive learning algorithm is applied to incrementally learn the shape variations also from the testing data. Considering that the training data only contain static facial expressions, which may not be able to represent all the shapes in the testing sequences, updating the probability distribution of weighting coefficients in terms of new estimated shapes is especially important.

Method		Mild	Middle	Extreme	Talking
BPCA	3D error	0.1193	0.1266	0.1641	0.1588
	Max. error	0.1862	0.1941	0.2956	0.2237
	Std. dev.	0.0306	0.0382	0.0564	0.0415
IPCA	3D error	0.0553	0.0591	0.0633	0.0599
	Max. error	0.0745	0.0736	0.0770	0.0752
	Std. dev.	0.0091	0.0068	0.0063	0.0114

Table 4.2: Average 3D reconstruction error / Max 3D error / standard deviation for our approaches

Other existing sequential algorithms [91, 100] for either rigid or non-rigid object recovery did not display any difference in the results, when compared with original batch method. This was expected, as these methods are essentially based on the same theory. The results demonstrate that the proposed method performs better than those algorithms without online shape model updates. This is because the probability distribution of shape coefficients is updated with incoming new shapes. As shown in Table 4.2, batch method BPCA is able to provide satisfactory results, but the errors are still

much bigger than the errors obtained for the proposed recursive method IPCA.

Sensitivity to noise and missing data

For real cases, most previously proposed approaches are very sensitive to noise, which lead failure to converge to correct solution. The next experiment was designed to test the influence of inaccurate measurement, by adding increasing levels of Gaussian noise to the measurement data $\mathbf{W}$. The algorithm introduced in Section 3.7 can be extended into the sequential approach for filling the missing entries in the measurement matrix.

We compare our batch and recursive methods with the other two batch approaches: factorisation with shape priors (SP)[37] and metric projection method (MP)[101] in terms of sensitivity to the noise present in the measurement data. The reconstruction errors are evaluated for 10 trials, with measurement error modelled by independent Gaussian noise. The level of additive noise is set to 2%, 4%, 6% and 8%. The results are shown in Figure 4.4. For higher levels of noise, the increase in average 3D error is similar for all four methods, but the proposed methods are relatively stable when compared to the previous methods and can achieve much smaller errors, especially the recursive method; even when the noise level has increased to 8%, the estimated maximum error is 0.1958.

We also performed a similar experiment using all the facial expression data to test the case when 2D observation is incomplete. Together with Figure 3.5(a), the reconstruction error for IPCA and BPCA are shown in Figure 4.5(a). Our recursive method IPCA has small standard deviation over all the tested levels of occlusion and achieved much smaller errors, both in terms of the mean and standard deviations. It is important to note that the results for talking sequences from BPCA has large errors when the amount of missing data increases, whereas the recursive method clearly outperforms batch method. It is because the basis shapes we used to predict missing

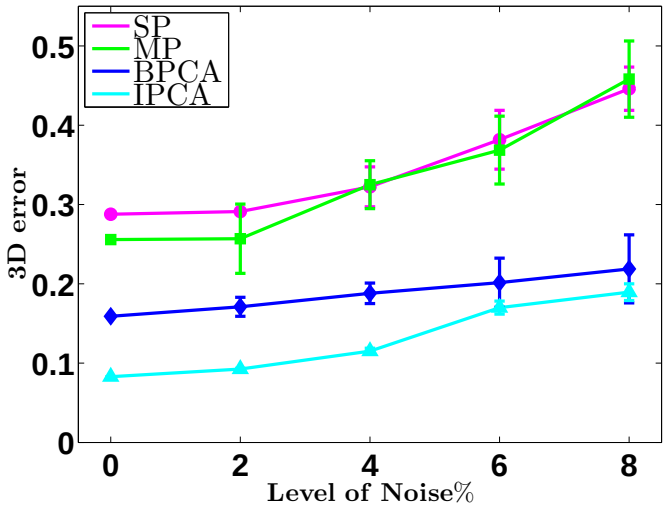


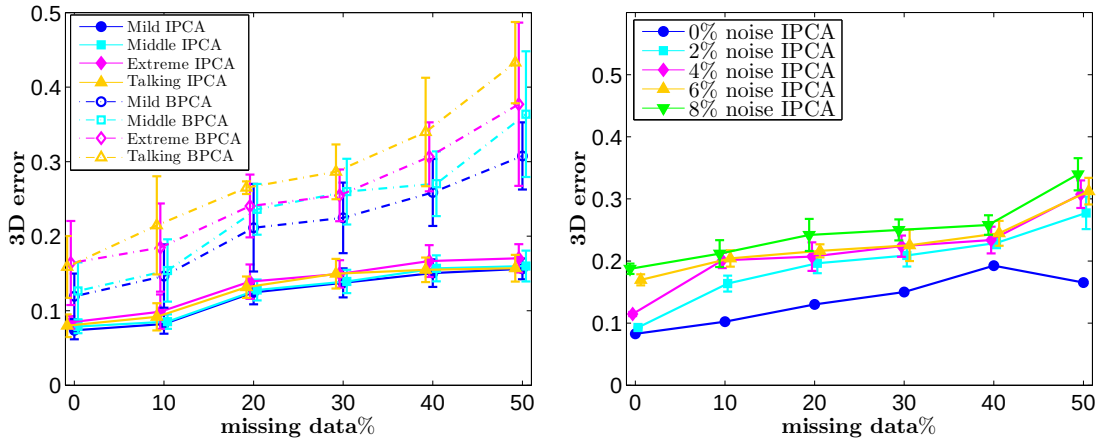
Figure 4.4: Sensitivity to noise, all the methods for the extreme surprise facial expression data.

measurement data are obtained from the updated eigenvectors, while the batch mode algorithm has only learned eigenvectors from the training data, which only contain facial expressions.

Figure 4.5(b) shows the results in the case of inaccurate and incomplete 2D measurements tested using extreme surprise facial expression sequence. Similar to Section 3.8.2, the missing points were selected randomly with levels 10%, 20%, 30%, 40%, 50% and the noise levels vary between 0% and 8%.

Visualised results

The results shown in Figure 4.6 are for tracking 83 points over a 229 frame sequence of a surprised facial expression. We present both front and side views of a selection of facial reconstructions extracted from the sequences, as well as the 2D images with extracted feature points. For comparison, the side view of reconstructions is to demonstrate the relative performance of depth information recovery. The front views of the results obtained from BPCA are not shown here, because they look very similar to the results



(a) (b)

Figure 4.5: (a) Reconstruction results for all of the facial expression data with occlusions. The figure shows the dependence on increasing amounts of missing data. (b)Reconstruction results for varying levels of missing data and 5 levels of noise for an extreme surprise facial expression sequence. Results using recursive method IPCA.

obtained from IPCA. As is visible in the figures, both approaches yield satisfactory reconstructions, whereas IPCA performs better in depth recovery. More comparison results on different facial expression sequences are shown in Appendix A.

4.5 Limitations

The main limitation of our approach is that the deformations of the object are represented in a linear subspace. An important problem is that the non-linear deformations are often observed. Although our method achieved satisfactory reconstructed results, it is only successful for small deformable objects, such as articulating face and simple human body movement. The current method is still unable to reconstruct the objects with large and complex deformations.

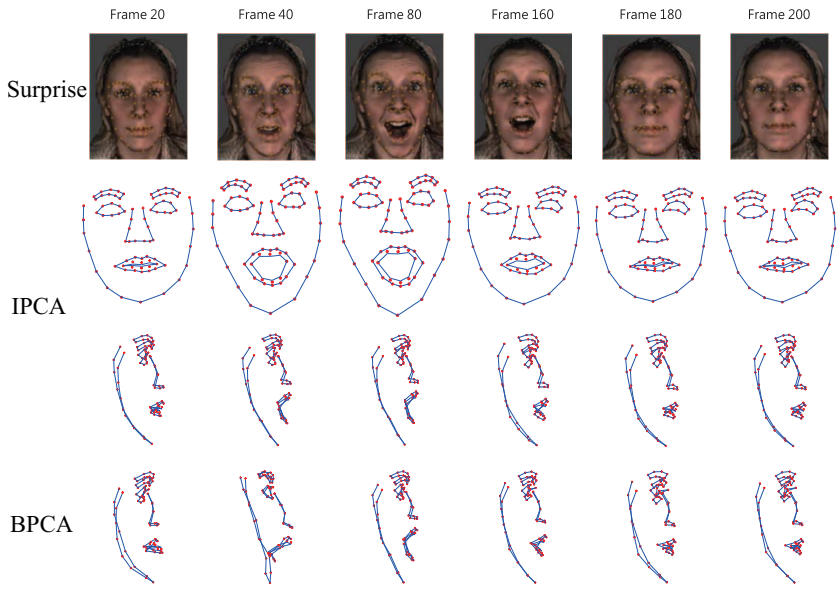


Figure 4.6: Results for surprised facial expression sequence. First row: Input images tracked with feature points. Second and Third row: Front and side views of the 3D reconstruction using IPCA. Fourth row: Side views of the 3D reconstruction using BPCA.

4.6 Summary

We have presented a new approach to solve the recursive deformable shape recovery problem and have demonstrated the accuracy and robustness of our method in a series of challenging situations. Our method successfully recovers shape and camera motion parameters as new frames arrive; additionally, it allows for updates to the model, thus accounting for new shape variations as objects deform over the sequence. We have also developed several extensions to the algorithm for deformable object recovery, which include use of learned shape model and distribution of the weights in the cost function, thus improving performance of the optimisation process. We believe our method is a suitable groundwork for later exploitation in real-time applications.

However, the current approach relies on a linear subspace model to represent the deformations of the object of interest. This approach is applicable to a relatively sim-

ple non-rigid object, especially when the reconstructed object is based on only a small number of basis shapes. To address this deficiency, we are currently working on shapes constrained to a smooth manifold representing learned nonlinear shape variability. The planned approach should be more accurate and well-adapted to large deformation models, which cannot be accurately represented by a linear subspace.

Chapter 5

Non-linear Manifold Learning in Deformable Shape Reconstruction: Part I

One of the existing limitations of the methods proposed so far is that they mainly address the problem of small deformations. The main reason for their failure when recovering objects with large, complex deformations is attributed to the reliance on a linear shape model. This chapter focuses on modelling non-linear deformable objects with large complex deformations, such as deformable cloth or articulated full-body motion. In this case, the existing methods based on linear manifold are no longer applicable. We argue that the linear models require more parameters than our method, which was based on the non-linear manifold learning approach. The proposed methodology has been validated quantitatively and qualitatively on 2D points sequences projected from the 3D motion capture data and real 2D video sequences. The comparisons of the proposed manifold based method against several state-of-the-art techniques are shown on different types of deformable objects.

5.1 Contributions

Note that the data dimensionality may not represent the true complexity of the problem, low-dimensional data is often embedded in much higher dimensional spaces. A specific type of shape variation might be governed by only a small number of parameters, therefore can be well represented in a low dimensional manifold. We learn a non-linear shape prior using the diffusion maps method. The method is able to reconstruct 3D deformable structures exhibiting large and complex deformations. The key contribution at this method is the introduction of the shape prior that constrains the reconstructed shapes to lie in the learned manifold.

5.2 Manifold learning techniques

In many problems, data is hard to represent or analyse due to its high dimensionality. However, such complex data might be governed by a small number of parameters. The goal of the manifold learning is to find the embedding function, mapping the data from a high dimensional space to a reduced dimensional space. Assuming $\mathcal{X}$ is a dataset with M samples, the goal of dimensionality reduction problems is to find an embedding Ψ from data $\mathcal{X} = \{\mathbf{X}_1 \cdots \mathbf{X}_M\}$ in a high N dimensional, observation space to a reduced n dimensional space $\{\mathbf{x}_1 \cdots \mathbf{x}_M\}$. A mapping is defined by:

$$\Psi : \mathbf{X} \mapsto \Psi(\mathbf{X}) = (\Psi_1(\mathbf{X}), \cdots, \Psi_K(\mathbf{X})), \text{ where } \mathbf{X} \in \mathbb{R}^N, K \ll N \tag{5.1}$$

This section describes some of the most important manifold learning techniques for dimensionality reduction problem. We start with a brief introduction to linear manifold and demonstrate the limitations of linear methods.

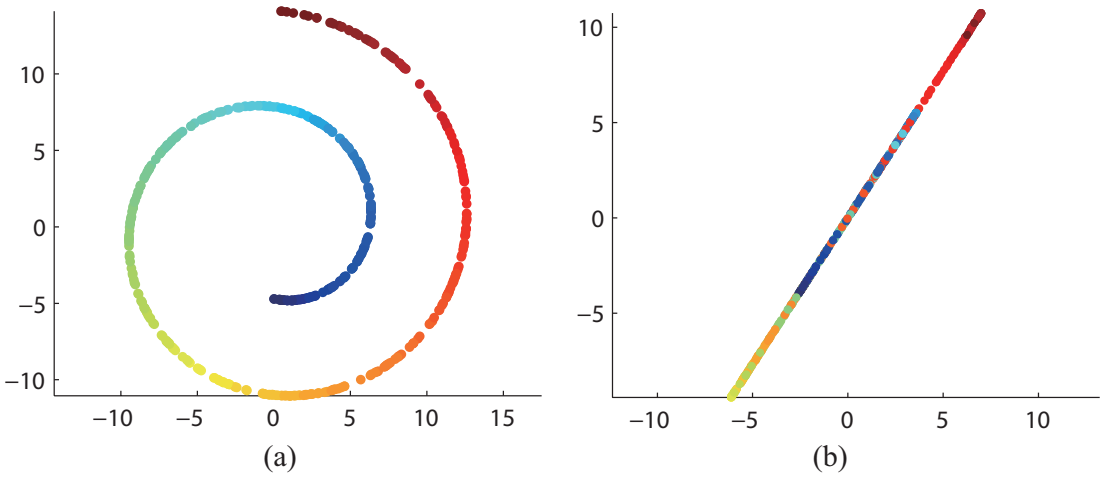


Figure 5.1: An example that linear methods cannot handle non-linear datasets. (a) Original 2D spiral data. (b) Linear mapping PCA from the original 2D space to the 1D real line is colour coded.

5.2.1 Linear manifold learning

PCA is the most widely used linear dimensionality reduction technique. The goal is to find an optimal subspace which captures as much of the variability in the data as possible. The subspace is defined by only a few principal components of the data covariance matrix. PCA is simple and efficient, as presented in Chapter 3; only using the first few components is enough to interpret the whole datasets, such as human facial expressions.

The linear manifold techniques are successful if the relationship between the variables is linear, but can fail to explain any non-linear co-variability present in the measurements. Figure 5.1 is an example that PCA cannot explain non-linear spiral data. The drawback of linear methods is that they try to preserve large distances between data points. However, in some cases, distances are only meaningful in local neighbourhoods. The following section presents non-linear graph-based methods which address this problem.

5.2.2 Graph-based methods

In contrast to linear methods, graph-based methods are non-linear and are able to handle a wider range of data variability and preserve local structures at the same time. The linear manifold method like PCA is straightforward, the recovered input data lies on a linear subspace of high dimensional space. The problem with this is that the input data may have complex non-linear dependencies and preserving local or indeed global structures in the data may not be possible utilising linear projections. The graph-based algorithms demonstrate a major advantage over the classical linear dimensionality reduction methods. They are non-linear and preserve local geometry of the data.

Graph-based algorithms usually consist of the following steps:

First, build the similarity graph $\mathbf{G}$ of the data. The connectivity of the data is represented using a local similarity measure. Contrary to the global methods, in which all the connections between data are being considered, the local graph only defines the distance within a certain neighbourhood. Outside the neighbourhood, the distance between pair of data can be seen as infinity.

In order to estimate the local properties, kernel function $k(\mathbf{X}_i, \mathbf{X}_j)$ is applied to the graph and used to define a weighted adjacency matrix $\mathbf{Y}$ of the graph $\mathbf{G}$. For example, applying a Gaussian kernel to the graph can be written as $Y_{ij} = \exp\left(-\|\mathbf{X}_i - \mathbf{X}_j\|^2/2\delta\right)$, where δ is a scale parameter. Each entry of $\mathbf{Y}$ is calculated as $Y_{ij} = k(\mathbf{X}_i, \mathbf{X}_j)$ if i^{th} and j^{th} vertex are connected, otherwise $Y_{ij} = 0$. More generally, the kernel function satisfies the following properties:

1. Symmetric: $k(\mathbf{X}_i, \mathbf{X}_j) = k(\mathbf{X}_j, \mathbf{X}_i)$.
2. Non-negative preserving: $k(\mathbf{X}_i, \mathbf{X}_j) \geq 0$.

According to the built adjacency matrix, the optimal embedding is able to preserve

the local geometry of the original data.

One typical group of graph-based methods is called "kernel eigenmap methods" which consists of some well-known techniques, such as Locally Linear Embedding (LLE) [115], Laplacian Eigenmaps [12] and Isomap [127]. As proved in [29], all these methods can be seen as special cases of a general framework based on diffusion processes, which is termed Diffusion maps.

5.2.3 The diffusion maps

Diffusion maps is a graph based technique with quasi-isometric mapping from original shape space to reduced low-dimensional diffusion space. It has become a popular method in data dimensionality reduction given their capability to recover underlying structures of a complex manifold, as well as robustness to noise and data outliers.

We firstly recall the original framework of diffusion maps as described in [29]. Given a set of shapes $\mathbf{X}_1 \cdots \mathbf{X}_M \in \mathcal{M}$, where $\mathcal{M}$ is the manifold embedded in $\mathbb{R}^N$, Euclidean distance for each pair of shapes $\|\mathbf{X}_i - \mathbf{X}_j\|^2$ is calculated to build a similarity graph. The entries of the adjacency matrix $Y_{ij}, i, j \in 1 \dots M$ define the weighted similarity graph for all connected vertexes. Using Gaussian kernel $Y_{ij} = \exp(-\|\mathbf{X}_i - \mathbf{X}_j\|^2 / 2\delta)$ in this case, where δ is chosen to be the average smallest non-zero value of $\|\mathbf{X}_i - \mathbf{X}_j\|^2$ which calculated as $\delta = \frac{1}{M} \sum_{i=1}^{M} \min_{j:\mathbf{X}_i \neq \mathbf{X}_j} \|\mathbf{X}_i - \mathbf{X}_j\|^2$. As mentioned before, instead of connecting all the data to learn the low-dimensional representation, diffusion maps as a sparse spectral technique only focuses on preserving the local similarities measured in the data space. Therefore we apply k-nearest neighbour (kNN) sparsification scheme, retaining k edges for each point and removing other connections to avoid outliers.

Since mapping the shapes to the reduced space $\mathbb{R}^n$ is not unique, the optimal embedding is proved to be the eigenvalues and the associated eigenvectors of the diffusion operator. The operator $\mathbf{P} = \mathbf{D}^{-1}\mathbf{Y}$, where degree matrix $\mathbf{D}$ is diagonal with

$d_{ii} = \sum_j Y_{ij},$ and $d_{ij} = 0 \ \forall i \neq j$, thus each entry of the operator $\mathbf{P}$ is constructed as $P_{ij} = Y_{ij}/d_{ii}$, which can be interpreted as the probability of transition from $\mathbf{X}_i$ to $\mathbf{X}_j$.

The similarity of the shapes can be represented by diffusion distance, which describes the intrinsic geometry of the data. The diffusion distance between two points in higher data space is equivalent to the Euclidean distance in the reduced diffusion space (The justification is provided in Appendix B), which is defined as,

$$L\left(\mathbf{X}_i, \mathbf{X}_j\right) = \|\Psi(\mathbf{X}_i) - \Psi(\mathbf{X}_j)\| \tag{5.2}$$

The diffusion distance can be computed using eigenvalues λ_l and eigenvectors φ_l of $\mathbf{P}$,

$$L^2\left(\mathbf{X}_i, \mathbf{X}_j\right) = \sum_{l \geq 1} \lambda_l^2 \left(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j)\right)^2 \tag{5.3}$$

Thus, the embedding for diffusion maps is derived as,

$$\Psi(\mathbf{X}_i) \mapsto \left[\lambda_1 \varphi_1(\mathbf{X}_i), \cdots, \lambda_K \varphi_K(\mathbf{X}_i)\right]^T \tag{5.4}$$

The scheme given in Algorithm 4 summarises the diffusion maps.

Algorithm 4 Outline of Classical Diffusion maps

1: Create similarity graph
2: Apply kernel function to the graph and build the adjacency matrix $\mathbf{Y}$, in which $Y_{ij} = \exp(-\|\mathbf{X}_i - \mathbf{X}_j\|^2/2\delta)$, $Y_{ij} \in \mathbf{Y}$.
3: Compute degree matrix $\mathbf{D}$, in which $d_{ii} = \sum_j Y_{ij},$ and $d_{ij} = 0 \ \forall i \neq j$, $d_{ii}, d_{ij} \in \mathbf{D}$
4: Build diffusion operator $\mathbf{P}$, in which $P_{ij} = Y_{ij}/d_{ii}$, $P_{ij} \in \mathbf{P}$.
5: Define embedding Ψ for diffusion maps in Equation 5.4.

Laplace-Beltrami operator

The Laplace-Beltrami operator was firstly introduced in [30] for providing the density invariant embedding of the data. As claimed in [94], the embedding provided by

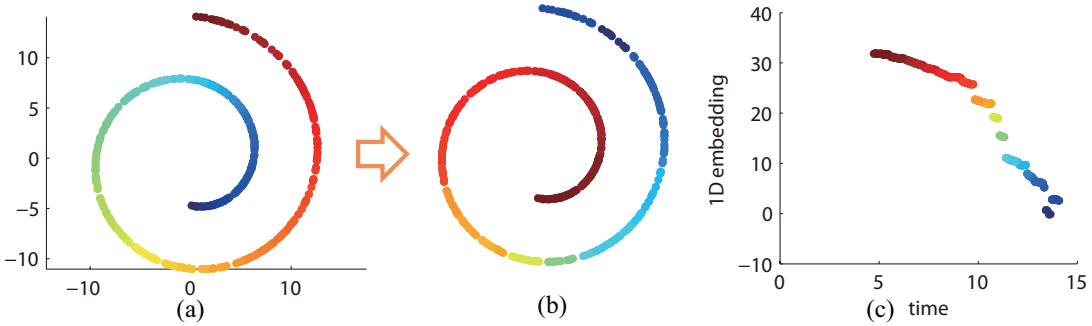


Figure 5.2: (a) Original 2D spiral data. (b) Non-linear method Diffusion maps mapping data from the original 2D space to the 1D real line is colour coded. (c) Embedding data in 1D. Diffusion maps does capture correctly the intrinsic 1D manifold.

eigenmap methods depends both on density and geometry of the data points. But the density may be unrelated to intrinsic geometry, thus a good representation of the data should not be variant to the density. The operator is similar to the diffusion operator, but with an additional re-normalisation step. Building Laplace-Beltrami operator summarised in Algorithm 5, is used to replace Step 3 and 4 in Algorithm 4.

Algorithm 5 Building Laplace-Beltrami operator

1: Define density $q(\cdot)$ as $q_i = \sum_{j=1}^{M} Y_{ij}$
2: Renormalise adjacency matrix $\hat{Y}_{ij} = Y_{ij}/q_i q_j$
3: Apply the normalised graph Laplacian construction to the renormalised adjacency matrix $d_i = \sum_{j=1}^{M} \hat{Y}_{ij}$
4: Define Laplace-Beltrami operator $P_{ij} = \frac{\hat{Y}_{ij}}{d_i}$

When embedding the data via the Laplace-Beltrami approximation in diffusion maps, we only need to replace the diffusion operator with the Laplace-Beltrami operator.

Figure 5.2 shows the embedding of "Spiral data" using diffusion maps with the Laplace-Beltrami operator. The "Spiral data" was originally shown in Figure 5.1, which cannot be modelled by the linear method. However, when using diffusion maps, 2D data can be well-represented in one dimensional reduced space.

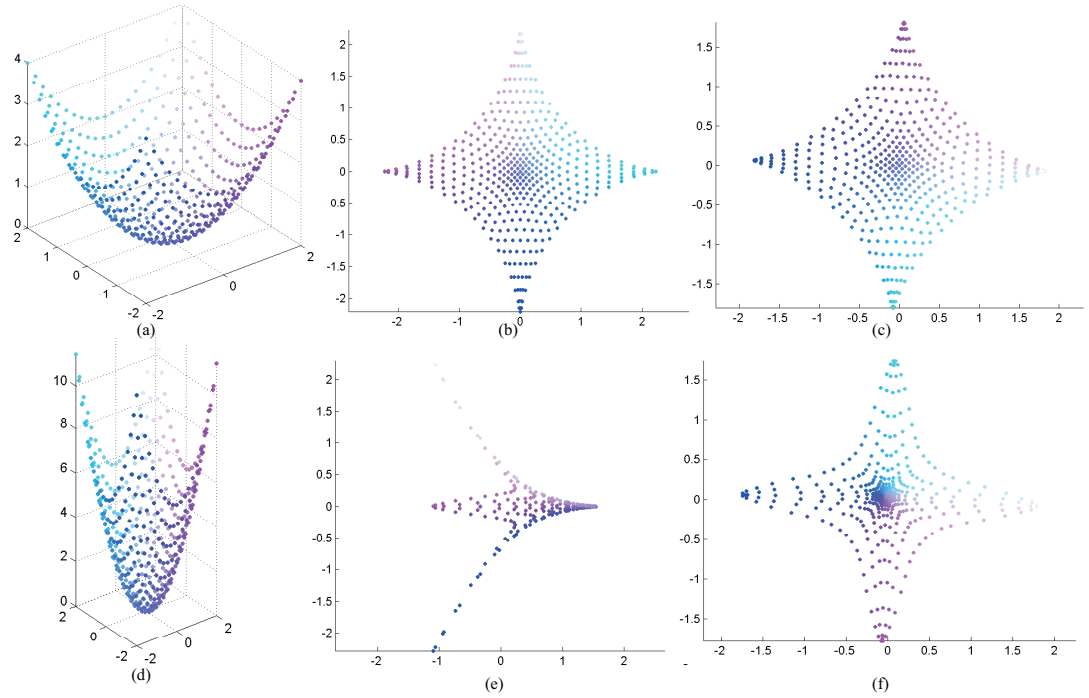


Figure 5.3: (a)Parabola surface with $\psi = 2$. (d)Parabola surface with $\psi = 0.7$. (b,e) Embedding of (a,d) via fully connected graph. (c,f) Embedding of (a,d) via k-nearest neighbour graph.

Figure 5.3 illustrates the simulated parabola surface given by the equation $f(x, y) = \frac{x^2+y^2}{\varphi}$ and its corresponding embeddings in two dimensional reduced space. In the figure, the left column illustrates two parabola surfaces with different value of ψ. The middle column represents the embedding via Laplace-Beltrami approximating using a fully connected graph. The right column is the embedding using k-nearest neighbour graph. The results suggest that building sparse graph for manifold learning focuses on retaining the local similarities measured in the input space.

A real data example is shown in Fig.5.4, where we illustrate the embedding of shapes from *cardboard* data [138] together with representative corresponding shapes extracted from 1000 training samples.

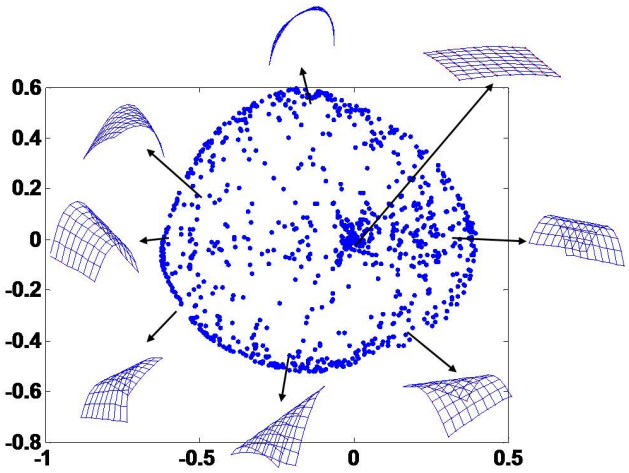


Figure 5.4: The reduced space of *cardboard* dataset

5.3 Shape model comparison

Recalling the Equation 2.13 in Section 2.3.1, the measurement matrix is denoted by $\mathbf{W} \in \mathbb{R}^{2F \times P}$ which contains 2D input points $\mathbf{x}_{tp} = [x_{tp}, y_{tp}]^{\mathrm{T}}$ with indices t and p referring to the p^{th} point in the t^{th} image.

$$\mathbf{W} = \begin{bmatrix} \mathbf{x}_{11} & \cdots & \mathbf{x}_{1P} \\ \vdots & \mathbf{x}_{tp} & \vdots \\ \mathbf{x}_{F1} & \cdots & \mathbf{x}_{FP} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{R}_F \end{bmatrix} \begin{bmatrix} -\mathbf{S}_1 - \\ \vdots \\ -\mathbf{S}_F - \end{bmatrix} = \mathbf{RS} \tag{5.5}$$

Without loss of generality, we assume that the coordinates of the feature points are given with respect to the centre of gravity calculated for all the points in the corresponding image. We also assume that the orthographic projection accurately models the image acquisition. The goal is to recover camera orientations matrix $\mathbf{R}$ and the concatenated time-varying shapes matrix $\mathbf{S}$, based only on the 2D measurement in matrix $\mathbf{W}$. It is an under constrained problem since shape and motion are both changing with time. Thus Equation 5.5 cannot be directly solved. Low rank shape model and

smooth trajectories model are successfully employed to deal with this problem. We now review these two models and propose our new non-linear manifold model.

Low-rank shape model

As introduced in Section 2.3.2, a deformable 3D shape $\mathbf{S}_t$ can be represented as a linear combination of K unknown but fixed basis shapes $\mathbf{B}_l$:

$$\mathbf{S}_t = \sum_{l=1}^{K} \alpha_{tl} \mathbf{B}_l \tag{5.6}$$

where the deformation coefficients α_l are adjustable over time. This low-rank shape model can be obtained by performing Singular Value Decomposition (SVD) or Principal Components Analysis (PCA). The measurement matrix can be decomposed and represented by pose, basis shapes and time varying coefficients matrices, therefore it can be rearranged as Equation 2.17. Since basis shapes $\mathbf{B} \in \mathbb{R}^{3K \times P}$, and $\mathbf{M} \in \mathbb{R}^{2F \times 3K}$ the rank of measurement matrix $\mathbf{W}$ is $3K$ at most in the absence of noise. The factor $\mathbf{M}$ and $\mathbf{B}$ are computed by factorising the measurements $\mathbf{W}$. The solution is not unique and is defined up to an ambiguity matrix $\mathbf{Q} \in \mathbb{R}^{3K \times 3K}$. According to [150], the limitation of the closed-form solution in this approach is that the motion matrix is nonlinear; when an inaccurate set of basis shapes have been chosen, it may not be possible to remove the affine ambiguity.

Smooth trajectories model

According to the duality theorem, described in [4], representing a non-rigid shape using the above shape basis model is dual to trajectory basis model, in which each point trajectory is represented as a K dimensional point within an unknown linear trajectory space. The trajectory for each point is approximated by a linear combination of a small

number of basis trajectories $\mathbf{A}_l$:

$$\mathbf{T}_p = \sum_{l=1}^{K} \mathbf{A}_l \boldsymbol{\beta}_{pl} \tag{5.7}$$

where $\boldsymbol{\beta}_{pl}$ are 1×3 coefficient vectors for the basis trajectory The basis trajectory can be predefined in an object independent way using Discrete Cosine Transform (DCT) basis and therefore avoid the training process. The model only needs to consider camera parameters and trajectory coefficients, thus requires less parameters than the shape basis model (see Table 5.1).

Non-linear manifold model

Our model departs from the linear shape model. The shape basis $\mathbf{B}$ in the proposed method are selected from the learned shape manifold. Unlike the low rank shape model, where all the reconstructed shapes are represented as a linear combination of unknown but fixed K basis shapes, in the proposed method, the basis shapes may be different in each frame. Although it may seem to increase the number of parameters in the model, it should be recognised that all the basis shapes are selected from the manifold and are not estimated as a part of the optimisation process. The parameters to be estimated in the proposed approach include only the camera motion and shape coefficients, representing the shape in the local linear barycentric coordinates system approximating the manifold at the location corresponding to the current estimate of $\mathbf{S}_t$.

Comparing the three models, the number of unknowns for each model is given in Table 5.1. In most cases, K<10, F, P>100, the proposed model requires less parameters than low rank shape model and has a similar order of magnitude as the trajectory model. Although the number of parameters depends on number of frames F in our method, it is important to note that they are not depending on the number of feature points P.

That makes our approach suitable for a shape which contains large number of feature points.

	Shape	Trajectory	Proposed
Camera	$3F$		
Coefficients	FK	$3KP$	$F(K+1)$
Basis	$3KP$	/	/
Total	$3F+FK+3KP$	$3F+3KP$	$3F+F(K+1)$

Table 5.1: Comparison of number of unknowns in low-rank shape model, trajectory model and our proposed non-linear manifold model

5.4 Deformable shape reconstruction

In this section, an overview of the proposed manifold based reconstruction algorithm is given, followed by a detailed description of the diffusion maps including description of out-of-sample and pre-image problems.

As known from [150], enforcing only the rotation constraints cannot guarantee a unique solution for the camera motion and the basis shapes. To solved this, the designed shape prior can help to attract a shape towards the manifold and therefore avoid incorrect reconstruction.

A summary of the algorithm for recovery of non-rigid object and estimation of camera motion is given in Algorithm 6. Initial shapes $\mathbf{S}'$ and camera motion $\mathbf{R}'$ are estimated by running a few iterations of the optimisation process in batch NRSfM, using the linear basis shapes model introduced in Section 3. For each initial shape, a Nyström extension is used for embedding these new samples into the reduced space. Intuitively, if the points in reduced space are relatively close, the corresponding shapes in high-dimensional space should represent similar shapes. Based on this observation, the reconstructed shape in each frame can be represented as the weighted sum of $K+1$ basis shapes from the learned manifold (The selection of number of K is discussed

in Section 3.6). The coefficients of corresponding basis shape are calculated based on the barycentric coordinates of $K+1$ closest points in reduced space. Once the basis shapes and their coefficients have been obtained, an optimisation is applied to minimise the image reprojection error with an additional smoothing term and basic rotation constraint over all frames. However, the quality of the optimisation result depends on the accuracy of initial shapes. Updating basis shapes in each iteration can help to circumvent the problem. The basis shapes are being kept updated as long as 2D measurement error r_t exceeds the defined threshold r_T (10^{-3} in our case) or the error between two adjacent frames is relatively large, which implies that the current results are unlikely to explain the shapes well.

Algorithm 6 Outline of Diffusion Maps based reconstruction

Input: Stream of 2D observations, diffusion map Ψ of training dataset $\mathbf{X}$ (Section 5.2.3)

Output: 3D deformable shapes $\mathbf{S}$ and camera motion $\mathbf{R}$ for each frame.

1: Initialisation of estimating Initial shapes $\mathbf{S}'$ and camera motion $\mathbf{R}'$.
2: **while** $(\|r\| > r_T)\, or\, \left(\|r_t\| - \|r_{t-1}\| > 10^{-3}\right)$ **do**
3: Shape projection onto manifold (shape Embedding) (Section 5.4.1)
4: Find $K+1$ closest points $\mathbf{b}_l, l = 1 \cdots K+1$ in low dimensional space, where K is the dimensionality of the reduced space.
5: Shape update (Section 5.4.2)
6: Non-linear optimisation by minimising 2D measurement error and shape smooth term to obtain updated shapes $\mathbf{S}_t$ and camera motion $\mathbf{R}_t$,$t=1 \cdots F$.(Section 5.4.3)
7: **end while**

5.4.1 Out-of-sample extension

In general, the diffusion map Ψ is only able to provide an embedding for the data which is given in the training set. However, in our reconstruction algorithm, it is necessary to calculate embedding for shapes which are not presented in the training set. Instead of re-training the whole manifold, a more efficient way is to assimilate the shape into the

lower dimensional feature space using both the embedding function and the geometric relation of new data with training samples. To extend the embedding for new data, the mapping can be approximated with the Nyström extension [6, 14].

The Nyström extension

In [14], the authors describe a series of extensions for eigendecomposition based unsupervised learning algorithms, such as LLE, Isomap, Laplace eigenmaps, and MDS. The idea is to extend the current embedding function known from the training set to a new point using Nyström extension, which is one of the popular techniques employed in machine learning.

Nyström extension can be easily extended to Diffusion maps. Suppose $\mathbf{S}_t \in \mathbb{R}^N$ is a new data which has not been presented in the training set. Knowing that for every sample in training dataset:

$$\forall \mathbf{X}_i \in \mathcal{X}, \sum_{\mathbf{X}_j \in \mathcal{X}} p(\mathbf{X}_i, \mathbf{X}_j)\varphi_k(\mathbf{X}_j) = \lambda_k \varphi_k(\mathbf{X}_i), k = 1 \dots M \tag{5.8}$$

Having a shape $\mathbf{S}_t$ not present in the training set $\mathcal{X}$, an embedding $\mathbf{S}_t \mapsto \left(\hat{\Psi}_1(\mathbf{S}_t), \cdots, \hat{\Psi}_K(\mathbf{S}_t)\right)$ of this new shape is calculated from:

$$\hat{\Psi}_k(\mathbf{S}_t) = \sum\nolimits_{\mathbf{X}_j \in \mathcal{X}} p(\mathbf{S}_t, \mathbf{X}_j)\varphi_k(\mathbf{X}_j) \tag{5.9}$$

where $p(\mathbf{S}_t, \mathbf{X}_j)$ is calculated the same as in Diffusion maps.

Related extension algorithms such as "geometric harmonics" proposed in [78] and manifold regularisation based natural extensions, developed by Belkin et al. [13], are also possible to solve the problem.

5.4.2 The pre-image problem

The pre-image problem is concerned with finding the inverse mapping of a point $\mathbf{x} \in \mathbb{R}^K$ given in the reduced space back to the manifold $\mathbf{X}_i = \Psi^{-1}(\mathbf{x}_i)$,with $\mathbf{X} \in \mathbb{R}^N$. Assuming we look for a shape $\mathbf{S}_t$ given by its embedding $\mathbf{x}_t$, if this shape $\mathbf{S}_t$ does not exist in the training dataset, the exact pre-image might not be found in that case. To resolve this problem, Arias *et al.* [6] proposed to find an approximate pre-image by optimising a certain optimality criteria. Inspired by this, we assume that the pre-image can be represented as a linear combination of its neighbours on the manifold selected from the training samples. The simplest way to achieve this is to identify the $K+1$ nearest points of $\mathbf{x}_t$ in the reduced space. This can be efficiently calculated by using a Delaunay triangulation. Since diffusion maps provides quasi-isometric mapping, the data must keep a similar structure when embedded into the reduced space and therefore the neighbours on the manifold correspond to the closest neighbours in the reduced space. Each point $\mathbf{x}_t$ can be represented as $\mathbf{x}_t = \sum_{l=1}^{K+1} \theta_{tl}\mathbf{b}_{tl}$, where $\mathbf{b}_{tl}$ is the l^{th} nearest point of $\mathbf{x}_t$. The weights θ_{tl} are computed as the barycentric coordinates of $\mathbf{x}_t$, thus can be obtained by optimising the following function:

$$\arg\min_{\theta_{tl}} \sum_{t=1}^{F} \left\| \mathbf{x}_t - \sum_{l=1}^{K+1} \theta_{tl}\mathbf{b}_{tl} \right\|^2 with \sum_{l=1}^{K+1} \theta_{tl}=1, 0 \leq \theta_t \leq 1 \tag{5.10}$$

Once the weights θ_{tl} are estimated, The shape $\mathbf{S}_t$ can be approximated as a set of weighted training samples $\mathbf{S}_t = \sum_{l=1}^{K+1} \theta_{tl}\mathbf{B}_{tl}$, where the training sample $\mathbf{B}_{tl}$ is the pre-image of $\mathbf{b}_{tl}$.

5.4.3 Cost function

The cost function to be minimised consists of the reprojection error, shape smoothing terms and rotation constraint. The cost function is given as:

$$\arg\min_{\mathbf{R}_t,\theta_{tl}} \sum_{t=1}^{F} \|\mathbf{W}_t - \mathbf{R}_t\mathbf{S}_t\|^2 + \varphi_{\mathbf{S}} \sum_{t=2}^{F} \|\mathbf{S}_t - \mathbf{S}_{t-1}\|^2 + \varphi_{\mathbf{R}} \sum_{t=1}^{F} \varepsilon_{rot} \tag{5.11}$$

where $\varepsilon_{rot} = \left\|\mathbf{R}_t {\mathbf{R}_t}^T - \mathbf{I}\right\|^2$ enforces orthonomality of all $\mathbf{R}_t$. $\varphi_{\mathbf{S}}$ and $\varphi_{\mathbf{R}}$ are regularisation constants selected experimentally (0.1 and 1 in our case which has been selected based on a systematic search of the parameter space.). The cost function above was minimised by using Levenberg-Marquardt algorithm.

5.4.4 Iterative estimation

The accuracy of the optimised results strongly depends on initialisation, since the mapping in the out-of-sample extension is based on initial shapes. To eliminate the effect, we iteratively updated the shapes and motion by embedding current estimated shapes to the reduced space. The basis shapes are updated until the 2D measurement error is smaller than predefined threshold r_T and the error between two adjacent frames is small enough. Figure 5.5 illustrates an example of how the initial shapes are redistributed in the reduced space after the algorithm has converged.

5.5 Experimental results

The proposed methodology has been validated quantitatively, and qualitatively on both motion capture and real data for different types of deformable object. To demonstrate the advantages of our method over previously proposed methods, the experiments are mainly focused on reconstructing complex deformations. To demonstrate the perfor-

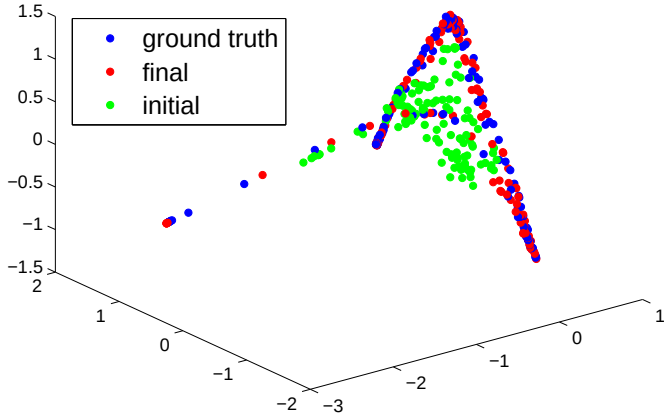


Figure 5.5: Embedded initial shapes (green dots), reconstructed shapes (red dots), together with ground truth shapes (blue dots) of *capoeira* sequence in reduced space.

mance of the algorithm, extensive experimental evaluation has been provided.

The models and algorithms used for comparison are as follows:

MP: The metric projection method [101].

PTA: The DCT based point trajectory approach [4].

CSF: The column space fitting method [51].

KSFM: The kernel non-rigid structure from motion [52].

IPCA: The incremental principal components analysis based method proposed in Chapter 4.

DM: The proposed method in this chapter.

The data which were used for testing include: two articulating face sequences, *surprise* and *talking*, both captured using a passive 3-D scanner with 3D tracking of 83 facial landmarks [85]; two surface models, *cardboard* and *cloth* [139]. Diffusion maps requires training process, so training datasets for two face sequences are taken from the BU-3DFE [152], and for two surface sequences are obtained from [139]. All the training data has been rigidly co-registered. The same testing data has been applied for other

methods, which do not require training.

5.5.1 The influence of embedding dimensionality

For the first set of experiments, we started with tests on motion capture data. The accuracy of 3D shape reconstruction is affected by the dimensionality of the manifold representing prior information. To find the relationship between manifold dimensionality and the reconstruction error, experiments were carried out with all the test sequences and dimensionality, changing between 3 and 10. To simplify the visualisation of results, all the sequences are separated into two groups, which are: facial sequences (*surprise*, *talking*), and surface sequences (*cardboard*, *cloth*). For evaluating the results, the normalised means of the 3D error were compared over all frames and all points, see Equation 3.17. Figure 5.6 shows the means of reconstruction error for each group and the overall average results when different manifold dimensions K are used. As expected, in general, increasing the number of manifold dimensions decreases error. This is especially true for the group of surface sequences, which represents relatively large deformations. Higher dimensional manifolds preserve more information from the original data leading to better results. However, for data with small deformations, the 3D error levels off and does not strongly depend on K. This does make sense as only a small number of basis shapes is required to describe the data variability, containing only a relatively small number of degrees of freedom.

5.5.2 Comparison with previous methods

For the comparative evaluation, performance of the proposed method is tested against all the five other approaches listed above for all 12 sequences. Table 5.2 summarises the results, showing 3D reconstruction errors of each method and each sequence, together with the optimal number of bases for which minimal reconstruction error on the test

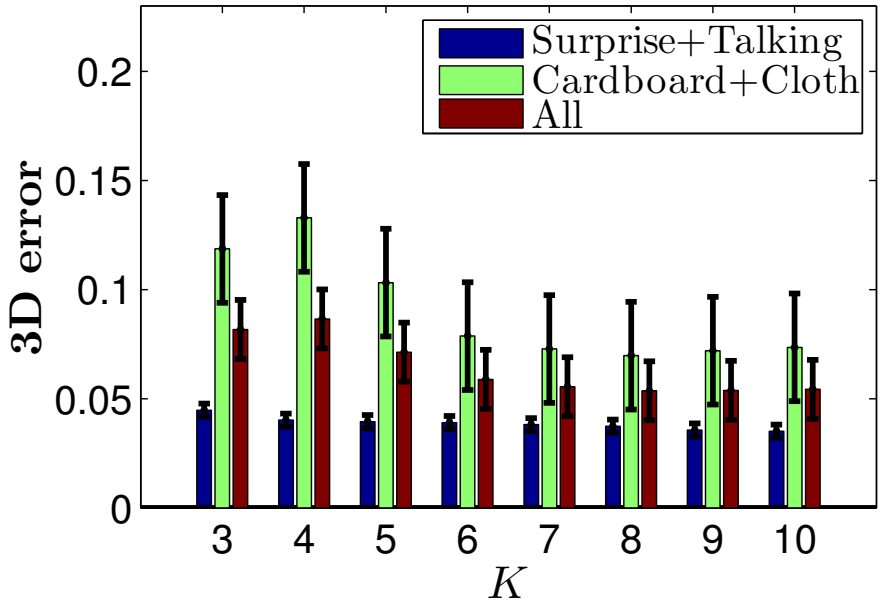


Figure 5.6: Average normalised mean 3D error and standard deviation of different number of dimensions in reduced space. Bars left to right: Group of facial sequences, group of surface sequences, , all the sequences.

data is obtained. We followed the same evaluation procedure as reported in [52]; the 3D errors of the PTA, CSF and KSFM methods are chosen with their best parameter K, by running the trials with K varying from 2 to 13. The best result for DM method is chosen by changing manifold dimension K from 3 to 10. Considering the ambiguity of estimated camera motion [4], the shapes are aligned using a single global rotation based on Procrustes alignment method.

As shown in the Table 5.2, trajectory based methods PTA, CSF and KSFM are able to provide results comparable to the proposed method on objects with small deformations (*e.g.* faces etc.). This is because these objects exhibit mostly a rigid motion, the deformations are only seen around the lips and chin. But those methods provide relatively large errors on highly non-rigid human motion sequences (*e.g.* dance etc.). DM is the only method that presents accurate reconstructed results almost every times, even for full-body motion capture sequences. Note that although the initial shapes of

	MP	PTA	CSF	KSFM	IPCA	DM
Surprise	0.2558	0.0386(12)	0.0396(3)	0.0381(4)	0.1289	**0.0352**(10)
Talking	0.0991	0.0862(10)	0.0573(3)	0.0498(4)	0.0986	**0.0350**(10)
Cardboard	0.4185	0.2894(8)	0.3237(3)	0.2753(2)	0.2445	**0.1064**(10)
Cloth	0.3997	0.3526(6)	0.2609(6)	0.1806(2)	0.1909	**0.0287**(7)

Table 5.2: Normalised mean 3D error calculated for different sequences.

our method may not belong to the manifold $\mathcal{M}$, after the optimisation process, the results demonstrate good convergence since the 3D errors are relatively small. An important observation is that, in the trajectory based methods, the optimal number of bases K has to be independently estimated for each sequence. Choosing too big K may lead to an ill-conditioned problem; but the point trajectory cannot be comprehensively represented if K is too small, while the results from our method are more predictable.

5.5.3 Real-data experiment

We tested our approach on a video sequence showing paper being bended, taken with a video camera. In the video, 81 features were tracked along 61 frames, showing approximately two periods of bending movement. Figure 5.7 shows a comparison of our reconstructed shapes with the results obtained from MP, PTA, KSFM methods.

5.6 Summary

The paper presented a new approach to integrate the idea from non-linear manifold learning techniques into the NRSfM framework, for the task of reconstructing complex and highly deformable shapes. The diffusion maps have been introduced in order to build non-linear shape prior manifold. This approach significantly improved the reconstruction quality and is well-adapted to large deformation of complex objects, especially

for non-rigid articulated body movement, which cannot be accurately represented in a linear subspace. The evaluation suggests that the robustness used by our approach is important in getting good results, even with noisy datasets.

It should be pointed out that the improved performance of the proposed method in terms of 3D shape reconstruction accuracy comes at the cost of required availability of a representative training dataset, and therefore the comparison of the proposed method with respect to the other methods may not be seen as fair. Indeed, in this sense it can also be argued that the method does not fit the definition of the SfM problem, due to the use of this additional information.

As we only use a limited number of shapes in the training process, the future work should focus on two different areas. One is collecting and generating data for building a sufficiently dense representation of the manifold to further improve the performance. The other is learning the manifold by only using a small number of training samples. Since in most cases, collecting sufficient number of 3D training data may not be acceptable, developing a method which is only based on small training set seems especially important. As manifold learning has shown to be a very powerful approach for analysis of the shapes, we believe the manifold based method is a suitable groundwork for the reconstruction of deformable shapes.

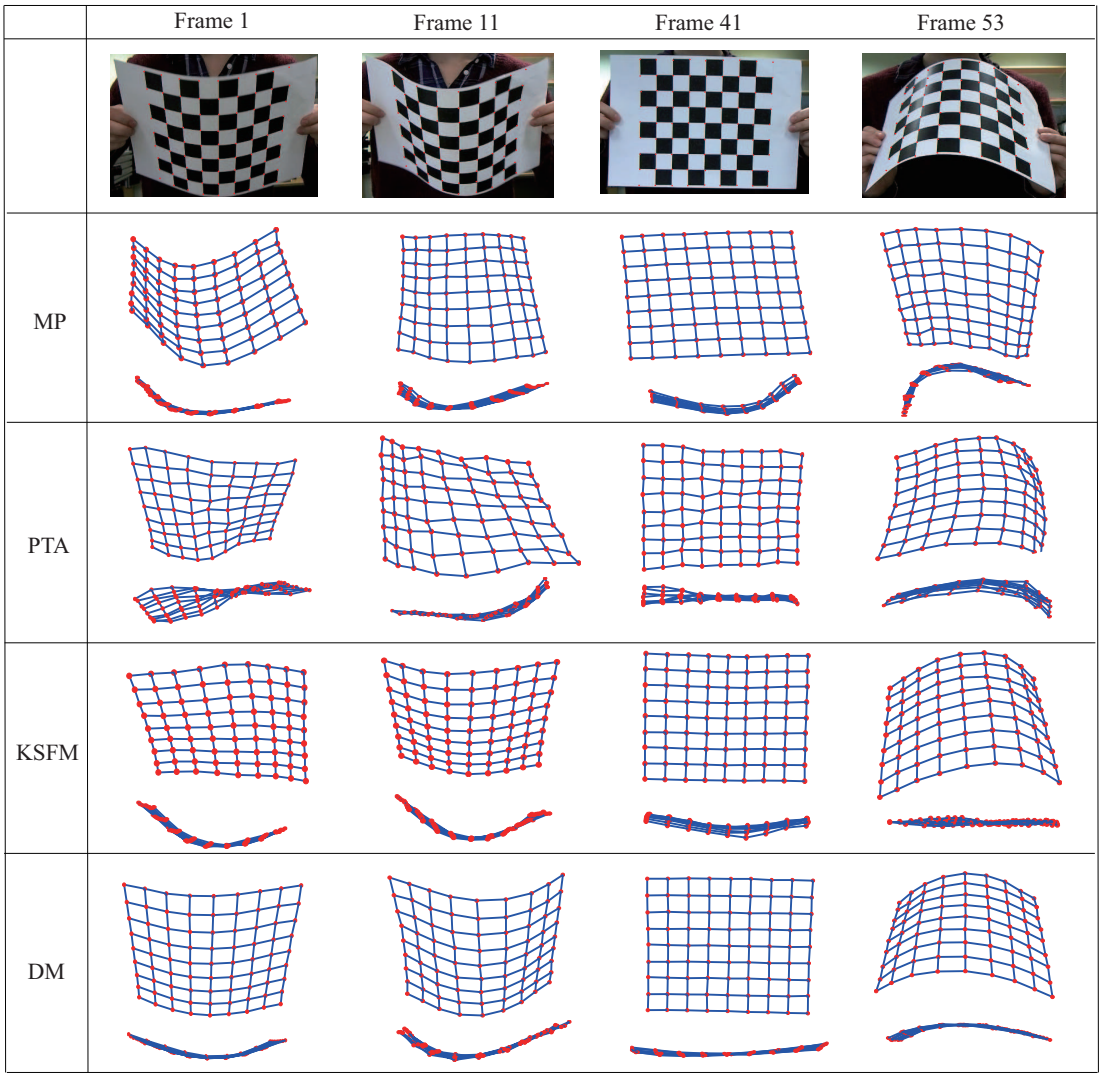


Figure 5.7: Selected 2D frames from the video sequence of a paper bending. Front and top views of the corresponding 3D reconstructed results using our method (DM), MP, PTA and KSFM

Chapter 6

Non-linear Manifold Learning in Deformable Shape Reconstruction: Part II

A common approach to recover structure of 3D deformable scene and camera motion from uncalibrated 2D video sequences is to assume that shapes can be accurately represented in linear subspaces. These methods are simple and have been proven effective for reconstructions of objects with relatively small deformations, but have considerable limitations when the deformations are large or complex. To solve this, in Chapter 5 the manifold learning techniques have been introduced and integrated into the problem of reconstruction of deformable objects. Although the method achieved better reconstruction performance, it still can be improved. Two methods presented in this chapter improve the current approach in two different aspects. First, the structure of data is learned from the data itself in the proposed method based on random forests techniques, rather than estimated using Euclidean distances between pairs of data items in the standard diffusion maps. Second, as claimed before, building a dense representation of the manifold requires a large amount of training data which is not feasible in many real applications. To address the problem, a method is proposed for estimating accu-

rate reconstructions by using a relatively small number of training samples. To better compare with previous method, the manifold is learned based on standard diffusion maps in this method. Both techniques described in this chapter are the extensions of the method previously proposed in Chapter 5.

Improved method I

The first part of this chapter describes a novel approach to reconstruction of deformable objects utilising a manifold decision forest technique. The key contribution of this work is the use of random decision forests for the shape manifold learning. The learned manifold defines constraints imposed on the reconstructed shapes. Due to nonlinear structure of the learned manifold, this approach is more suitable to deal with large and complex object deformations when compared to the linear constraints.

Deformable shape recovery from a single uncalibrated camera is a challenging, under-constrained problem. Most of the existing methods are restricted by the fact that they try to explain the complex deformations using a linear model. Recent methods have integrated the manifold learning algorithm to regularise the shape reconstruction problem by constraining the shapes as to be well represented by the learned manifold. Using shape embedding as initialisation was introduced in [110]. Hamsici et.al [57] modelled the shape coefficients in a manifold feature space. The mapping was learned from the corresponding 2D measurement data of upcoming reconstructed shapes, rather than a fixed set of trajectory bases.

Contrary to other techniques using manifold in the shape reconstruction, our manifold is learned based on the 3D shapes rather than on 2D observations. The proposed implementation is based on the manifold forest method described in [33]. The main advantage of using manifold forest as compared for example to standard diffusion maps [29] is the fact that in the manifold forest the neighbourhood topology is learned from

the data itself rather than being defined by the Euclidean distance. The method has been tested on different types of 3D motion capture data and real 2D video sequences. Performance of the proposed method has been assessed against several state-of-the-art algorithms, demonstrating that the method significantly outperforms the existing ones. To the best of our knowledge, random forests technique has never been applied in the context of non-rigid shape reconstruction. This work is the first to integrate the ideas of manifold forests and deformable shape reconstruction.

6.1 Randomized decision forest

Random forests have become a popular method, given their capability to handle high dimensional data, efficiently avoid over-fitting without pruning, and possibility of parallel implementation. We firstly give a brief review of the randomized decision forests and their use in learning diffusion map manifolds. Although other choices are possible, our method is focused only on the binary decision forest.

6.1.1 Decision tree

The Decision tree is one of the most popular classification and regression algorithms in data mining and machine learning field. The basics of decision trees were introduced in [21] by Breiman et al. Inspired by this model, other algorithms focused on learning optimal decision trees by selecting the best attribute to split the dataset at each node have been proposed. The typical ones are ID3 and C4.5 algorithms, both proposed by Quinlan [107, 108]. The decision trees in our method are built by making decision in each node of the tree based on randomly selected features. Like most machine learning algorithms, the operation of randomized decision trees can be divided into training and testing phases.

Tree training

In supervised learning a training point usually appear as a pair of data $(\mathbf{x}, y)$, where $\mathbf{x}$ is the input feature vector, and y represents the label. Given a set of training data $\mathcal{X}$. The trees are randomised, by randomly selecting a subset of feature at each internal node. The decision function at the internal node is used to decide whether the data $\mathbf{X}_i$ reaching that node should be assigned to its left or right child node. That is, at node m the training set $\mathcal{X}_m$ is split into $\mathcal{X}_m^L$ and $\mathcal{X}_m^R$ according to the results of test function $h(\mathbf{x}, \alpha_m)$. The split parameters α_m of the test function at node m is selected as result of the maximisation of the information gain which produce the highest confidence in the final distributions:

$$\alpha_m^* = \arg\max_{\alpha_m} I_m \tag{6.1}$$

with energy model,

$$\begin{aligned} I_m &= I(\mathcal{X}_m, \mathcal{X}_m^L, \mathcal{X}_m^R, \alpha_m) \\ \mathcal{X}_m^L &= \{(\mathbf{x}, y) \in \mathcal{X}_m \,|h(\mathbf{x}, \alpha_m) = 0\,\} \\ \mathcal{X}_m^R &= \{(\mathbf{x}, y) \in \mathcal{X}_m \,|h(\mathbf{x}, \alpha_m) = 1\,\} \end{aligned} \tag{6.2}$$

During the training process, starting from the root node $m = 0$, the data are split and sent to left or right child node by finding the optimal split parameters according to the objective function defined in 6.1. Each child node receives different subset of the training set, with $\mathcal{X}_m = \mathcal{X}_m^L \bigcup \mathcal{X}_m^R$ and $\mathcal{X}_m^L \bigcap \mathcal{X}_m^R = \varnothing$. The tree is constructed following this procedure with randomly selected features $\mathbf{x}$ at each internal nodes until the data arrives at a leaf. However, it is unnecessary to grow a tree till each data has been occupied its own leaf node, which would lead expensive computation, difficult interpretation and will result in over-fitting. Stopping criteria would affect tree structure and it needs to be applied in order to get the optimal structure. It is common to stop

growing the tree if the number of samples at a node is too small or the depth of tree exceeds the pre-defined limit.

Equation 6.2 is a general case of energy model. The frequent choose is to maximise the information gain as,

$$I_m = H\left(\mathcal{X}_m\right) - \sum_{i \in \{L,R\}} \frac{\left|\mathcal{X}_m^i\right|}{\left|\mathcal{X}_m\right|} H\left(\mathcal{X}_m^i\right) \tag{6.3}$$

$H\left(\centerdot\right)$ represents the Shannon entropy for discrete probability distributions and indicates a cardinality for the dataset. The entropy is defined as,

$$H\left(\mathcal{X}_m\right) = -\sum_{c \in \mathcal{C}} p(c) \log(p(c)) \tag{6.4}$$

where $\mathcal{C}$ represents a set of all classes and is the probability function of class c. c indicates the class label.

In the case of continuous probability distributions, $H\left(\centerdot\right)$ represents differential entropy which is an extension of Shannon entropy,

$$H\left(\mathcal{X}_m\right) = -\int_{y \in \mathcal{Y}} p(y) \log p(y) dy \tag{6.5}$$

where $\mathcal{Y}$ contains all the continuous labels, and $p(\cdot)$ is the probability density function.

Tree testing

The tree testing is rather simple. A previously unseen data can be sent to the left or right child node depending on the result of the testing function $h\left(\cdot, \cdot\right)$ until it arrives to a leaf. After training, the samples are assigned to each of the leaf node. Since the data splitting are applied at every internal nodes based on the features, intuitively the samples who reach the same leaf contain similar attributes. The new testing data is

more likely to end up in a leaf which has similar training samples. Each leaf node produces the posterior distributions, as $p(y\,|\mathbf{x})$. The tree predictor can be obtained by using maximum a posteriori probability estimate as $y^* = \arg\max_y p(y\,|\mathbf{x})$.

Limitations

Although random decision trees have various advantages and were proven to be useful, several limitations still remain in their applications. One significant problems in decision trees is over-fitting, that is the learners may create over complex trees which do not generalise well to new samples. A common strategy is to remove sections of tree that provide little information of the data which may only cause by noise. However, it does not solve it completely. Furthermore, single learner is also not suitable for high dimensional data.

6.1.2 Ensemble trees

Ensemble learning technique aims to construct a set of weak classifiers and combines them to create a strong classifier. In contrast to many single classifier models where only one hypothesis is learned from the training data, the ensemble methods try to build multiple learners solving the same problem. Since each single model in an ensemble has their limitations, the ensemble learning can manage the strengths and weaknesses, producing a better overall accuracy.

Ensembles of trees also called random decision forests which combine the idea of decision trees and ensemble learning methods. A random decision forest is an ensemble of such decision trees. The trees are trained independently from each other. Once the random forest has been trained, the new sample can be simply put through all trees. During the testing, each tree yields its own hypothesis. Evaluating the prediction of an ensemble is typically combines all tree predictions by simply averaging all the

distributions produced by each tree [20]. For example, a forest consists of total T number of trees, we denote the posterior distribution of t^{th} tree as $p_t(y\,|\mathbf{x}), t \in \{1 \dots T\}$, then the prediction model is,

$$p(y\,|\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} p_t(y\,|\mathbf{x}) \tag{6.6}$$

Figure 6.1 illustrates a synthetic multi-class classification example. We investigate the effect of the forest size (number of trees in a forest), one of the most influential parameters of a forest. Three-class spiral data are generated as training set. The data contain two dimensions, where each dimension represents a feature (Figure 6.1a). Figure 6.1 b-d show the testing classification posterior of all the points in feature space with varying number of trees ($T = 1, 10, 100$) using in the training. All the experiments were run with tree depth $D = 6$, and used a general oriented weak learner model [33]. The colour are obtained from the combination of three solid colours (red, green and blue) representing the uncertainty of classes. e.g. highly mixed colour corresponds low predictive confidence of the points in this region. According to the visualised results, using only a single tree produces undesirable, over confident prediction results. Increasing the number of trees in the forest can help to get much smoother posteriors. The results have shown that the accuracy of an ensemble trees can significantly exceed the single tree model.

6.2 Density forests

The problem is closely related to data clustering. Although significant amount of research have been done on forest-based data clustering, we followed the work in [33], where it is proposed to use an unsupervised information gain based optimisation.

Given a set of observed data without training labels $\mathcal{X} = \{\mathbf{X}_1 \dots \mathbf{X}_M\}$, the in-

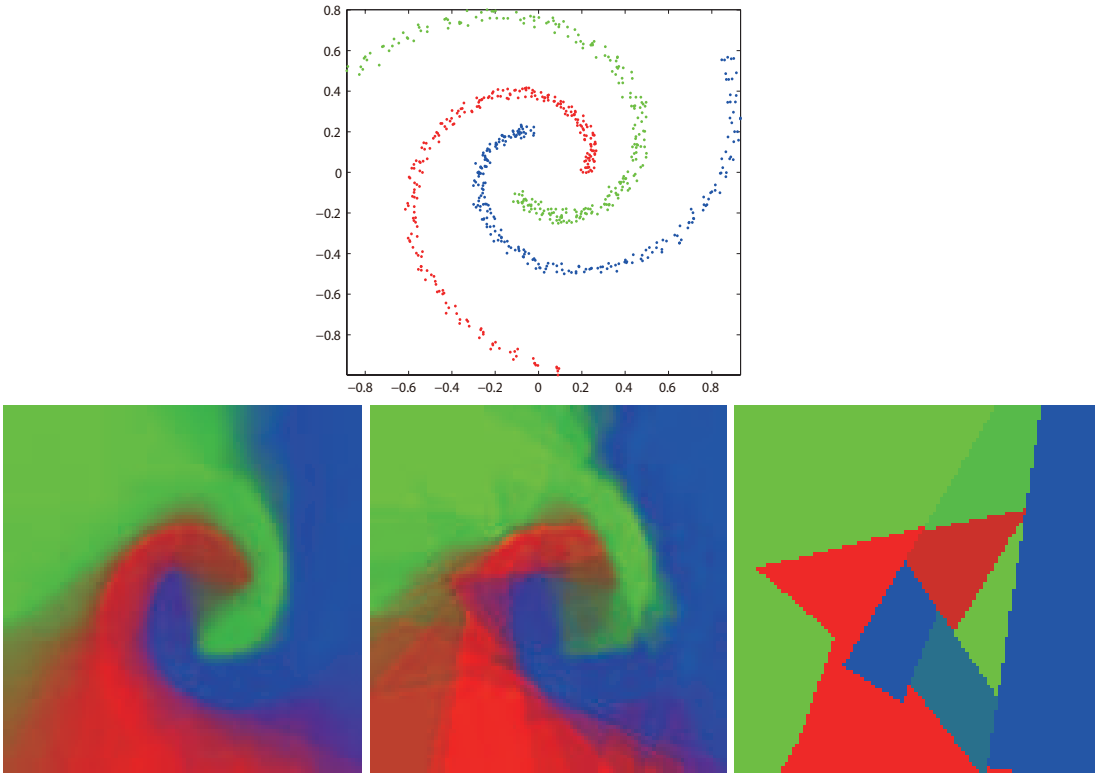


Figure 6.1: A classification example evaluate the performance of random forests with varying tree numbers. 1^{th} row: Input three-class labelled spiral data. 2^{nd} row: The testing posterior produced by random forest consists of $T = 100, 10, 1$ individual trees respectively.

dividual trees are trained independently in parallel. The optimal parameters at m^{th} internal node are obtained by maximising the information gain (see Equation 6.1), with the generic information gain defined as in Equation 6.3. Since the training labels are not provided, unsupervised entropy is defined as the differential entropy of a d-variate Gaussian distribution,

$$H\left(\mathcal{X}_m\right) = \frac{1}{2}\log\left((2\pi e)^d \left|\Lambda(\mathcal{X}_m)\right|\right) \tag{6.7}$$

where $\Lambda(\mathcal{X}_m)$ is the covariance matrix of $\mathcal{X}$ with size $d \times d$. Substitute 6.7 into 6.3,

the information gain can be rewritten as,

$$I_m = \log\left(|\Lambda(\mathcal{X}_m)|\right) - \sum_{i\in\{L,R\}} \frac{|\mathcal{X}_m^i|}{|\mathcal{X}_m|} H\left(\left|\Lambda(\mathcal{X}_m^i)\right|\right) \tag{6.8}$$

Once the training data has reached the leaf, the output of the testing data $\mathbf{x}$ in the t^{th} tree is represented by a multi-variate Gaussian distribution $\mathcal{N}(\cdot)$,

$$p_t(\mathbf{x}) = \frac{\pi_{l(\mathbf{x})}}{Z_t}\mathcal{N}(\mathbf{x}, \mu_{l(\mathbf{x})}, \Lambda_{l(\mathbf{x})}) \tag{6.9}$$

$l(\mathbf{x})$ denotes the leaf reached by the testing data $\mathbf{x}$. μ_l and Λ_l are the mean and associated covariance matrix of all points reaching the leaf l. $\pi_{l(\mathbf{x})}$ is the scaling vector indicating the proportion of all training points reaching the leaf l. Z_t is seen as the partition function providing probabilistic normalisation [33].

The forest density is given by the average of all tree densities in the ensemble model,

$$p(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} p_t(\mathbf{x}) \tag{6.10}$$

Figure 6.2b-d illustrates the output of density forests trained on the input data - the shape of a three-arm spiral in Figure 6.2a, for varying numbers of trees ($T = 1, 10, 100$) and tree depth ($D = 4, 6, 10$). Bright pixels represent high density values and dark pixels represent low density values. As observed in the figure, deeper trees ($D = 10$) may lead to over-fitting. This is particularly true when only few trees are used in a forest. However due to the randomness of each trees (trees are independent with respect to each other), increasing forest size T helps to produce smooth densities, thus avoid over-fitting problem and greatly improve the results. On the other hand, since the distribution of input data is rather complex in this example, under-fitting problem may be caused by a smaller D.

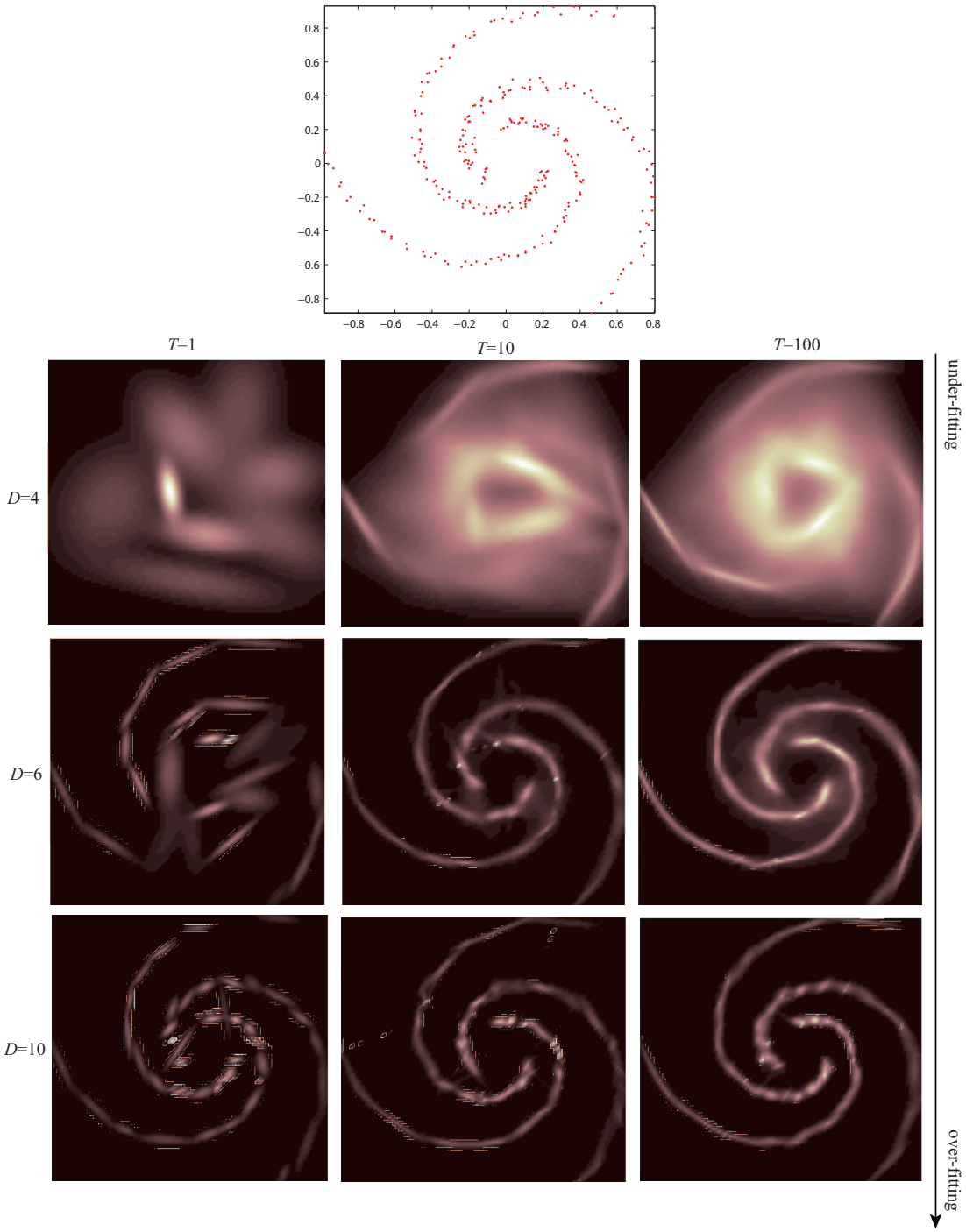


Figure 6.2: A density forest example with varying tree numbers T and tree depths D. 1^{st} row: A three-arm unlabelled spiral data. 2^{nd} to 4^{th} row Forest densities for different T and D.

6.3 Forest model for manifold learning

Manifold learning, as introduced in Section 5.2, aims to find smooth mapping, such that $\Psi : \mathbf{X} \rightarrow \Psi(\mathbf{X})$, where $\mathbf{X} \in \mathbb{R}^n, n \ll N$, while preserving local geometry of the dataset $\mathcal{X}$ with $\mathbf{X} \in \mathcal{X}$.

In Chapter 5 we introduced the idea of using diffusion maps technique in dimensionality reduction problem. Now the manifold forests can be constructed upon diffusion maps [29] with the neighbourhood topology learned through random forest data clustering. It generates efficient representations of complex geometric structures even when the observed samples are non-uniformly distributed. The diffusion map is a graph-based non-linear technique with quasi-isometric mapping from original shape space onto a lower dimensional diffusion space.

Manifold forests are closely related to density forests, but with extra steps on building affinity matrix and estimating the mapping function $\Psi(\cdot)$. Details are provided next.

6.3.1 The affinity model

In the proposed method, the affinity model in manifold learning is built by applying random forest clustering. Let $\mathcal{X} = \{\mathbf{X}_1 \dots \mathbf{X}_M\}$ be a dataset with M training samples, the data partition is defined based on the leaf node $l(\centerdot)$ that the input data $\mathbf{X}_i$ would reach. The entries of the affinity matrix $\mathbf{Y}^t$ for tree t are calculated as,

$$W_{ij}^t = e^{-L^t(\mathbf{X}_i, \mathbf{X}_j)}, i, j \in 1 \dots M \tag{6.11}$$

where the distance L is obtained using different affinity models. The most commonly accepted one is the use of Gaussian kernel, where the affinity model is defined as,

$$L^t(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} \frac{\|\mathbf{X}_i - \mathbf{X}_j\|^2}{\delta} & l(\mathbf{X}_i) = l(\mathbf{X}_j) \\ \infty & \text{otherwise} \end{cases} \tag{6.12}$$

The length parameter δ is chosen to be the average smallest non-zero value of $\|\mathbf{X}_i - \mathbf{X}_j\|^2$. Applying binary model is another option. As a special case of Gaussian model with $\delta \to \infty$, building binary affinity is simpler and can be considered to be a parameter-free.

$$L^t(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} 0 & l(\mathbf{X}_i) = l(\mathbf{X}_j) \\ \infty & \text{otherwise} \end{cases} \tag{6.13}$$

This is a parameter-free model that the distance between a pair of points is zero if they end up in the same leaf, otherwise set the distance as infinity. However, as affinity matrix calculated based on a single tree is not representative, the ensemble of T trees is used to get an overall affinity matrix $\mathbf{Y}$ by averaging over all affinity matrices from each single tree: $\mathbf{Y} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{Y}^t$.

6.3.2 Estimating the mapping function

Coifman et al. presented a justification behind using normalised graph Laplacian [29] by connecting them to diffusion distance. Each entry of the diffusion operator $\mathbf{P}$ is constructed as $P_{ij} = \hat{Y}_{ij}/d_{ii}$ with $d_{ii} = \sum_j \hat{Y}_{ij}$. $\hat{\mathbf{Y}}$ is a renormalised affinity matrix of $\mathbf{Y}$ using an anisotropic normalised graph Laplacian, such that $\hat{Y}_{ij} = Y_{ij}/q_i q_j$ with $q_i = \sum_j Y_{ij},\ q_j = \sum_i Y_{ji}$. The convergence of optimal embedding Ψ for diffusion maps is proven in [29] and is found via eigenvectors φ and its corresponding n biggest

eigenvalues λ of the operator $\mathbf{P}$, such that $1 = \lambda_0 > \lambda_1 \geq \ldots \geq \lambda_n$,

$$\Psi : \mathbf{X}_i \mapsto [\lambda_1 \varphi_1(\mathbf{X}_i), \cdots, \lambda_n \varphi_n(\mathbf{X}_i)]^T \tag{6.14}$$

The detail of using diffusion maps has been presented in section 5.2.3. Figure 6.3 is an example of embedding of original 2D spiral data to 1 dimensional real line with colour coded. The figure shows that manifold forest capture correctly the intrinsic 1D manifold. The plots in Figure 6.4 shows the 3D parabola surface $f(x, y) = \frac{x^2+y^2}{\phi}$, with $\phi = 2$ (same as in Figure 5.3a) and the mapping into the 2D plane using binary and Gaussian affinity models described above. Although the shape in reduced space can reflect the original shape better when the similarity measure is calculated in terms of the Euclidean distance for the data ending up in the same leaf, define the distance using binary one can greatly improve the computation speed, especially for the data in a very high dimensional space.

Compare the embedding using Gaussian affinity with Figure 5.3b, the embedding obtained from the manifold forests achieves better distribution in the reduced space than only using diffusion maps.

Figure 6.5 shows the embedding of shape from cardboard data, together with representative corresponding shapes extracted from 1000 training samples. The illustration of the embedding results obtained by applying manifold forests seems more evenly distributed than applying diffusion maps shown in 5.4, especially for the points belong to the border of the manifold.

6.4 Random forests in deformable shape reconstruction

Once the manifold has been build from the training dataset, the reconstruction can be processed following the steps described in section 5.4. Brief descriptions are provided

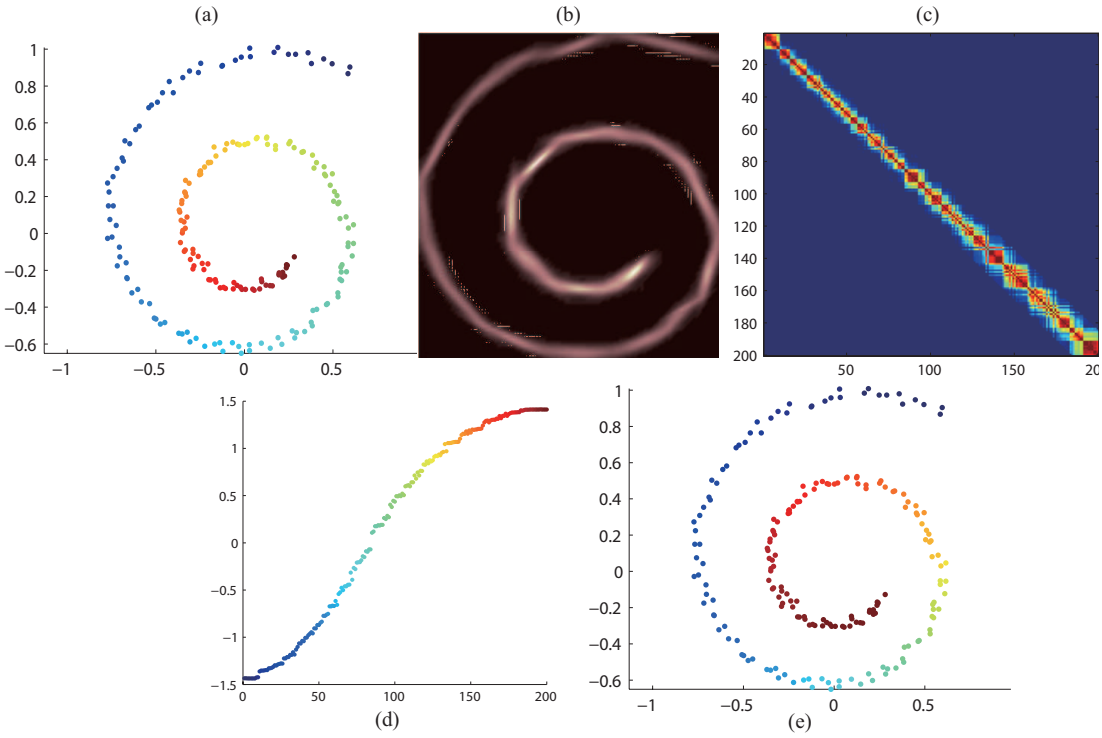


Figure 6.3: (a) Original 2D spiral data. (b) Forest density, computed with parameters $T = 100$, $D = 6$. (c) Ensemble model of affinity matrix. (d) Manifold forest mapping data from the original 2D space to the 1D real line is colour coded. (e) Embedding data in 1D.

next. More details and mathematical justification can be found in section 5.4.

Initialisation

Initial shapes and camera motion are estimated by running a few iteration of the optimisation process using the linear method described in Chapter 3. Our method is not significantly sensitive to the initial solution as the method can iteratively update the shapes by projecting them on the learned manifold until convergence.

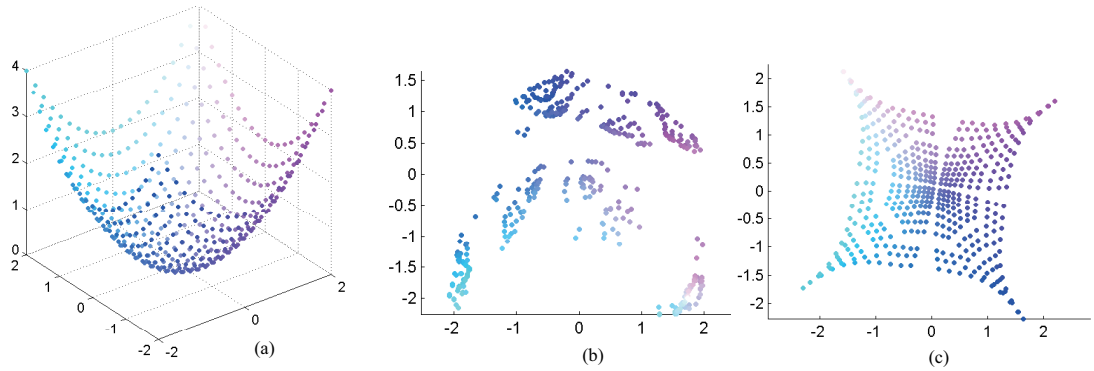


Figure 6.4: Manifold forest and non-linear dimensionality reduction. (a): Input 3D parabola surface. (b): Non-linear mapping from the original 3D space to the 2D reduced space based on binary affinity model. (c) Embedding based on Gaussian affinity model.

Mapping out-of-sample points

The manifold forests method briefly described in section 6.3 is used to find a meaningful representation of the data, but the mapping Ψ is only able to provide an embedding for the data present in the given training set. Suppose a new shape $\mathbf{S}_t \in \mathbb{R}^N$ becomes available after the manifold had been learned, instead of re-learning the manifold which is computationally expensive, an efficient way is to interpolate the shape onto the lower dimensional feature space. For each new shape, such embedding is calculated based on the Nyström extension [6],

Inverse mapping

Given a point $\mathbf{b} \in \mathbb{R}^n$ in the reduced space, finding its inverse mapping $\mathbf{S}_t = \Psi^{-1}(\mathbf{b})$ from the feature space back to the input space is a typical pre-image problem. As claimed in [6], the exact pre-image might not exist if the shape $\mathbf{S}_t$ has not been seen in the training set. However, according to the properties of isometric mapping, if the points in the reduced space are relatively close, the corresponding shapes in high dimensional space should represent similar shapes since they have small diffusion distances.

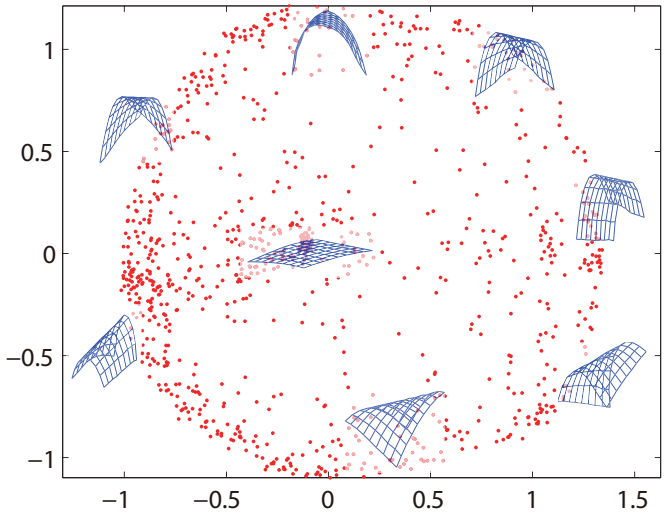


Figure 6.5: reduced space obtained from manifold forest of cardboard dataset

Based on this, the point $\mathbf{b}_t$ can be approximated as a linear combination of its weighted neighbouring points in feature space, such that $\mathbf{b}_t = \sum_{l=1}^{n+1} \theta_{tl}\mathbf{x}_{tl}$, where $\mathbf{x}_{tl}$ is the l^{th} nearest point of $\mathbf{b}_t$ and the weights θ_{tl} are computed as the barycentric coordinates of $\mathbf{b}_t$. Once the weights are estimated, the shape $\mathbf{S}_t$ can be calculated as well based on a set of weighted training samples $\mathbf{S}_t = \sum_{l=1}^{n+1} \theta_{tl}\mathbf{X}_{tl}$, where the training samples $\mathbf{X}_{tl}$ are the pre-images of $\mathbf{x}_{tl}$, and are equivalent to the basis shapes in Equation5.6.

Non-linear refinement

The cost function is given as,

$$\underset{\mathbf{R}_t,\theta_{tl}}{\arg\min} \sum_{t=1}^{F} \|\mathbf{Y}_t - \mathbf{R}_t \cdot \mathbf{S}_t\|^2 + \varphi_{\mathbf{S}} \sum_{t=2}^{F} \|\mathbf{S}_t - \mathbf{S}_{t-1}\|^2 + \varphi_{\mathbf{R}} \sum_{t=1}^{F} \varepsilon_{rot} \tag{6.15}$$

where $\varepsilon_{rot} = \left\|\mathbf{R}_t \cdot {\mathbf{R}_t}^T - \mathbf{I}\right\|^2$ enforces orthonomality of all $\mathbf{R}_t$. $\varphi_{\mathbf{S}}$ and $\varphi_{\mathbf{R}}$ are regularisation constants.

However, the underlying problem is that the quality of the optimisation result strongly depends on the accuracy of initial shapes. To avoid this, we update the basis

shapes in each iteration until 2D measurement error is less than the defined threshold (10^{-3} in our case) and the error between two adjacent frames is relatively small.

6.5 Experiments on improved method I

A number of experiments were carried out to evaluate the proposed method. We compare the proposed random forest method (denoted as **RF**) with several state-of-the-art algorithms these experiments. The algorithms and testing sequences used for the comparison have been introduced in section 5.5.

6.5.1 Quantitative evaluation

Different number of bases n

The accuracy of reconstruction is affected by the dimensionality of the reduced space n, corresponding to number of shape basis. The first test looked at the relation between manifold dimensionality and the shape reconstruction error. All sequences were separated into 2 groups: facial sequences (*Surprise*, *Talking*) and surface sequences (*Cardboard*, *Cloth*). The forests have been trained with the average 600 number of trees. The results in Figure 6.6 show that with increasing dimension of the reduced space n the shape reconstruction error is reduced. As expected, a higher number of bases is required to describe a complex shape deformation, e.g. surface sequences.

6.5.2 Qualitative Evaluation

Motion capture data

Table 6.1 shows the 3D reconstruction error for RF, DM, IPCA and KSFM which on average provide better results than other trajectory based methods. The relative normalised means of the 3D error are compared over all frames and all points. For

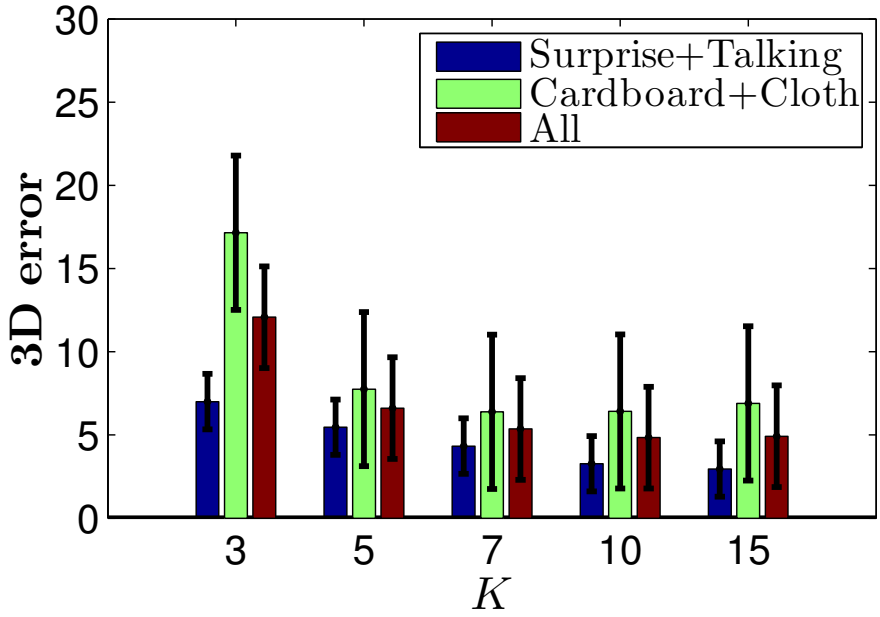


Figure 6.6: Average normalised mean 3D error and standard deviation of different number of dimensions in reduced space. Bars left to right: Group of facial sequences, group of surface sequences, , all the sequences.

RF method the initialisation error and the error produced by the proposed algorithm with and without non-linear refinement are presented. The errors shown in the table correspond to the optimal n value selection. This is achieved by running the trials with n varying from 2 to 15. The best selected n value for each tested method is shown in brackets. The reconstructed shapes are aligned using a single global rotation based on Procrustes alignment. As shown in the table, RF has better performance than other methods, especially for the large deformations. Even though the initial error is big, the RF method is still able to provide accurate reconstruction results.

Real data

The algorithms used in the motion capture experiments above are applied to real data in Figure 6.12. In the video, 81 point features were tracked along 61 frames showing approximately two periods of paper bending movement.

	DM	RF		
		Initial	No Opt.	Opt.
Surprise	0.0352(10)	0.3154	0.2929	**0.0241**(15)
Talking	0.035(10)	0.9657	0.0837	**0.0343**(10)
Cardboard	0.1064(10)	0.2674	0.1606	**0.0940**(10)
Cloth	0.0287(7)	0.2967	0.1729	**0.0254**(7)

Table 6.1: Relative normalised mean reconstruction 3D error for DM and RF methods. The optimal number of bases n, for which the 3D errors are shown in the table, is given in brackets for each tested method

Improved method II

The second part of this chapter presents a method for recovering deformable shape and motion from uncalibrated 2D video sequence in the presence of missing data. Highly deformable shapes are hard to describe under previously used assumptions, such as global constraint enforcing shapes to lie within a linear subspace. Considering that the data dimensionality may not represent the true complexity of the problem, we suggest that the shapes can be well-modelled in a low dimensional manifold. However, building a dense representation of the manifold requires a large amount of training data which is not feasible in many real applications. The main contribution of this novel approach is to accurately estimate 3D reconstructions utilising manifold learned from a relatively small number of training samples. The problem is addressed by grouping shapes into evolving clusters, with the shapes in each cluster represented in the linear subspace, estimated based on the observations and the prior learned manifold. Results are presented using motion capture data and real video sequences, showing that the proposed method can better model shapes with complex deformations compare to several state-of-the-art techniques, and is robust against noise and missing data.

Novelty

The main contribution of this part is a novel approach for recovery of 3D non-rigid structures with large and/or complex deformations. The proposed method is shown to be flexible allowing a method extension to handle the case with missing measurements e.g. due to occlusion or feature track loss. The proposed method is based on a recently introduced manifold learning technique, Diffusion maps. As claimed in Chapter 5, building a dense representation of the manifold enables to achieve better reconstruction performance when compared to other state-of-the-art approaches, but collecting sufficient number of training data may not be feasible in practice. The algorithm described in this section is an improved version of the original diffusion maps based algorithm proposed in Chapter 5, with three main differences. First, the improved algorithm enables reconstruction with small number of training samples. Second, the proposed cost function includes additional term to relax the constraint on local basis shapes. Unlike previous method in Chapter 5 these shapes do not have to match the local training samples. Third, the proposed algorithm has additional step solving the missing data problem.

6.6 Methodology

The method presented in Chapter 5 introduced the non-linear manifold, learned based on 3D training samples, as shape prior for non-rigid shape reconstruction. Given the learned shape manifold and the observed 2D measurements, the algorithm iteratively refines the 3D reconstructed shapes for each frame by using its $n + 1$ nearest shape neighbours on the manifold, as basis shapes. Although the method is able to achieve high quality shape reconstructions, the requirement of large number of training data to build a sufficiently dense representation of the manifold is not feasible for most real applications. To overcome this, the method proposed in this paper relaxes the

constraint for basis shapes so as to make the algorithm more adaptable to the case when only a relatively small number of training samples have been used for the manifold learning.

6.6.1 Shape clustering

Given a set of estimated shapes $\mathcal{S} = \{\mathbf{S}_1, \dots \mathbf{S}_F\}$, the aim of the clustering is to partition F shapes into K clusters, in which the shapes have similar structure, with each shape cluster denoted by $\mathcal{T}_i, i \in 1 \dots K$. The clusters are obtained by performing the Delaunay triangulation in the reduced space. As defined in [15], any "angle-optimal" triangulation of a set of points is a Delaunay triangulation of these points. This can help to avoid "skinny triangles", for which the corresponding shape of each vertex could be significantly different, thus may lead to meaningless reconstructions.

Diffusion maps are based on distance preserving mapping, meaning that the points relatively close in reduced space correspond to the similar shapes. As a consequence we stipulate that the points in the reduced space belong to the same Delaunay simplex (i.e. cluster), can be modelled by the same linear subspace embedded in $\mathbb{R}^N$, and therefore all corresponding reconstructed shapes (represented by that cluster) can be approximated by a linear combination of the same set of unknown but fixed basis shapes. Thus all the shapes in the cluster i can be represented as $\mathbf{S}_t = \sum_{l=1}^{n+1} \theta_{tl} \mathbf{B}_l^i, \forall t \in \mathcal{T}_i$, where a set of basis shapes $\mathcal{B}^i = \left\{\mathbf{B}_1^i \dots \mathbf{B}_{n+1}^i\right\}$ is spanning the tangent linear subspace representing all the shapes from the cluster i.

The reconstructed shapes are often different from the training samples, therefore cannot be perfectly mapped into the manifold $\mathcal{M}$. As the result we relax the constraint for the basis shapes, only "encouraging" them to be close to the basis shapes spanning the tangent subspace, instead of being exactly the same. The additional constraint

applied to the i^{th} set of basis shapes is,

$$\varepsilon_{bs}^{i} = \sum_{l=1}^{n+1} \left\| \mathbf{B}_{l}^{i} - \mathbf{X}_{l}^{i} \right\|^{2}, \mathbf{X}_{l}^{i} \in \mathcal{X} \tag{6.16}$$

Figure 6.7 illustrates an example of how the initial shapes are redistributed in the reduced space after algorithm has converged. As shown in (a) the initial shapes are embedded in a two dimensional space which fall into three clusters, $K = 3$. (b) shows the embedding of optimal shapes which produced by the non-linear optimisation (see Section 6.6.2) with $K = 11$.

This approach differs from the original diffusion maps based method which was presented in Chapter 5 as all the shapes belonging to the same cluster are being jointly optimised, whereas in original one all the shapes would have been reconstructed independently if not for the temporal smoothness constraint(not used in the algorithm proposed in this paper). Additionally the proposed algorithm relaxes the constraint on the tangent subspace as it only encourages that the basis shapes to be "close" to this subspace.

6.6.2 Non-linear refinement

The parameters $\theta_{tl}, \mathbf{B}_{l}^{i}$ and $\mathbf{R}_{t}$ are optimised simultaneously by minimising the 2D re-projection error with additional constraints on basis shapes and rotation matrices. The cost function can be written as,

$$E(\mathbf{R}_{t}, \mathbf{B}_{l}^{i}, \theta_{tl}) = \sum_{t \in \mathcal{T}_{i}} \left\| \mathbf{W}_{t} - \mathbf{R}_{t} \sum_{l=1}^{n+1} \theta_{tl} \mathbf{B}_{l}^{i} \right\|^{2} + \lambda_{B} \varepsilon_{bs}^{i} + \lambda_{R} \sum_{t \in \mathcal{T}_{i}} \varepsilon_{rot} \tag{6.17}$$

where $\varepsilon_{rot} = \left\| \mathbf{R}_{t} \mathbf{R}_{t}^{T} - \mathbb{I} \right\|$ enforces orthonomality of all $\mathbf{R}_{t}$. The parameters λ_{B} and λ_{R} are regularisation constants selected experimentally. A non-linear optimisation based

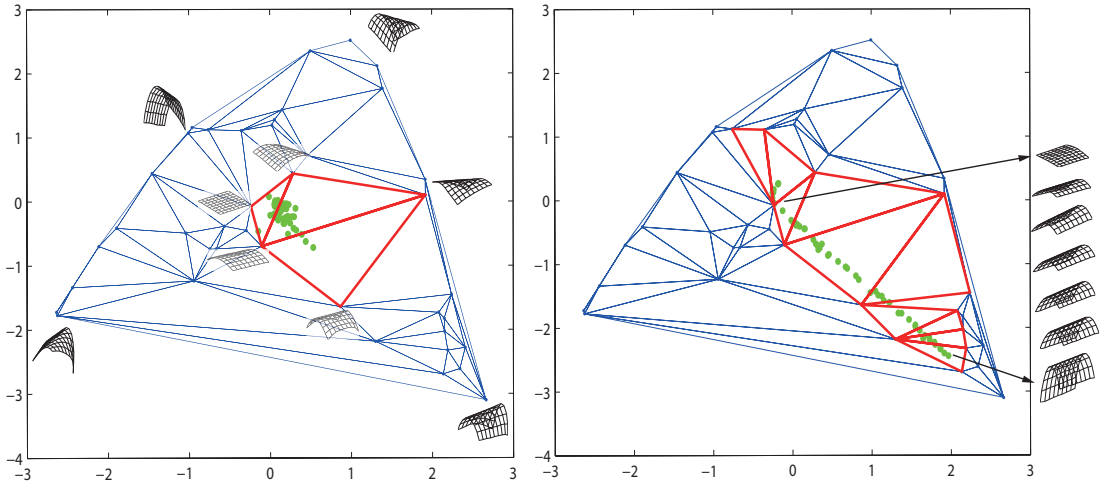


Figure 6.7: Delaunay triangulations (blue line) in the reduced space; Left: Embedded initial shapes (green dots) in a reduced space and the actual used triangles (red line), together with representative corresponding shapes from the total of 40 training samples; Right: Embedded reconstructed shapes (green dots) in a 2D reduced space and the actual used triangles (red line), with some reconstructed shapes

on bundle adjustment using Levenberg-Marquardt algorithm was applied to minimize this cost function.

As usual the quality of the provided initial shapes may seriously affect the results of the optimisation, we try to avoid this by updating the basis shapes $\mathcal{B}^i$(re-cluster the data) and the corresponding shape coefficients in each iteration until 2D measurement error is less than the defined threshold (10^{-3} in this case) and the error between two adjacent frames is relatively small. The pre-image of the vertices of Delaunay triangles are used to constraint the basis shapes, Figure 6.7 shows which Delaunay simplexes are being used along the iterations. The algorithm for iteratively 3D shape estimation is summarised in Algorithm 7.

6.6.3 Reconstruction with missing data

The algorithm described above assumes the measurements $\mathbf{W}$ are complete, all the feature points are identified in all the images in the sequence. In practice, some of the

Algorithm 7 Iteratively 3D shape estimation

Input: 2D points with known correspondence, diffusion map calculated from the training dataset $\mathcal{X}$.

1: Initialisation: Obtain initial shapes $\mathbf{S}'$ and camera motion $\mathbf{R}'$. for each frame t.
2: **repeat**
3: Compute the embedding $\hat{\Psi}$ of new shapes $\mathbf{S}_t \mapsto \hat{\Psi}(\mathbf{S}_t)$
4: Find $n+1$ nearest neighbours $\mathbf{x}_{tl}$ and its corresponding training samples $\mathbf{X}_{tl}$ of the embedded point $\mathbf{b}_t$
5: Calculate the barycentric coordinates θ_{tl} of $\mathbf{b}_t$
6: Perform clustering $\mathcal{T}_i$ of the estimated shapes $\mathcal{S}$
7: Refine $\theta_{tl}, \mathbf{B}_l^i, \mathbf{R}_t$ as to the cost function Equation 6.17
8: Update the reconstructed 3D shapes $\mathbf{S'}_t = \sum_{l=1}^{n+1} \theta_{tl} \mathbf{B}_l^i$
9: Set $\mathbf{S}_t = \mathbf{S'}_t$
10: **until** $(\|r\| > r_T)$ *and* $\left(\|r_t\| - \|r_{t-1}\| > 10^{-3}\right)$

Output: 3D reconstructed shapes $\mathcal{S}$ and camera motion $\mathcal{R}$.

points cannot be detected in all the images due to the occlusions, feature detection problems, or tracking failures and therefore acquiring complete set of measurements is unlikely. We present two methods which efficiently handle the case of missing data in the shape estimation problem.

Linear approach

If the input data is incomplete, instead of considering more complex and time-consuming optimisation algorithms, we briefly summarise a recently proposed linear method based on Principal Component Analysis (PCA) presented in section 3.7, with the missing data recovered before estimating the shapes and motion.

Assuming p feature points lie on the surface of an object, we set $\mathbb{I} = \bar{\Pi}_t + \bar{\Pi}_t^*$, where $\mathbb{I}$ is the identity matrix and $\bar{\Pi}_t$ is a $p \times p$ diagonal matrix such that $\bar{\Pi}_t(k,k) = 0$ indicates that the point k is missing in image t, otherwise $\bar{\Pi}_t(k,k) = 1$. The observations of time t can be represented as $\hat{\mathbf{W}}_t = \mathbf{W}_t \Pi_t$ and the missing measurements as $\hat{\mathbf{W}}_t^* = \mathbf{W}_t \Pi_t^*$, where matrix Π_t and Π_t^* are obtained from $\bar{\Pi}_t$ and $\bar{\Pi}_t^*$ by removing all columns for which entries are all zeros. According to Equation 2.17, measurements can be factorised using

motion $\mathbf{M}$ and shape bases $\mathbf{B}$ matrices, the incomplete measurement can be written as: $\hat{\mathbf{W}}_t = \mathbf{M}_t\mathbf{B}\Pi_t$.

We firstly compute the motion matrix $\mathbf{M}_t$ using the available 2D measurements and the eigenshapes $\mathbf{E}$, approximating the unknown bases $\mathbf{B}$, obtained from the training dataset $\mathcal{X}$, $\mathbf{M}_t = \hat{\mathbf{W}}_t(\mathbf{E}\Pi_t)^\dagger$, where $(\cdot)^\dagger$ indicates Moore-Penrose pseudo-inverse. The missing entries can be calculated as $\hat{\mathbf{W}}_t^* = \mathbf{M}_t\mathbf{E}\Pi_t^*$. Thus the completed measurement matrix is,

$$\mathbf{W}_t = \hat{\mathbf{W}}_t\Pi_t^T + \hat{\mathbf{W}}_t^*\Pi_t^{*T} \tag{6.18}$$

Non-linear approach

Since PCA is a linear manifold, the linear method is only able to cope well with small deformations. Although the method is not suitable when the deformations are relatively large or complex, it still can be used for providing a good starting point for the optimisation using the non-linear approach. The diffusion maps based method can be easily extended to handle the case with missing data. To facilitate this, modification of the Eq. 6.17 is introduced where the cost function can be rewritten as $E(\mathbf{R}_t, \mathbf{B}_l^i, \theta_{tl}, \mathbf{W}_t\Pi_t^*)$. And therefore depends explicitly on the missing observations $\mathbf{W}_t\Pi_t^*$. As results the cost function in Equation 6.17 is simultaneously minimised with respect to rotation, shape basis, shape coefficients and the missing observations. It should be pointed out that we only optimise the missing entries in the observation not the whole 2D measurements $\mathbf{W}_t$.

6.7 Experiments on improved method II

We evaluate the performance of the proposed method on both motion capture and real data. To identify the original diffusion maps and the proposed one, we use **DM1** to

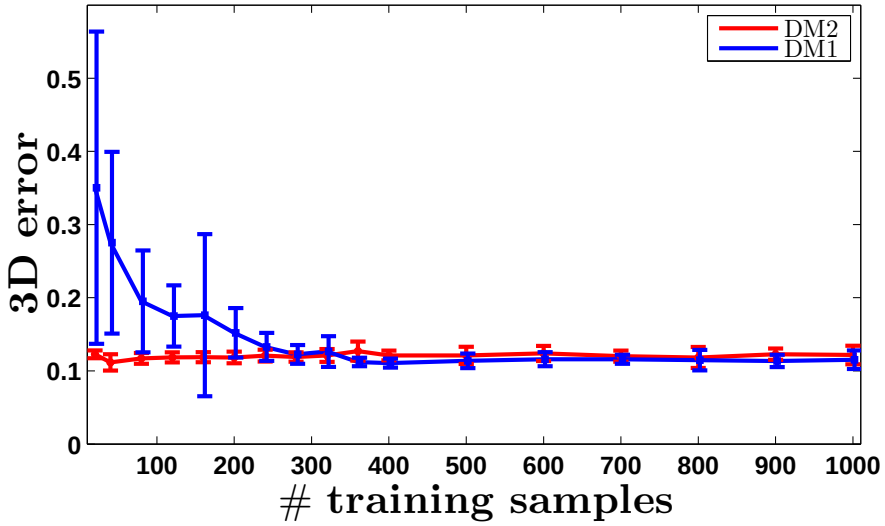
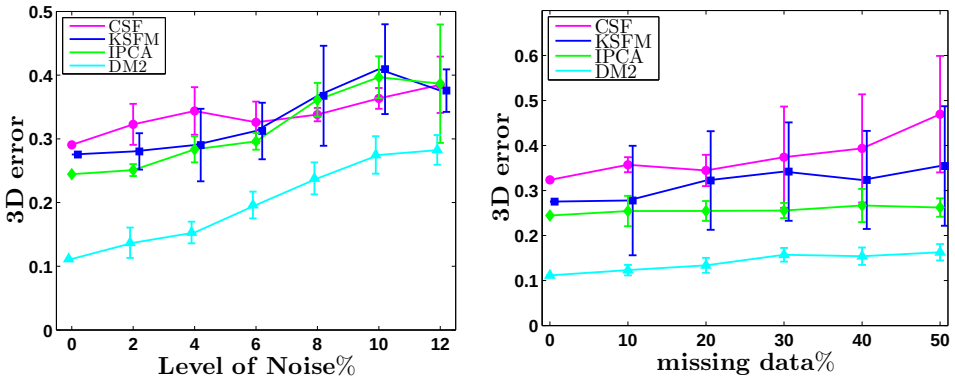


Figure 6.8: 3D error as function of the number of training samples for the *cardboard* data.

represent the original method without basis shape optimisation, requiring large amount of training data, and **DM2** represents the improved one.

6.7.1 Quantitative evaluation

As it was stipulated in the previous sections, only a small number of training samples are required by the proposed method. We firstly investigate the effect of the number of training shapes on the reconstruction accuracy. The average reconstruction errors with the standard deviation calculated over 10 trials (each using different data subset for training) are shown in Figure 6.8. It can be seen that although the two methods are comparable when over 400 training samples are used, DM2 is more stable and outperforms DM1 when relatively small shape sample is used for training. For the comparative evaluation, performance of the proposed method is tested against three previous approaches. The experiment is design to test the robustness of our approach when data is corrupted by noise. The measurements $\mathbf{W}$ were perturbed by Gaussian noise with varied level of noise. For each selected level of noise, the experiments were



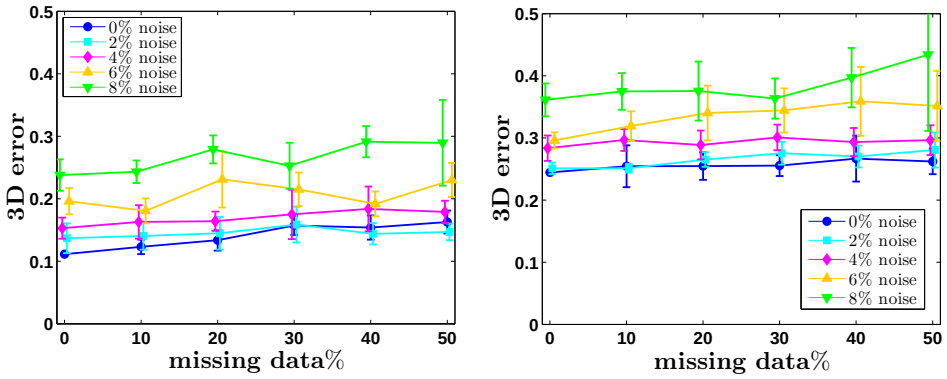
(a) (b)

Figure 6.9: (a) Reconstruction error as function of the measurement noise for the *cardboard* data. (b) The influence of the observations missing data on the reconstruction error.

repeated 10 times. The results in Figure 6.9(a) show our method provides smaller reconstruction errors.

To simulate the missing observations, we randomly discard 10%, 20%, 30%, 40% and 50% of the 2D entries in $\mathbf{W}$. The results in Figure 6.9(b) are calculated by averaging over 10 trials. With the missing data ratio of up to 50% , the average (maximum) 3D and 2D reconstruction errors were 0.1629 (0.1881) and 0.0032 (0.0053) respectively, where errors were calculated as $\|\mathbf{W} - \mathbf{W}'\|/\|\mathbf{W}\|$, where $\mathbf{W}'$ is the reconstructed measurement matrix.

In real cases, missing data and measurement noise are distorting the observations in the same time. The aim of the following experiment is to evaluate the methods' performance in such situations. We compare results of the 3D error obtained using the PCA based method to fill the missing entries in the measurement and then apply DM2, with the results obtained using the non-linear approach. Results plotted in Figure 6.10 show the reconstruction error as function of the amount of the missing data for different level of noise in the observations. As it can be seen that both methods are robust with respect to missing data, however, the non-linear method provides smaller errors both



(a) (b)

Figure 6.10: Reconstruction results for varying levels of missing data and 5 levels of noise for the *cardboard* data. (a) Results using non-linear method with DM2; (b) Results using linear method.

in terms of means and standard deviations.

6.7.2 Qualitative evaluation

Motion capture data

Table 6.2 shows the 3D reconstruction error for different methods on different sequences. For DM we present both initial error and final result produced by DM1 and DM2. The errors are chosen with the optimal number of basis n, with the optimal n selected based on running the trials with n varying from 2 to 10. As shown in the table, DM1 and DM2 consistently outperform other methods, especially for the sequences with large deformations. Even though the initial error is big, the proposed method is still able to provide accurate reconstruction results. DM1 and DM2 are comparable, but DM2 uses much less training data than DM1, e.g. for cardboard sequence, DM1 required a dense representation of the manifold, for which 1000 shapes have been used for training, while DM2 only used 40 shapes for training. More results comparing DM1 against other approaches can be found in Chapter 5.

In Figure 6.11, we visually compare the results of KSFM and DM2 against ground

	Initial	DM1	DM2
Surprise	0.3154	0.0352(10)	**0.0208**(10)
Talking	0.9657	0.0350(10)	**0.0280**(10)
Cardboard	0.2674	**0.1064**(10)	0.1114(10)
Cloth	0.2967	**0.0287**(7)	0.0556(5)

Table 6.2: Normalised mean 3D error (number of bases n) of reconstruction results using different methods.

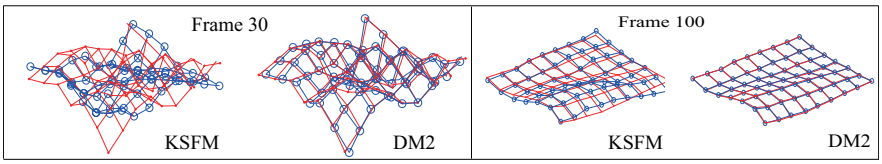


Figure 6.11: Reconstruction results on *cloth* sequence. Reconstructed 3D shapes (blue circles), together with ground truth (red dots) are displayed.

truth shapes. We can observe that DM2 generally gives better results, especially for the cloth sequence. This was to be expected since shapes can be better modelled in a non-linear manifold.

Real data

The algorithms used in the motion capture experiments above were applied to real data as shown in Figure 6.12. In the video, 81 features were tracked along 61 frames showing approximately two periods of paper bending movement.

6.8 Summary

In this chapter, two improved non-linear manifold methods have been proposed based upon our original diffusion maps method discussed in Chapter 5. Both methods perform well, when compared to other methods, especially for large and complex deformations. We firstly improved the method by building the non-linear manifold with random forests techniques which learned the neighbourhood topology from the data itself rather than

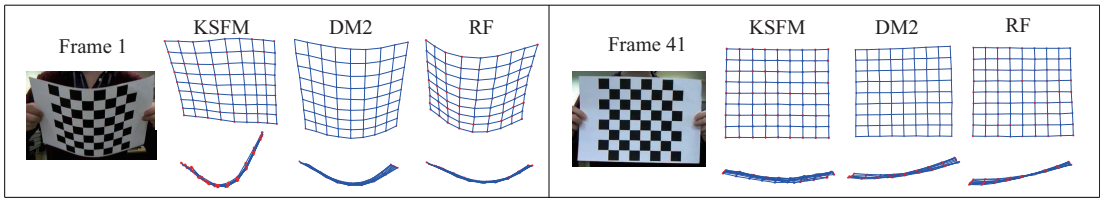


Figure 6.12: Selected 2D frames from the paper bending video sequence . Front and top views of the corresponding 3D reconstructed results using the proposed methods (DM2 and RF) and KSFM.

being defined by the Euclidean distance in standard diffusion maps method.

The second part of this chapter introduced another improvement which aims to produce accurate solution to the shape recovery problem by using much less training data. The advantage of the proposed method is that the non-linear manifold is only learned from small number of samples, and the reconstructed shapes are clustered into several local linear subspaces. By combining non-linear manifold technique and low-rank shape model, the method achieves better performance when compared with linear based methods. However the comparison of the proposed method with respect to the other methods may be seen as unfair, as better reconstruction accuracy of the proposed method comes at the cost of required availability of a representative training dataset.

It should be noticed that selection of the training shapes has not been optimised leading to some badly shaped triangles in the clustered reduced space. The reconstruction results are affected if corresponding shapes are being clustered in such triangles. Future work will attempt to address the problem by either refining the Delaunay mesh or introducing a criterion for selection of the optimal training shapes. We are also investigating several extensions of this work to more challenging cases, such as to deal with the outliers and real time implementation.

Chapter 7

Consideration of Practical Implementation

A number of approaches were proposed in the thesis to solve the problem of 3D non-rigid reconstruction. The proposed algorithms assume that the feature points have been detected in the images and the 2D correspondences are provided as input to the reconstruction algorithms; see Figure 7.1. The discourse to this point has focused on the algorithmic development of solutions to the reconstruction problem. The purpose of this chapter is to relate the practical concerns and issues associated with the implementation of the methods. This chapter describes several popular methods to detect and describe the local features in the images, as well as keypoint matching and video tracking, and therefore completes the description of the entire system. Recalling that the output of the reconstruction algorithm is a set of 3D points in each frame, to visualise the 3D objects in more realistic way, post processing including image-based rendering relies on both the original input images and the reconstructed structures.

7.1 Keypoint detection and matching

Feature detection is an essential component in many computer vision applications. Extracting feature points from images is usually performed as the first step in many

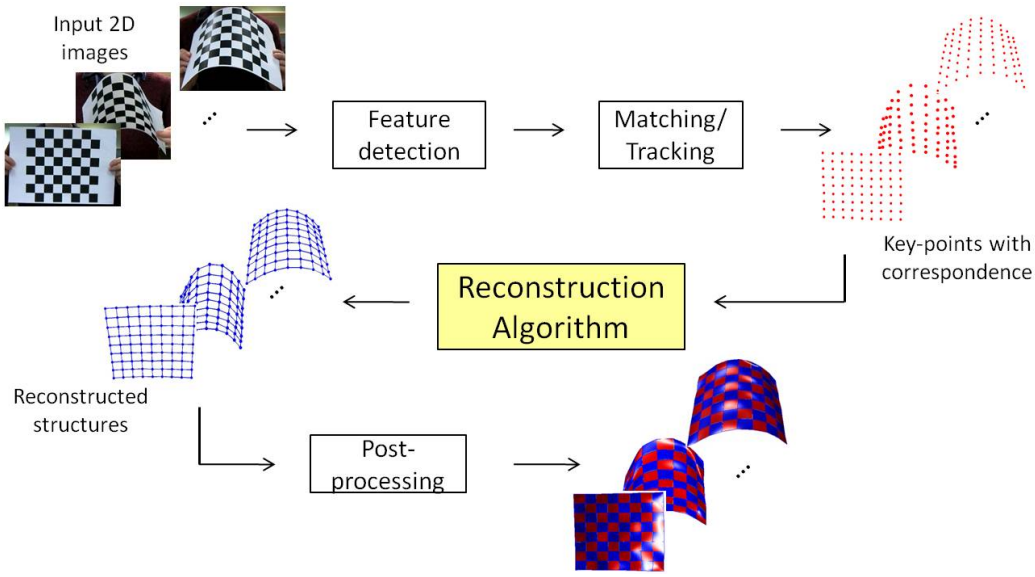


Figure 7.1: The flowchart of a complete 3D objects reconstruction system

algorithms. The same is true for the reconstruction algorithms reported in this thesis. Given that all the presented methods perform a feature based reconstruction, accurate feature detection is an important preprocessing step.

The Harris operator is one of the most popular feature detectors, which was proposed back in 1980s [62]. In order to find the distinctive features in an image, the detector calculates a corner score based on differential of the local energy. Given an image intensity I, an area indicated by co-ordinates area (x, y) and a relative shift in co-ordinates denoted by (u, v), the change E is produced by weighted sum of squared differences (SSD) between these two patches,

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u, y+v) - I(x,y)]^2 \tag{7.1}$$

where w indicates the window image. Using Taylor expansion, the image I can be rewritten as,

$$I(x+u, y+v) = I(x,y) + I_x u + I_y v + O^2(u^2, v^2) \tag{7.2}$$

Eliminate the higher order term $O^2(u^2, v^2)$ which is assumed to be small, E is approximated as,

$$\begin{aligned} E(u,v) &\approx \sum_{x,y} w(x,y)[I_x u + I_y v]^2 \\ &= [u,v]\mathbf{M}[u,v]^T \end{aligned} \tag{7.3}$$

with $\mathbf{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$. $\mathbf{M}$ can be factorised as,

$$\mathbf{M} = u^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} u \tag{7.4}$$

where u is an orthonormal matrix, λ_1 and λ_2 are the two largest eigenvalues of $\mathbf{M}$. Points of interest are defined in terms of λ_1 and λ_2, which can be grouped into three cases:

- If λ_1, λ_2 are both small, the window image is most likely in the flat region, which is not suitable to be extracted as keypoint.
- If $\lambda_1 \gg \lambda_2$ or $\lambda_2 \gg \lambda_1$, the window image is on the edge.
- If λ_1, λ_2 are both large, then a corner feature is found.

The Harris corner detector has corner selection criteria, with a score A calculated for each pixel. If the score exceeds a certain threshold, the pixel is marked as a corner. The score is calculated as,

$$A = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{7.5}$$

where k is a sensitivity parameter.

Although it is called corner detector, corner point is not the only feature which can be detected with this operator. The corner detector may be used to identify an image location with large gradients in both directions. To improve the performance,

Shi-Tomasi corner detector [120] is built entirely on the Harris detector, but directly computes the score as $A = \min(\lambda_1, \lambda_2)$. Finding maxima in the smaller eigenvalue to locate good features can efficiently detect more stable corner features, which can be used for matching or tracking.

However the major issue of the eigenvalue based corner detectors is that they are not scale-invariant, thereby they cannot provide good features for matching images with different sizes.

Scale Invariant Feature Transform, known as SIFT, is one of the most successful algorithms for keypoint detection and matching, originally introduced by Lowe in 1999 [80]. SIFT aims to find local image features invariant to image scaling, rotation, illumination changes and to some extent changes in the case of multiple images of the same scene.

The SIFT algorithm for generating the set of image features includes the following major stages: (a) Scale-space extrema detection as the first stage of the algorithm is to search the interest points over all scales and image locations. The potential interest points are identified by taking the maxima or minima of a Difference of Gaussians function that occur at multiple scales. (b) Once the potential location has been obtained, the next step is to determine the accurate position of each candidate points by interpolating of nearby data. Typically, far more keypoints are detected than needed, some of which are sensitive to noise or are localised along the edges, a low contrast points are discarded after applying a threshold on minimum contrast, and the edge responses are eliminated following an additional threshold on ratio of principal curvatures [81]. The key step to achieve invariance to image location, scale and rotation is to assign the orientations to the remaining keypoints based on local image gradient directions. (c) The descriptor vector for each keypoint is generated for the local image region that is highly distinctive to the remaining variations, such as different illumination and the

change of 3D viewpoint.

Keypoints are matched, based on the corresponding Euclidean distance.

Although SIFT attempts to approximate the Laplacian of Gaussian by a Difference of Gaussians filter in order to speed up the computation, the high dimensionality of SIFT descriptor still makes for a time-consuming matching process. In consideration of this shortcomings of SIFT, another novel detector-descriptor called SURF (Speeded Up Robust Features) was proposed later for achieving faster solution to the matching step [11]. Other extensions of SIFT descriptor, which include PCA-SIFT [75], GLOH (Gradient Location and Orientation Histogram) [87] and local descriptor HMAX [111], are shown to outperform the original method in many different tasks [88].

7.2 Video tracking

In most non-rigid object reconstruction problems, the input data is given as a video clip or an image sequence, instead of a set of independent images. In such cases, unlike image matching, the target objects are in consecutive video frames thus the motion might be predicted from the previous frames. The aim of video tracking is to generate the trajectory of each feature point over time by locating its position in every frame of the video. Tracking can be classified into three broad categories: point tracking, kernel tracking and silhouette tracking, each of which is used in different situations. For this work, reconstruction of the structure of an object requires feature points to be detected in consecutive frames. Therefore, this section concentrates on the issue of point tracking only.

Establishment of point correspondence is a complicated problem, especially in the presence of noise, occlusions, misdetections. Two types of algorithms are proposed, namely deterministic and statistical. The deterministic algorithms minimise the cost

function by applying different motion constraints. Many of the methods proposed in the literature fit into this category, including the individual, combined, and global motion constraint for coherent tracking of points that lie on the same object [140]. Later developments include the multi-camera tracking algorithm proposed in [72], which attempts to preserve temporal coherency of speed and position [72]. In contrast, probabilistic methods involve incorporating prior knowledge of the scene or object and take uncertainties into account when determining correspondence. This class of methods are more flexible, thus are able to track more complex objects in a relatively complex scene, e.g. temporary occlusion of objects as they move behind and then past the obstructions. Kalman filter [73, 22] and particle filter [50] based tracking are the most typical filtering methods. Kalman filter produces estimates of the state of a linear system where the measurements have to have Gaussian distribution. In other cases, for non-linear, not-Gaussian states, the state estimation can be performed using particle filters.

Chapter 8

Methods Comparison and Analysis

As this thesis has progressed, each chapter has included an evaluation of the proposed methods, illustrated by way of comparison to existing approaches. The main purpose of this chapter is to provide a review of all the methods proposed in the thesis.

The proposed algorithms concern the problem of recovering the 3D structure of deformable objects from a sequence of images. The methods impose different types of 3D prior shape model to better constrain the highly ambiguous problem. To investigate the performance of the algorithms developed in the course of this work, this chapter provides comparison amongst all the proposed methods, including

BPCA: The batch approach with linear constraints.

IPCA: The incremental approach of estimating the deformable objects based on linear model.

DM1: The non-linear manifold based approach. A dense shape manifold is learned using diffusion maps.

DM2: Improved version of DM1, which uses a smaller number of training samples to build a manifold.

RF: Improved version of DM1, which introduces manifold forests when learning the

shape manifold.

The testing sequences and the data preparation retain the form, as introduced in previous chapters, and require no further explanation. The results listed in Table 8.1 are the 3D reconstructed shape error for reconstructions comprising four specific facial expression sequences. Three of the sequences are different levels of surprise (mild, middle and extreme) and one talking sequence. The purpose of this experiment is to evaluate the effect on different levels of deformation, when all the proposed methods are applied on same type of shape variation. The experimental results obtained for other sequences are shown in Table 8.2.

	Mild	Middle	Extreme	Talking
BPCA	0.0960	0.1319	0.1591	0.1651
IPCA	0.0757	0.0725	0.1289	0.0986
DM1	0.0270	0.0223	0.0352	0.0350
DM2	**0.0184**	**0.0164**	**0.0208**	**0.0280**
RF	0.0213	0.0203	0.0241	0.0343

Table 8.1: Normalised mean 3D error calculated in facial related sequences.

	BPCA	IPCA	DM1	DM2	RF
Surprise	0.1591	0.1289	0.0352	**0.0208**	0.0241
Talking	0.1651	0.0986	0.0350	**0.0280**	0.0343
Cardboard	0.2648	0.2445	0.1064	0.1114	**0.0940**
Cloth	0.3739	0.1909	0.0287	0.0556	**0.0254**

Table 8.2: Normalised mean 3D error calculated in different sequences.

The linear method BPCA constraints the model based on batch PCA. The method introduced the shape constraints through integration of the prior information in the cost function. The optimal shape coefficients are found within the higher probability of learned weighting distribution. The shape model in this method is based on original low rank shape model, but relaxes the constraints for fixed basis shapes due to the

fact that shapes may not be perfectly described by eigenshapes spanning the tangent space. Whilst BPCA uses prior information learning without online learned shape prior, in contrast to IPCA - which is also based on a linear model - incrementally updates the model when new frame arrives. The method used an adaptive algorithm for construction of shape constraints imposing stability on the online reconstructed shapes.

Comparing the results produced by IPCA with BPCA, the recursive method outperforms the batch method, especially for talking sequence, due to its ability to learn from the previous shapes in the sequence; batch method can only learn the model from the training dataset, which for these experiments only consists of facial expression data (talking sequences are not included in the training data). Although the sequential solution obtained by IPCA is slightly better than BPCA, its performance is still not comparable to other non-linear methods. As reported in the Table 8.1, the results provided by linear methods BPCA and IPCA are less accurate than those obtained from non-linear approaches, especially for relatively larger deformations applied for the faces, which shows that the linear model is not able to explain non-linear deformations.

Since non-linear deformations are often observed, application of the linear model does not seem feasible for such objects. To deal with this, a series of non-linear manifold based approaches were proposed. DM1, DM2 and RF are all local methods based on non-linear manifold learning approaches. The manifold built in DM1 and DM2 is based on diffusion maps, RF uses manifold forests for manifold learning. Within these three methods, DM1 was developed first and is the precursor to the other two methods - which requires a large number of training samples to build a dense manifold. The development of RF followed this model, with the manifold forests built upon the diffusion maps. Whereas DM2 requires only a relatively small quantity of training data. It can be seen that three methods are comparable, but RF achieves better performance on most

articulated body motion sequences, as the manifold is learned from the data itself. As shown in Table 8.1, DM2 outperforms DM1. The likely cause of this is the outliers which exist in the training data provided in facial expression database. Recalling that the samples used for training in DM2 are randomly selected, it is quite possible that these outlier training shapes were fortunately discarded. Although DM2 does require a reduced quantity of training samples, it should be noticed that the computation time for DM2 exceeds that for DM1 primarily because there is no need for optimisation of the basis shapes in DM1.

Summarising the results presented in Table 8.1 and Table 8.2, it is clear that linear methods BPCA and IPCA cope well when the objects contain small and simple deformation, but fail to explain highly deformable shapes. IPCA is the only approach which is able to incrementally reconstruct the shape rather than provide the whole sequence. Although it can learn the current frame based on previous frames, as a linear method, IPCA still fails to represent complex shapes, while the reconstruction 3D error is rather small in the three non-linear approaches. An overview of the features of all the proposed methods is presented in Table 8.3.

	Manifold type	Initialisation	Recursive	Missing data
BPCA	PCA	Rigid	No	Yes
IPCA	PCA	Rigid	Yes	Yes
DM1	Diffusion maps	Linear method	No	Yes
DM2	Diffusion maps	Linear method	No	Yes
RF	Manifold forests	Linear method	No	Yes

Table 8.3: Summary of presented algorithms

Chapter 9

Conclusions

The work reported in this thesis mainly tackled the problem of modelling 3D deformable objects and estimating camera motion trajectories, based on a set of observed images. Although the reconstruction of non-rigid objects has seen significant research interest, the inherent high number of degrees of freedom render the problem difficult to solve. In the case of rigid objects, the shape of the object remains constant and the results may be gradually refined over time. In contrast, the non-rigid objects are subject to deformations and consequently it does not follow that more measurement data necessarily leads to better results. The solution strategy adopted in this work was to introduce various 3D shape prior constraints, in order to reduce the dimensionality of the problem. The following summarises the work reported in this thesis as a whole, offering insight and observations on the work. The original contributions are indicated, and improvements made over extant methods are highlighted. Then the discussion turns to the potential for further work. Various directions for future research are outlined and those most closely aligned with this research are given consideration.

9.1 Summary

In summary, the proposed algorithms are classified according to the type of prior learned shape manifold: linear manifold based approaches, which include using the most fundamental linear model PCA to learn the prior for batch and incremental algorithms, respectively; and non-linear manifold based methods, which the prior learned manifold is built upon non-linear embedding techniques.

Although the problem does contain a high number of degrees of freedom, the complexity of the problem may not be reacted by the dimensionality of the data. The whole idea of our 3D prior shape information is to constrain the shape in a trained low dimensional subspace.

9.1.1 Linear manifold based approaches

In Chapter 3, a model constraint approach was introduced to estimate the shape of a deforming object using prior learned 3D shape model. Instead of only minimising the 2D re-projection error, several constraints are imposed on shape bases and the corresponding weighting coefficients. Several extensions have been developed for this prototype algorithm. The proposed extensions include use of learned shape model and distribution of the weights in the cost function, which improves performance of the optimisation process. The idea was introduced in:

- Lili Tao, Bogdan J. Matuszewski and Stephen J. Mein, Model constraints for non-rigid structure from motion, *British Machine Vision Conference (BMVC 2011) PhD Workshop*, 2011.

Based on the batch model, the recursive method, which also uses linear manifold learning, was presented in Chapter 4 and was originally introduced in:

- Lili Tao, Bogdan J. Matuszewski and Stephen J. Mein, Non-rigid structure from motion with incremental shape prior, *19th IEEE International Conference on Image Processing (ICIP 2012)*, 2012.

The algorithm can also be extended in the case when measurement data is incomplete. The extension of the algorithm was introduced in:

- Lili Tao, Stephen J. Mein, Wei Quan and Bogdan J. Matuszewski, Recursive non-rigid structure from motion with online learned shape prior, *Computer Vision and Image Understanding 117*, 2013.

This method successfully recovers shape and camera motion parameters as new frames arrive; additionally, it allows for recursively updating the model, thus accounting for new shape variations as the object deforms over the sequence. This method is a suitable groundwork for later exploitation in real-time applications.

9.1.2 Non-linear manifold based approaches

Contemporary approaches, including the work reported in this thesis, rely on a linear model to represent the deformations of the object of interest. However, this approach is applicable to a relatively simple non-rigid object, especially when the reconstructed object is based only on a small number of basis shapes, such as facial expressions, which can be well-represented based only on a linear model. But for articulated human motion or other complex deformed surfaces, it would be difficult to constrain the shape in a linear subspace. Thus the argument was made that to persist with the linear model in the recovery of large and complex deformations can only lead to large reconstruction errors. To address this deficiency, further work is therefore required to constrain shapes to a smooth manifold, representing learned non-linear shape variability.

Chapter 5 began by introducing dimensionality reduction techniques. A comparison

was made between different types of manifold learning methods from purely linear methods to several local methods which are able to handle non-linear datasets. A non-linear shape prior was learnt using one of the graph based methods - diffusion maps. Such manifold is used as a shape prior, with the reconstructed shapes constrained to lie in the manifold. The non-linear manifold based method is more accurate and well-adapted to large deformation models, which cannot be accurately represented by a linear subspace, e.g. reconstruction of full human body. This method has been introduced in:

- Lili Tao and Bogdan J. Matuszewski, Non-rigid structure from motion with diffusion maps prior, *26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, 2013.

Based on this approach, improvements were made to the current implementation with respect to two different aspects. Firstly, in Chapter 6 the use of an alternative method to learn the manifold was introduced. The random forests technique was integrated into manifold learning and employed in the 3D reconstruction problem. The paper which related to this idea is:

- Lili Tao and Bogdan J. Matuszewski, Deformable shape reconstruction from monocular video with manifold forests, *15th International Conference on Computer Analysis of Images and Patterns (CAIP 2013)*, 2013.

Note, however, that it was established that building a dense manifold requires a large number of training samples, which may not the acceptable in practice. Another improvement is to achieve comparable reconstructed results using only a small number of training samples which was recently presented in:

- Lili Tao and Bogdan J. Matuszewski , 3D deformable shape reconstruction with diffusion maps, *24th British Machine Vision Conference (BMVC 2013)*, 2013.

9.2 Future work

The potential future directions of this research include the following

Learning manifold using 2D data

Although the additional prior used in the proposed methods was imposed to better constrain the highly ambiguous problem and achieve better reconstructed performance, using 3D data as training samples is not necessarily always feasible in many real applications. Instead, the collection of 2D data seems more applicable in practice, and therefore it is necessary to use 2D data in the construction of a shape manifold. Having obtained the shape learned manifold either using 3D or 2D data, another inevitable problem is that the shape prior are only applicable to specific types of objects, so learning the manifold based only on observations would make the problem more tractable.

One of the fundamental issues of building the manifold based on 2D data is that, when the 2D shapes are observed from different points of view, the measurement shapes may look completely different even if they are obtained from a same shape. To address this problem, the most recent research modelled the shape in a manifold feature space, in which the mapping was learned from the input data. The method proposed to use rotation invariant kernel [57] when calculating the rotation invariant similarity between two 2D shapes. But such kernel can only help to eliminate the effect when rotation happens in 2D, and fails to solve the above mentioned problem. Further research is required to address this problem.

Dense reconstruction

The recovery of the 3D structure of objects (sparse reconstruction) is achieving maturity as a research field. A large number of methods and algorithms have been developed

and there are numerous studies for improving the reconstruction performance. But few among these methods are able to handle dense 3D reconstruction, especially in the case of non-rigid objects. The work in [97] performs a real-time dense reconstruction for rigid objects in a static scene. The method successfully estimates the camera motion and simultaneously creates dense 3D surface. Modelling the detailed 3D objects is a difficult task, and becomes even more challenging when applied to a deformable object. There has been comparatively little work in the field of non-rigid dense reconstruction. Piecewise approaches [116] achieved impressive results, but the connection of all local patches together into a single smooth surface requires an additional post-processing step. Some template based approaches work only under the assumption that the exact template is known in advance.

The methods reported in this work rely on extracted features, rather than pixel level information. For achieving dense reconstructions, the system ought to be designed such that account is taken of every pixel in the input images. Further work towards this goal would need to take into consideration, firstly the matching of images, then obtaining the dense 2D correspondences between images, before using the correspondences as input for reconstruction in the next stage of the algorithm.

Real-time computation

Another problem common to most existing methods is the reliance on batch processing, the limitation of which is an inability to process data online; this is particularly problematic for objects exhibiting large deformations. Future work in this direction could investigate the use of manifold based approaches to extend the current work to more challenging cases, such as recursive method or real-time reconstruction. Whilst it is acknowledged that the current MATLAB implementation lacks actual real-time capability, the limitations are implementation specific rather than inherent in the algo-

rithmic solutions. Extension of the currently adopted approach to a real-time solution might well be feasible given the bundle adjustment used for model refinement. The task of optimising for execution time is treated by reducing computational cost, therefore given an appropriate optimisation, real-time performance could be achieved.

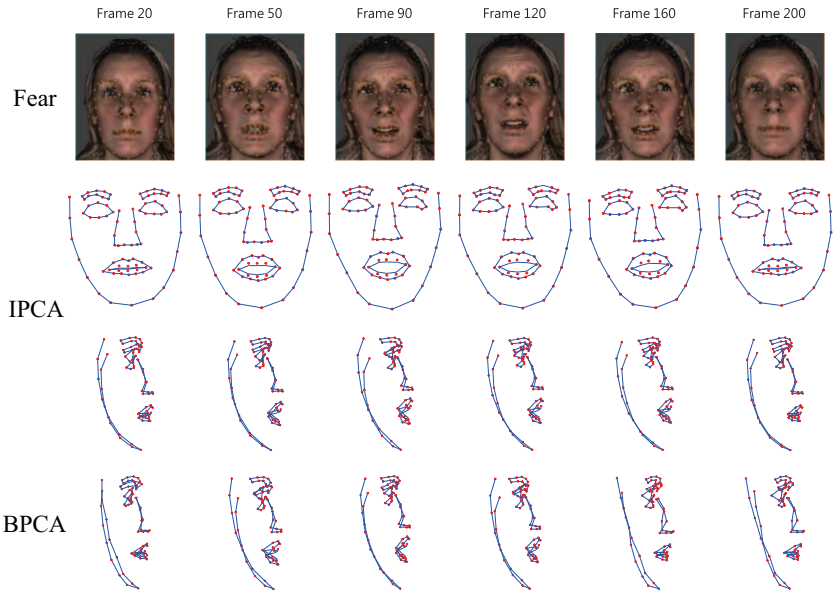
Appendices

Appendix A

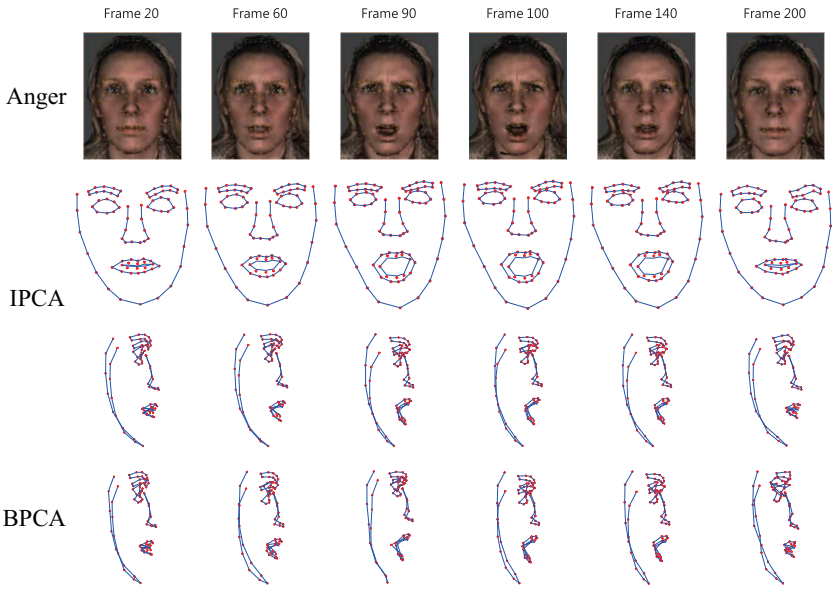
More results on IPCA and BPCA

This section provides more results on facial expression sequences. The explanation of the data is presented in Section 4.4.2.

The comparison results shown in A.1 are for tracking 83 points over a 269 frame sequence of fear and a 259 frame sequence of anger.



(a) fear facial expression sequence



(b) anger facial expression sequence

Figure A.1: Results for different facial expression sequences. First row: Input images tracked with feature points. Second and Third row: Front and side views of the 3D reconstruction using IPCA. Fourth row: Side views of the 3D reconstruction using BPCA. (a): fear facial expression sequence. (b): anger facial expression sequence.

Appendix B

Diffusion distance

This section discusses the mathematical justification of diffusion maps presented in Chapter 5.

Proposition 1: *The diffusion distance between two points in higher data space is equivalent to the Euclidean distance in the reduced diffusion space.*

Proof:

The problem is to prove that the diffusion distance between data points $\mathbf{X}_i$and$\mathbf{X}_j$ in shape space is $L^2(\mathbf{X}_i, \mathbf{X}_j)$, is equal to the Euclidean distance in reduced space $\sum_l \lambda_l^2 \left(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j)\right)^2$

The diffusion distance is given by,

$$L^2(\mathbf{X}_i, \mathbf{X}_j) = \sum_l |P_{il} - P_{lj}|^2 \tag{B.1}$$

where $P_l = \sum_l \lambda_l \varphi_l(\cdot)\phi_l^T$. φ_l and ϕ_l are the right and left eigenvector of $\mathbf{P}$. Thus the

above equation is written as,

$$\begin{aligned}&\left|\sum_l \lambda_l \varphi_l(\mathbf{X}_i)\phi_l^T - \sum_l \lambda_l \varphi_l(\mathbf{X}_j)\phi_l^T\right|^2 \\ &= \left|\sum_l \lambda_l \phi_l^T(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\right|^2\end{aligned} \tag{B.2}$$

Substitute $\phi_l = \mathbf{D}^{\frac{1}{2}}\varphi'_l$ to Equation B.2, where φ'_l is the eigenvector of $\mathbf{P}' = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$ (see Proposition 2), we then obtain,

$$\left|\sum_l \lambda_l \varphi_l'^T \mathbf{D}^{\frac{1}{2}}(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\right|^2 \tag{B.3}$$

In diffusion space, the distance can be written as,

$$\begin{aligned}&\left(\sum_l \lambda_l \varphi_l'^T(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\mathbf{D}^{\frac{1}{2}}\right)\mathbf{D}^{-1}\left(\mathbf{D}^{\frac{1}{2}}\sum_l \lambda_l \varphi_l'^T(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\right)^T \\ &= \sum_l \lambda_l \varphi_l'^T(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\sum_l \lambda_l \varphi_l'^T(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j))\end{aligned} \tag{B.4}$$

Because $\left\{\varphi_l'^T\right\}$ is an orthonormal set, thus, we get $\varphi_l'^T\varphi'_l = 0$. Therefore Equation B.4 can be rewritten as $\sum_l \lambda_l^2 \left(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j)\right)^2$, that is,

$$L^2\left(\mathbf{X}_i, \mathbf{X}_j\right) = \sum_l \lambda_l^2 \left(\varphi_l(\mathbf{X}_i) - \varphi_l(\mathbf{X}_j)\right)^2 \tag{B.5}$$

Proposition 2: *Given a diffusion operator* $\mathbf{P}$ *as* $\mathbf{P} = \mathbf{D}^{-1}\mathbf{Y}$*, the matrix* $\mathbf{P}'$ *defined as* $\mathbf{P}' = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$ *has*

1. the same eigenvalues as $\mathbf{P}$*.*

2. the left and right eigenvectors of $\mathbf{P}$*,* φ_l *and* ϕ_l *can be represented by the right eigen-*

vectors φ'_l *of* $\mathbf{P'}$ *, as* $\mathbf{D}^{\frac{1}{2}}\varphi'_l$ *and* $\mathbf{D}^{-\frac{1}{2}}\varphi'_l$*, respectively.*

Proof:

Substitute $\mathbf{P} = \mathbf{D}^{-1}\mathbf{Y}$ into $\mathbf{P'} = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$ such as,

$$\mathbf{P'} = \mathbf{D}^{-\frac{1}{2}}\mathbf{Y}\mathbf{D}^{-\frac{1}{2}} \tag{B.6}$$

Since the affinity matrix $\mathbf{Y}$ is symmetric, thus $\mathbf{P'}$ is symmetric as well. and therefore it exists an orthonormal set of eigenvectors of $\mathbf{P'}$ written as,

$$\mathbf{P'} = \mathbf{V'}\mathbf{\Lambda}\mathbf{V'}^T \tag{B.7}$$

where $\mathbf{V'}$ is and $\mathbf{\Lambda}$ the orthonormal eigenvectors and the diagonal matrix containing the eigenvalues of $\mathbf{P'}$, respectively. From $\mathbf{P'} = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}}$, we can obtain,

$$\mathbf{P} = \mathbf{D}^{-\frac{1}{2}}\mathbf{P'}\mathbf{D}^{\frac{1}{2}} \tag{B.8}$$

Substitute Equation B.7 into Equation B.8 to obtain,

$$\begin{aligned}\mathbf{P} &= \mathbf{D}^{-\frac{1}{2}}\mathbf{V'}\mathbf{\Lambda}\mathbf{V'}^T\mathbf{D}^{\frac{1}{2}} \\ &= (\mathbf{D}^{-\frac{1}{2}}\mathbf{V'})\mathbf{\Lambda}(\mathbf{D}^{-\frac{1}{2}}\mathbf{V'})^{-1} \\ &= \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}\end{aligned} \tag{B.9}$$

Therefore, the eigenvalues of $\mathbf{P}$ and $\mathbf{P'}$ are the same. The right eigenvectors of $\mathbf{P}$ is defined by the columns of $\mathbf{V} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V'}$. The same, the left eigenvectors of $\mathbf{P}$ is defined by the row of $\mathbf{V}^{-1} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V'}$. Thus the right and left eigenvectors of $\mathbf{P}$, φ_l and ϕ_l, can be derived by the eigenvectors φ'_l of $\mathbf{P'}$, such as,

$$\varphi_l = \mathbf{D}^{\frac{1}{2}}\varphi'_l \text{ and } \phi_l = \mathbf{D}^{-\frac{1}{2}}\varphi'_l \tag{B.10}$$

Bibliography

[1] Hotel motion sequence. `http://vasc.ri.cmu.edu//idb/html/motion/hotel/index.html`. CMU motion database.

[2] J.K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, 1998.

[3] J.K. Aggarwal and S. Park. Human motion: Modeling and recognition of actions and interactions. In *3D Data Processing, Visualization and Transmission*, pages 640–647. IEEE, 2004.

[4] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1442–1456, 2011.

[5] Ijaz Akhter, Yaser Sheikh, and Sohaib Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1541. IEEE, 2009.

[6] Pablo Arias, Gregory Randall, and Guillermo Sapiro. Connecting the out-of-sample and pre-image problems in kernel methods. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07.*, pages 1–8. IEEE, 2007.

[7] M. Artac, M. Jogan, and A. Leonardis. Incremental pca for on-line visual learning and recognition. In *16th International Conference on Pattern Recognition*, volume 3, pages 781–784. IEEE, 2002.

[8] S. Avidan and A. Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000.

[9] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[10] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72(3):239–257, 2007.

[11] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[12] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[13] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434, 2006.

[14] Y. Bengio, J.F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *Advances in neural information processing systems*, 16:177–184, 2004.

[15] M. Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry*. Springer, 2008.

[16] Å. Björck. *Numerical methods for least squares problems*. Siam, 1996.

[17] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[18] W. Brand. Morphable 3d models from video. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–456. IEEE, 2001.

[19] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition*, volume 2, pages 690–696. IEEE, 2000.

[20] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[21] Leo Breiman. *Classification and regression trees*. CRC press, 1993.

[22] T. J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):90–99, 1986.

[23] F. Brunet, R. Hartley, A. Bartoli, N. Navab, and R. Malgouyres. Monocular template-based reconstruction of smooth and inextensible surfaces. In *Computer Vision–ACCV 2010*, pages 52–66. Springer, 2011.

[24] A. M. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, volume 2, pages 316–322. IEEE, 2005.

[25] H. H. Buelthoff and A. L. Yuille. Shape-from-x: Psychophysics and computation. In *Fibers' 91, Boston, MA*, pages 235–246. International Society for Optics and Photonics, 1991.

[26] B. Caprile and V. Torre. Using vanishing points for camera calibration. *International journal of computer vision*, 4(2):127–139, 1990.

[27] S. Chandrasekaran, B. Manjunath, Y. Wang, J. Winkeler, and H. Zhang. An eigenspace update algorithm for image analysis. *Graphical Models and Image Processing*, 59(5):321–332, 1997.

[28] Y Choe and R. L. Kashyap. 3-d shape from a shaded and textural surface image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):907, 1991.

[29] R.R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[30] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, and S.W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, 2005.

[31] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[32] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[33] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Computer Vision*, 7:81–227, 2012.

[34] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2018–2025. IEEE, 2012.

[35] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Ninth IEEE International Conference on Computer Vision*, pages 1403–1410. IEEE, 2003.

[36] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[37] A. Del Bue. A factorization approach to structure from motion with shape priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[38] A. Del Bue, X. Llad, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1191–1198. IEEE, 2006.

[39] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007.

[40] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[41] 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 *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 771–778. IEEE, 2010.

[42] P. Etyngier, F. Segonne, and R. Keriven. Shape priors using manifold learning techniques. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[43] M. Farenzena, A. Bartoli, and Y. Mezouar. Efficient camera smoothing in sequential structure-from-motion using approximate cross-validation. In *Computer Vision–ECCV 2008*, pages 196–209. Springer, 2008.

[44] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Computer Vision–ECCV'92*, pages 563–578. Springer, 1992.

[45] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In *Computer Vision–ECCV 2010*, pages 297–310. Springer, 2010.

[46] J. Fayad, A. Del Bue, L. Agapito, and P. Aguiar. Non-rigid structure from motion using quadratic deformation models. In *British machine vision conference*, volume 33. Citeseer, 2009.

[47] J. Fortuna and A. M. Martinez. Rigid structure from motion from a blind source separation perspective. *International journal of computer vision*, 88(3):404–424, 2010.

[48] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1272–1279. IEEE, 2013.

[49] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 2004.

[50] N. J. Gordon, D. J. Salmond, and A. F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 107–113. IET, 1993.

[51] P.F.U. Gotardo and A. M. Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2051–2065, 2011.

[52] P.F.U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 802–809. IEEE, 2011.

[53] P.F.U. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3065–3072. IEEE, 2011.

[54] J.C. Gower and G.B. Dijksterhuis. *Procrustes problems*, volume 3. Oxford University Press Oxford, 2004.

[55] P. M. Hall, A. D. Marshall, and R. R. Martin. Incremental eigenanalysis for classification. In *British Machine Vision Conference*, volume 98, pages 286–295. Citeseer, 1998.

[56] P. M. Hall, A. D. Marshall, and R. R. Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.

[57] O.C. Hamsici, P.F.U. Gotardo, and A.M. Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *Computer Vision–ECCV 2012*, pages 260–273. Springer, 2012.

[58] M. Han and T. Kanade. Creating 3d models with uncalibrated cameras. In *Applications of Computer Vision*, pages 178–185. IEEE, 2000.

[59] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *Computer Vision and Pattern Recognition*, pages 542–549. IEEE, 2000.

[60] M. Han and T. Kanade. Multiple motion scene reconstruction from uncalibrated views. In *Eighth IEEE International Conference on Computer Vision*, volume 1, pages 163–170. IEEE, 2001.

[61] I. Haritaoglu, D. Harwood, and L. S. Davis. W4: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[62] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[63] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *Computer Vision–ECCV 2008*, pages 276–289. Springer, 2008.

[64] R. I. Hartley. Euclidean reconstruction from uncalibrated views. In *Applications of invariance in computer vision*, pages 235–256. Springer, 1994.

[65] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[66] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[67] A. Hertzmann and S. M Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005.

[68] A. Heyden and K. Astrom. Euclidean reconstruction from image sequences with varying and unknown focal length and principal point. In *Computer Vision and Pattern Recognition, 1997. Proceedings.*, pages 438–443. IEEE, 1997.

[69] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13):981–991, 1999.

[70] B.K.P. Horn and M.J. Brooks. *Shape from shading*. MIT press, 1989.

[71] Y. Hung and W. Tang. Projective reconstruction from multiple views with minimization of 2d reprojection error. *International Journal of Computer Vision*, 66(3):305–317, 2006.

[72] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 952–957. IEEE, 2003.

[73] Rudolph Emil Kalman et al. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[74] K. Kanatani and Y. Sugaya. Factorization without factorization: complete recipe. *Mem. Fac. Eng. Okayama Univ*, 38(1&2):61–72, 2004.

[75] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.

[76] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 225–234. IEEE, 2007.

[77] G. Klein and D. Murray. Improving the agility of keyframe-based slam. In *Computer Vision–ECCV 2008*, pages 802–815. Springer, 2008.

[78] S. Lafon, Y. Keller, and R.R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, 2006.

[79] X. Lladó, A. Del Bue, and L. Agapito. Non-rigid metric reconstruction from perspective cameras. *Image and Vision Computing*, 28(9):1339–1353, 2010.

[80] D. G Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision*, volume 2, pages 1150–1157. IEEE, 1999.

[81] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[82] B. D Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th international joint conference on Artificial intelligence*, 1981.

[83] S. Mahamud and M. Hebert. Iterative projective reconstruction from multiple views. In *Computer Vision and Pattern Recognition, 2000.*, volume 2, pages 430–437. IEEE, 2000.

[84] M. Marques and J. Costeira. Estimating 3d shape from degenerate sequences with missing data. *Computer Vision and Image Understanding*, 113(2):261–272, 2009.

[85] B. J. Matuszewski, W. Quan, L-K. Shark, A. S. McLoughlin, C. E. Lightbody, H. C. Emsley, and C. L. Watkins. Hi4d-adsip 3-d dynamic facial articulation database. *Image and Vision Computing*, 30(10):713–727, 2012.

[86] L. Mei, J. Liu, A. Hero, and S. Savarese. Robust object pose estimation via statistical manifold modeling. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 967–974. IEEE, 2011.

[87] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[88] P. Moreno, M. J. Marín-Jiménez, A. Bernardino, J. Santos-Victor, and N. de la Blanca. A comparative study of local descriptors for object category recognition: Sift vs hmax. In *Pattern Recognition and Image Analysis*, pages 515–522. Springer, 2007.

[89] F. Moreno-Noguer, J. M. Porta, and P. Fua. Exploring ambiguities for monocular non-rigid shape estimation. In *Computer Vision–ECCV 2010*, pages 370–383. Springer, 2010.

[90] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3d stretchable surfaces from single images in closed form. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1842–1849. IEEE, 2009.

[91] T. Morita and T. Kanade. A sequential factorization method for recovering shape and motion from image streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):858–867, 1997.

[92] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009.

[93] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24, 1995.

[94] B. Nadler, S. Lafon, R.R. Coifman, and I.G. Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006.

[95] H. V. Neto and U. Nehmzow. Incremental pca: An alternative approach for novelty detection. *Towards Autonomous Robotic Systems*, 2005.

[96] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1505. IEEE, 2010.

[97] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtam: Dense tracking and mapping in real-time. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2320–2327. IEEE, 2011.

[98] J. Oliensis and R. Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2217–2233, 2007.

[99] J. Östlund, A. Varol, D. T. Ngo, and P. Fua. Laplacian meshes for monocular 3d shape recovery. In *Computer Vision–ECCV 2012*, pages 412–425. Springer, 2012.

[100] M. Paladini, A. Bartoli, and L. Agapito. Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In *Computer Vision–ECCV 2010*, pages 15–28. Springer, 2010.

[101] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2905. IEEE, 2009.

[102] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[103] M. Perriollat, R. Hartley, and A. Bartoli. Monocular template-based reconstruction of inextensible surfaces. *International journal of computer vision*, 95(2):124–137, 2011.

[104] C. J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(3):206–218, 1997.

[105] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[106] L. Quan. Self-calibration of an affine camera from multiple views. *International Journal of Computer Vision*, 19(1):93–105, 1996.

[107] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[108] J. R. Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[109] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[110] V. Rabaud and S. Belongie. Linear embeddings in non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2427–2434. IEEE, 2009.

[111] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025, 1999.

[112] J.W. Roach and J.K. Aggarwal. Computer tracking of objects moving in space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):127–135, 1979.

[113] D.P. Robertson and R. Cipolla. Structure from motion. *Practical Image Processing and Computer Vision*, 2009.

[114] D. A. Ross, D. Tarlow, and R. S. Zemel. Learning articulated structure and motion. *International Journal of Computer Vision*, 88(2):214–237, 2010.

[115] S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[116] Chris Russell, Joao Fayad, and Lourdes Agapito. Dense non-rigid structure from motion. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 509–516. IEEE, 2012.

[117] M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):931–944, 2011.

[118] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3d surface registration. In *Computer Vision–ECCV 2008*, pages 581–594. Springer, 2008.

[119] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization and beyond*. the MIT Press, 2002.

[120] J. Shi and C. Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600. IEEE, 1994.

[121] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[122] D. Sinclair, L. Paletta, and A. Pinz. Euclidean structure recovery through articulated motion. In *Proceedings of the Scandinavian Conference on Image Analysis*, volume 2, pages 991–998. Citeseer, 1997.

[123] H. Strasdat, J.M.M. Montiel, and A.J. Davison. Real-time monocular slam: Why filter? In *Proceedings of the Scandinavian Conference on Image Analysis Robotics and Automation (ICRA)*, pages 2657–2664. IEEE, 2010.

[124] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Computer Vision–ECCV'96*, pages 709–720. Springer, 1996.

[125] J-P Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure-from-motion. In *Computer Vision and Pattern Recognition, 2007*, pages 1–8. IEEE, 2007.

[126] J. Taylor, A.D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2761–2768. IEEE, 2010.

[127] J.B. Tenenbaum, V. De Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[128] C. Tomasi and T. Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ., 1991.

[129] 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, 1992.

[130] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

[131] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–493. IEEE, 2001.

[132] P. Tresadern and I. Reid. Articulated structure from motion by factorization. In *Computer Vision and Pattern Recognition*, volume 2, pages 1110–1115. IEEE, 2005.

[133] B. Triggs. Factorization methods for projective structure and motion. In *Computer Vision and Pattern Recognition, 1996.*, pages 845–851. IEEE, 1996.

[134] B. Triggs. Autocalibration and the absolute quadric. In *Computer Vision and Pattern Recognition, 1997*, pages 609–614. IEEE, 1997.

[135] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustmenta modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000.

[136] E. Trucco and A. Verri. *Introductory techniques for 3-D computer vision*, volume 93. Prentice Hall Englewood Cliffs, 1998.

[137] S. Ullman. *The interpretation of visual motion*, volume 28. MIT press Cambridge, MA, 1979.

[138] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2248–2255. IEEE, 2012.

[139] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1811–1818. IEEE, 2009.

[140] C. J. Veenman, M. J.T. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54–72, 2001.

[141] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *Computer Vision–ECCV 2012*, pages 426–440. Springer, 2012.

[142] O.J. Wagner, M. Hagen, A. Kurmann, S. Horgan, D. Candinas, and S.A. Vorburger. Three-dimensional vision enhances task performance independently of the surgical method. *Surgical endoscopy*, 26(10):2961–2968, 2012.

[143] G. Wang, Z. Hu, F. Wu, and H-T. Tsui. Single view metrology from scene constraints. *Image and Vision Computing*, 23(9):831–840, 2005.

[144] G. Wang, H.T. Tsui, and Q. Wu. Rotation constrained power factorization for structure from motion of nonrigid objects. *Pattern Recognition Letters*, 29(1):72–80, 2008.

[145] R. White and D. A. Forsyth. Combining cues: Shape from shading and texture. In *Computer Vision and Pattern Recognition*, volume 2, pages 1809–1816. IEEE, 2006.

[146] M. Wilczkowiak, E. Boyer, and P. Sturm. Camera calibration and 3d reconstruction from single images using parallelepipeds. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 142–148. IEEE, 2001.

[147] J. Winkeler, B. Manjunath, and S. Chandrasekaran. Subset selection for active object recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE, 1999.

[148] R. Wolz, P. Aljabar, J.V. Hajnal, J. Lotjonen, and D. Rueckert. Manifold learning combining imaging with non-imaging information. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pages 1637–1640. IEEE, 2011.

[149] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139, 1980.

[150] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Computer Vision-ECCV 2004*, pages 573–587. Springer, 2004.

[151] J. Yan and M. Pollefeys. A factorization-based approach for articulated nonrigid shape, motion and kinematic chain recovery from video. *Pattern Analysis and Machine Intelligence*, 30(5):865–877, 2008.

[152] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3d facial expression database for facial behavior research. In *7th international conference on Automatic face and gesture recognition, 2006. FGR 2006.*, pages 211–216. IEEE, 2006.

[153] A. Zaheer, I. Akhter, M. H. Baig, S. Marzban, and S. Khan. Multiview structure from motion in trajectory space. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2447–2453. IEEE, 2011.

[154] R. Zhang, P-S Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[155] H. Zhao, P. C. Yuen, and J. T. Kwok. A novel incremental principal component analysis and its application for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(4):873–886, 2006.

[156] H. Zhou, X. Li, and A.H. Sadka. Nonrigid structure-from-motion from 2-d images using markov chain monte carlo. *IEEE Transactions on Multimedia*, 14(1):168–177, 2012.