Karlsruhe Institute of Technology

Faculty of Informatics
Institute for Anthropomatics
Chair Prof. Dr.-Ing. R. Stiefelhagen
Facial Image Processing and Analysis Group

Non-Rigid Structure from Motion for Building 3D Face Model

DIPLOMA THESIS OF

Chengchao Qu

ADVISORS

Dipl.-Inform. Hua Gao
Dr.-Ing. Hazım Kemal Ekenel

MARCH 2011

Facial Image Analysis and Processing Group
Institute for Anthropomatics, Prof. Dr.-Ing. Rainer Stiefelhagen
Karlsruhe Institute of Technology
Title: Non-Rigid Structure from Motion for Building 3D Face Model
Author: Chengchao Qu

Chengchao Qu
Mathy-Str. 35
76133 Karlsruhe
chengchao.qu@student.kit.edu

Statement of Authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the word of others.

Karlsruhe, 2011-03-31

Chengchao Qu

Abstract

In computer vision, reconstructing realistic 3D face models has been a persistent challenge over the past years. Various techniques in different research domains have been intensively studied in seeking to recover highly accurate deformable face models which are robust against noise.

This work focuses on recovering the 3D structure and motion of deformable objects from sequences of 2D feature points, which are taken by a monocular camera and these points are either tracked using a reliable motion capture system or hand-labeled afterwards. In recent years, considerable success has been achieved in this area for static scenes or rigid objects. However, with the non-rigid scenario, the problem is underconstrained and much more difficult than expected. Thus, we build a low-dimensional subspace model to describe the deformation shape bases, which finds a balance between effective modeling and restricting the degrees of freedom. In order to factorize the 2D input data into 3D structure and motion, a probabilistic framework is preferred to the deterministic closed-form solutions because of its robustness against noise, which we think is inevitable in real-world measurements. Therefore, our approach is based on a derivation of Probabilistic Principal Component Analysis (PPCA). Parameters of shape bases are distributed over a prior distribution, and learned or marginalized out by the Expectation-Maximization (EM) algorithm iteratively. We further improve this probabilistic model by endowing the shape parameter distribution with relational information using Probabilistic Relational Principal Component Analysis (PRPCA).

We address the problem of recovering camera rotation. The orthonormality constraints of the rotation matrices are also extensively studied. Instead of imposing numerical optimizations on the constraints, the internal geometric properties of the rotation matrices are taken into account. The conventional Newton's method for optimization problems is extended to the Riemannian rotation manifold, which ultimately resolves the constraints into free optimization on the manifold.

The system is evaluated on two real-world face datasets. Evaluation results of the PRPCA extension gives evidence to the improved performance over the baseline algorithm when modeling a universal model from multiple subjects. On the other hand, our manifold based optimization technique outperforms the state-of-the-art approach in almost all cases in the experiments. Robustness in handling noisy data shows the capability of our system to deal with real-world image tracks.

Kurzzusammenfassung

In der Computervision ist in den letzten Jahren die 3D-Rekonstruktion realistischer Gesichtsmodelle eine ständige Herausforderung gewesen. Zu diesem Zweck wurden diverse Techniken in unterschiedlichen Forschungsbereichen untersucht, um sehr genaue verformbare Gesichtsmodelle zu ermitteln, die robust gegen Rauschen sind.

Diese Arbeit konzentriert sich auf die Rekonstruktion der 3D-Struktur und Bewegung verformbarer Objekte aus einer Sequenz von 2D-Merkmalspunkten, die durch eine monokulare Kamera aufgenommen werden. Die Merkmalspunkte werden entweder durch ein zuverlässiges Bewegungserfassungssystem verfolgt oder von Hand nach der Aufzeichnung markiert. In den letzten Jahren wurden in diesem Bereich beachtliche Erfolge für statische Szenen und starre Objekte erzielt. Für unstarre Fälle ist dieses Problem allerdings nur schwach eingeschränkt und viel schwieriger als erwartet. Deshalb bauen wir auf einen effizienten niederdimensionalen Unterraum, der die Freiheitsgrade beschränkt, um die verformbaren Gesichtsmodelle darzustellen. Für die Faktorisierung der 2D-Eingabe in 3D-Struktur und Bewegung ist ein probabilistisches System der deterministischen, analytisch geschlossenen Lösung vorzuziehen, da dies robuster gegenüber Rauschen ist, was für reale Messungen unvermeidbar ist. Aus diesem Grund basiert unser Algorithmus auf der Probabilistic Principal Component Analysis (PPCA). Die Parameter der Formbasen besitzen eine A-priori-Verteilung verteilt und werden mittels des Expectation-Maximization (EM) Algorithmus iterativ gelernt oder marginalisiert. Darüber hinaus verbessern wir das probabilistische Modelle mit der Probabilistic Relational Principal Component Analysis (PRPCA), die den Parametern relationale Bedeutung zwischen den Frames gibt.

Wir behandeln auch das Problem, die richtigen Rotationsmatrizen der Objekte zu finden. Dafür wird die Orthonormalitätsbedingung umfangreich untersucht. Anstatt Randbedingungen für die numerische Optimierungen festzulegen, werden die internen geometrischen Eigenschaften der Rotationsmatizen berücksichtigt. Das konventionelle Newton-Verfahren wird für die Riemannschen Rotationsmannigfaltigkeit erweitert, was letztendlich die Orthonormalitätsbedingung in eine freie Optimierung auf der Mannigfaltigkeit auflöst.

Das System wird auf zwei realen Gesichtsdatenbanken evaluiert. Die Evaluationsergebnisse zeigen, dass die PRPCA Erweiterung bessere Ergebnisse als der Baseline Algorithmus erzielt, wenn ein allgemeines Gesichtsmodell für mehrere Personen konstruiert wird. Außerdem übertrifft unsere Mannigfaltigkeit-basierte Optimierungstechnik in den Experimenten die Performanz des state-of-the-art Ansatzes in fast allen Fällen. Die Robustheit im Umgang mit Rauschen zeigt die Leistungsfähigkeit im Umgang mit reellen Bildaufnahmen.

Contents

1 Introduction **1**
1.1 Motivation 1
1.2 Previous Work 2
1.2.1 Structure from Motion 2
1.2.2 Non-Rigid Structure from Motion 3
1.3 Thesis Overview 5

2 Basic Principles **7**
2.1 Expectation-Maximization Algorithm 7
2.1.1 Maximum Likelihood Estimation 7
2.1.2 Expectation-Maximization 8
2.2 Probabilistic Principal Component Analysis 10
2.2.1 Principal Component Analysis 11
2.2.2 Probabilistic Principal Component Analysis 12
2.2.3 Probabilistic Relational Principal Component Analysis 16
2.3 Manifold Optimization 18
2.3.1 The Newton′s Method 19
2.3.2 Geometric Foundation of Manifolds 19
2.3.3 Generalization of Newton′s Method on Manifold 22

3 Methodology **25**
3.1 Problem Formulation 25
3.2 Initialization 26
3.3 PPCA Shape Model 28
3.4 PRPCA Shape Model 32
3.5 Rotation Update on Manifold 36

4 Experiments **41**
4.1 Experimental Setup 41
4.2 Experimental Data 42
4.2.1 Vicon Dataset 42
4.2.2 BU-3DFE Dataset 43
4.3 Experimental Results 43
4.3.1 Results of Relational Information 43
4.3.2 NRSFM Results with Different Numbers of Basis Shapes 45
4.3.3 NRSFM Results with Noise 50
4.3.4 Subject Specific Analysis on the BU-3DFE Dataset 52
4.3.5 Convergence 53

5 Conclusion **57**

Bibliography **59**

List of Figures

1.1 Simple illustration of rank three, two and one deformation bases 4

2.1 EM algorithm in an alternative description 10
2.2 PCA example 13
2.3 PPCA process 14
2.4 PPCA and PRPCA 17
2.5 Tangent space 20
2.6 Parallel transport 21
2.7 Nonlinear optimization schemes 22
2.8 Generalization of the Newton's method on manifold 23

3.1 NRSFM factorization 26
3.2 PDM result on the Vicon dataset 35

4.1 The Vicon face dataset 43
4.2 The BU-3DFE face dataset 44
4.3 PDM result on the BU-3DFE dataset 46
4.4 Relational matrix and sample frames from the BU-3DFE dataset 47
4.5 Reconstruction error on the Vicon dataset 47
4.6 Reconstruction error on the BU-3DFE dataset 48
4.7 Reconstruction results on the Vicon dataset 49
4.8 Reconstruction results on the BU-3DFE dataset 50
4.9 Reconstruction error with additive noise on the Vicon dataset 51
4.10 Reconstruction error with additive noise on the BU-3DFE dataset 52
4.11 Convergence results without additive noise on the Vicon dataset 54
4.12 Convergence results without additive noise on the BU-3DFE dataset . . . 54
4.13 Convergence results with additive noise on the Vicon dataset 55
4.14 Convergence results with additive noise on the BU-3DFE dataset 56

List of Tables

4.1 Subject specific reconstruction results on the BU-3DFE dataset 53
4.2 Subject independent reconstruction results on the BU-3DFE dataset 53

List of Algorithms

2.1 Expectation-Maximization algorithm 9
2.2 Newton's method for optimization on manifold 23

3.1 NRSFM initialization algorithm 28
3.2 Generalized Procrustes algorithm for image alignment 34
3.3 Rotation update for optimizing the objective function on manifold 39

List of Abbreviations

AAM	Active Appearance Model
ASM	Active Shape Model
EM	Expectation-Maximization
iid	independent and identically distributed
LDA	Linear Discriminant Analysis
LDS	Linear Dynamical System
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimation
NRSFM	Non-Rigid Structure from Motion
PCA	Principal Component Analysis
PDF	Probability Distribution Function
PDM	Point Distribution Model
PPCA	Probabilistic Principal Component Analysis
PRPCA	Probabilistic Relational Principal Component Analysis
SFM	Structure from Motion
SVD	Singular Value Decomposition

1. Introduction

The last decade has seen rapidly growing attention from the researchers in computer vision and pattern recognition communities on the reconstruction of 3D shape and motion of objects over time—known as Non-Rigid Structure from Motion (NRSFM). The motivation therefor comes from the excellent ability of human being to model deformable shapes, which are ubiquitous in the world surrounding us, on all levels from micro to macro. And doubtless the recovery for face model is one of the most studied topics, which relies on the promising results of the state-of-the-art face detection [YKA02], face recognition [ZCPR03] and face tracking [YJS06] systems for providing precise input point tracks of faces. A wide range of different research fields, machine learning, computer graphics, optimization, theoretical and numerical geometry, to mention a few, are applied to find solutions.

While the simultaneous recovery of shape and motion for rigid objects using multi-view [HZ04] or factorization [TK92] has been very well understood, most objects including faces in the real-world do not only move rigidly (e.g. pose changes), they deform over time (e.g. facial expressions) as well. A big problem in practice is noise since even the state-of-the-art trackers can only provide inaccurate point tracks if placed in an unconstrained environment, which is one of the main focuses of this work.

In the rest of this chapter, the motivation and objectives will be detailed in Section 1.1, which is followed by an outline of existed researches related to this work in Section 1.2. Finally in Section 1.3 an overview of structure and content of this thesis will be described.

1.1 Motivation

Recovering scene geometry and camera motion from 2D monocular sequence of images, has achieved significant success for the 3D geometry of static objects. Colloquially, Structure from Motion (SFM) is very similar to stereo vision that 3D structure is modeled from images of the same object of which corresponding features are tracked. The distinction lies in that for SFM, images are taken at different points of time, while for the latter case, images are taken simultaneously, thus with the same 3D motion and structure. The widely used factorization method was first introduced by Tomasi and Kanade [TK92]. Orthonormality constraints are adopted on the rotation matrices in order to recover structure and motion in a single step. Unfortunately, faces, like most biological objects and natural scenes, are flexible. 3D rigid motions, i.e. camera rotation and translation, along with non-rigid deformations, stretching and bending etc., are mixed altogether in their image measurement. Hence it turns out to be a challenging and tricky task to extend the existing rigid algorithms to the non-rigid scenario.

It is known that the problem of NRSFM is inherently underconstrained and thus intractable if each point of the object moves arbitrarily. In practice, however, many objects, e.g. faces do deform under certain rules. A possible approach [BV99] is to learn an application-specific 3D model of non-rigid structure from the training data to constrain deformation. Another possibility from Ullman [Ull83] is to hard-code and learn a model incrementally. Some approaches [Bra01, TYAB01] were proposed from another perspective to remove the need of such a prior model, which is not applicable in most real-world situations. The shape model, i.e. shape bases, is treated as unknowns to be solved, with only the orthonormality constraints on camera rotations being utilized. Xiao et al. [XCK06] proved that due to lack of further constraints their method would lead to ambiguous and not optimal solutions and introduced the basis constraints.

Most of the state-of-the-art NRSFM algorithms make use of a linear subspace model to represent the shape model as a weighted combination of shape bases. In general, this model is expected to be sensitive to the manual choice of the number of bases. Additionally, Xiao and Kanade [XK04] pointed out that in case the bases are not of full rank three, it would also suffer from degeneracies. Thus, improvement over the existing NRSFM algorithm free of those issues is a main focus of this work, while ideally keeping robust to noise.

1.2 Previous Work

In this section we give a review on researches done in the past that are considered to be most relevant to our work. Algorithms for the feature extraction step e.g. face detection and tracking are outside the scope of this work. Instead we mainly concentrates on algorithms for rigid and non-rigid SFM.

1.2.1 Structure from Motion

In computer vision and the study of visual perception, Structure from Motion refers to the process of recovering the three-dimensional structure of an object by analyzing the rigid motion over a time span. Modern SFM algorithms employ the factorization method for orthographic camera projection proposed by Tomasi and Kanade [TK92]. Factorization attempts to retain the geometric invariants through the temporal window. The observation data is stacked into a matrix consisting of (x, y) points for the feature tracks. The rank theorem ensures that this input matrix can be factorized into two matrices, one corresponding to the camera motion, and the other representing the shape. Although the resulting matrices from Singular Value Decomposition (SVD) are not unique, they only differ by a linear transformation. By imposing metric constraints, the SFM problem for rigid objects is solved. And later, this approach was extended to various camera projection models.

Poelman and Kanade [PK93] studied how the orthographic SFM could be applied to the para-perspective projection, which closely approximates perspective projection while retaining linear algebraic properties. They also showed that the initial rank theorem for the orthographic case was also valid for their scenario and gave a solution with different orthonormality constraints and motion recovery techniques. Triggs [Tri96] further extended the camera model to full perspective. If there is more than one object moving in the image stream, Costeira and Kanade [CK98] presented a new method to separate them and recover independently. No prior knowledge of the number of objects included is needed because with the introduction of the new mathematical construct called shape interaction matrix, computation of each shape is not explicitly done. Han and Kanade [HK01] also solved the recovery of multiple objects for uncalibrated views. Yan and Pollyfeys [YP05] regarded articulated bodies as a combination of a number of intersecting rigid motion subspaces. They analyzed the rank constraint of two linked parts of an object and handled axes and joints separately. A novel but simple approach on the basis of subspace clustering was proposed.

1.2.2 Non-Rigid Structure from Motion

In the seminal work of Bregler et al. [BHB00] and Torresani et al. [TYAB01] for solving NRSFM in the early 2000s, they assumed that the 3D shape of an object can be explained as a linear combination of deformation shapes applied to a dominant rigid component. In this way the non-rigid scenario is formulated as a factorization problem and the low rank of the image measurements is analyzed. The advantage of the low rank linear shape model lies in that it does not prescribe any particular type of 3D shape or deformation. Because in general, this model requires that the number of basis shapes should be known, an inaccurate choice can lead to performance fall. Theoretically, if the number is underestimated, it is not sufficient to represent all variations of the object; otherwise the extra degree of freedom is unconstrained and is unlikely to generalize well, which will end up fitting noise.

Using the linear representation Xiao et al. [XCK06] proposed a closed-form scheme for solving the NRSFM problem. They proved that in the previous work imposing orthonormality constraints alone on camera rotations is only sufficient when deformations at constant velocities. In other cases the increased degree of freedom will cause the solution ambiguous and even invalid since any linear transformation of the shape bases generates a new set of eligible bases. The additional basis constraints will determine the shape bases uniquely. In [XK04], Xiao and Kanade pointed out that even enforcing both sets of linear constraints above could still lead to ambiguity, if there exist bases not of full rank three. Figure 1.1 illustrates a simple example of those degenerate deformations, which occur quite frequently in the real-world. Workaround with a further positive semi-definite constraint eliminates the ambiguity raised by rank two deformation bases. Although recently Akhter et al. [ASK09] argued that orthonormality constraints are in fact valid enough and the primary challenge in NRSFM is due to the difficulty in the optimization problem rather than the ambiguity in orthonormality constraints, the assertion by Xiao et al. is still widely accepted as conventional heuristics for closed-form approaches.

Some drawbacks in the prior work, e.g. error-prone with the amount of noise added, and inability to handle missing data tracks, are addressed initially by Torresani et al. in [THB04] and further refined and extended to weak perspective camera model in [THB08]. In their work, 3D shapes are drawn from non-uniform Probability Distribution Function (PDF) with a Gaussian prior on each shape in the subspace instead of the common linear subspace model, which is a specific usage of Probabilistic Principal Component Analysis (PPCA). The parameters of the PDF are unknown in advance, which will be optimized using a novel Expectation-Maximization (EM) algorithm together with the 3D shapes and rigid motions. In other words, the PPCA model is employed as a hierarchical Bayesian prior for the learning process. By marginalizing out deformation coefficients in

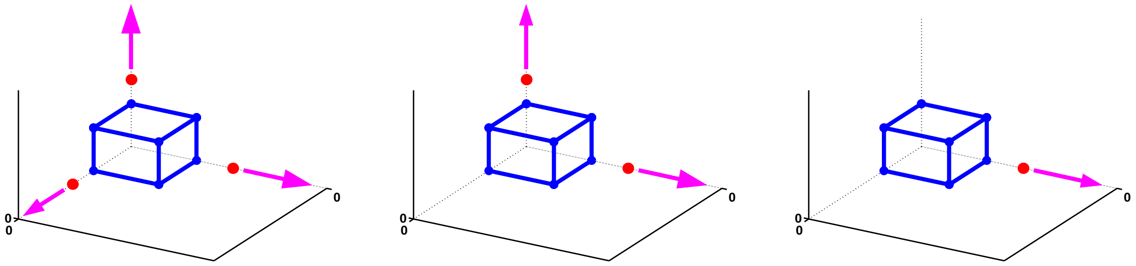


Figure 1.1: Left: Three points moving along directions in a 3D space forms a rank three deformation basis. Middle: Two points moving along directions in a 2D plane forms a rank two deformation basis. Right: One point moving along a direction forms a rank 1 deformation basis. [XK04]

the EM algorithm overfitting is avoided while the robustness against noise and missing data of this statistical model is preserved. Another advantage of PPCA over the simple subspace model is that degeneracies of closed-form solutions do not occur so that the ambiguity problem suggested by Xiao et al. [XK04] does not happen here. Since the assumption of independent and identically distributed (iid) samples from a Gaussian doesn't represent the temporal smoothing nature of the deforming object given sequential input image stream, a more sophisticated Linear Dynamical System (LDS) model is also set to replace the PPCA model and outperforms it in certain circumstances with much noise and missing data. As the importance of initialization cannot be overlooked in the iterative approach, the rigid motion and mean shape component are obtained by the Tomasi-Kanade algorithm [TK92]. The other shapes are fitted onto the remaining residual consecutively and the process is iterated. A comprehensive performance evaluation of some state-of-the-art NRSFM algorithms in this work reveals better results achieved with most synthetic and real-world datasets.

Recently more promising research on NRSFM is also done using various forms of linear and non-linear optimization techniques to minimize the 3D reprojection error. In order to overcome the degeneracy problem some additional heuristic constraints are introduced. In fact, the subspace spanned by the camera motion is a subset of a smooth manifold due to the orthogonality properties of rotation matrices. Shaji and Chandran [SC08] proposed a canonical Riemannian metric in place of the functionally and computationally convenient Euclidean metric. The span subspace of the rotation matrices and articulated shape weights can be seen as rotation group $SO(3)$ manifold and $\mathbb{R}^K$ manifold respectively, where K stands for the number of morph shapes. A main contribution of this work is the generalization of the Newton algorithm to the Riemannian case. The optimization is then performed on the tangent vectors in each tangent space along the geodesic of the product manifold. Because the convergence speed of the Hessian is quadratic, the desired solution is obtained within the first few iterations. Furthermore, the Wiberg algorithm [OD07] is employed to solve the shape update.

A novel approach was presented by Rabaud and Belongie [RB09]. Other than recovering the whole 3D shapes and motion parameters in almost all the existing applications, this approach only focuses on an embedding of the possible ones within the input image sequence. The intuition is: given enough image frames, a non-rigid deformed 3D shape can be observed several times in different view angles. If some of the frames share a low 3D reconstruction error, they are highly likely to represent a similar 3D shape, otherwise it means a poorly matched set of frames. Following this principle, triplets of frames are compared for an exploit of all repetitions in possible shape deformations. Then the generalized non-metric multi-dimensional scaling framework [AWC$^+$07] is used to estimate the weight of each deformation shape. The shape and motion are obtained by the Kronecker Constraints and rotation constraints thereafter. The last but not least, bundle adjustment is employed as a further optimization step, which minimizes the reprojection error. This closed-form approach can reach 0% of error in a clean synthetic dataset, however with the amount of noise added the performance may drop faster than statistical methods like PPCA.

Taylor et al. [TJK10] treated non-rigid 3D objects as "soup" of plausibly near rigid 3D triplets. The idea comes from the fact that even complex non-rigid motions can be decomposed into local rigid transformation groups made of few points. The algorithm starts with a traversal pairwise computation of distances in each triangle because the length of edges on a 3D triangle is computationally more efficient than pose. Only nearby features belonging to a triangle in the 2D Delaunay triangulation are considered rigid to reduce complexity. Then poses and coordinates of the triangles are independently recovered as rigid SFM using non-linear optimization. Finally the depths and flips of the triangles in each view are refined. This approach of locally rigidity does not suffer from degeneracy compared to other global approaches and comparable results are provided, too.

1.3 Thesis Overview

The major goal of this work is to learn 3D deformation shape model from 2D input image frames of human faces. Since the internal and external noise of the image tracks are inevitable, a more robust algorithm in those extreme conditions is desired.

Regarding to such problems, we address the geometric properties of the orthonormality constraints and generalize the Newton's optimization method to the underlying manifold of the camera rotation matrices. That means, non-linear optimization can be carried out on the manifold without doing any imprecise approximations. Moreover, we employ a probabilistic framework to model the NRSFM factorization, as it is more robust to noise than the closed-form factorization techniques. An advanced model with relational shape information is also given. Experimental results on the Vicon and the BU-3DFE datasets confirm that the manifold optimization approach outperforms the state-of-the-art algorithm under noisy conditions. The relational information also helps while estimating generic shape models using images of different subjects.

In the remainder of this thesis, the theoretical and functional principles of the whole system are discussed in detail. The basic ideas and theories needed for the core algorithms are described in Chapter 2. Then we explain our NRSFM and manifold-based estimation techniques in Chapter 3. Experiments are intensively conducted in Chapter 4, and the results against the state-of-the-art algorithms and their explanations are also demonstrated. In the end conclusions and directions of future research are drawn in Chapter 5.

2. Basic Principles

This chapter describes the theoretical fundamentals and core principles that our system is based on. The first section covers the Expectation-Maximization algorithm, which iteratively solves our optimization problem. Then Principal Component Analysis and its probabilistic variants that are studied extensively in this work are discussed. In the last section geometric techniques of optimization on manifolds are introduced.

2.1 Expectation-Maximization Algorithm

In this section, the Expectation-Maximization algorithm as well as the Maximum Likelihood Estimation are described, which operate together as the statistical framework of this work.

2.1.1 Maximum Likelihood Estimation

The Maximum Likelihood Estimation (MLE) is one of the most widely used parametric methods for fitting statistical models and estimating their parameters. It gains its popularity mainly from the good convergence properties with an increased number of training samples and its simplicity compared to other methods [DHS01].

Suppose that a set of samples $\mathcal{D}$ contains N independent and identically distributed (iid) samples $\mathbf{x}_1, \dots, \mathbf{x}_N$, with the assumption of statistical independence we have

$$p(\mathcal{D}|\boldsymbol{\theta}) \equiv p(\mathbf{x}_1, \dots, \mathbf{x}_N|\boldsymbol{\theta}) = \prod_{k=1}^{N} p(\mathbf{x}_k|\boldsymbol{\theta}),$$

where the unknown parameter vector $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^\top$ is seen as variable in this function, which is also known as the likelihood function of $\boldsymbol{\theta}$ with respect to $\mathcal{D}$. The goal of the MLE is to find the estimator $\hat{\boldsymbol{\theta}}$ that maximizes the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$. Due to the monotonicity of the logarithm function, it is analytically easier to define the logarithm of the likelihood function, i.e. the log-likelihood as

$$\mathcal{L}(\boldsymbol{\theta}) \equiv \ln p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{k=1}^{N} \ln p(\mathbf{x}_k|\boldsymbol{\theta}).$$

Thus the maximum likelihood estimator can be found out as

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}).$$

If the likelihood function $p(\mathcal{D}|\boldsymbol{\theta})$ is well behaved and differentiable, and $\nabla_{\boldsymbol{\theta}}$ is the gradient operator

$$\nabla_{\boldsymbol{\theta}} = \begin{bmatrix} \frac{\partial}{\partial\theta_1} \\ \vdots \\ \frac{\partial}{\partial\theta_p} \end{bmatrix},$$

finally $\hat{\boldsymbol{\theta}}$ can be obtained using standard differential calculus by taking the partial derivative with respect to $\boldsymbol{\theta}$ and setting the set of p equations to zero

$$\nabla_{\boldsymbol{\theta}}\mathcal{L} = \sum_{k=1}^{N} \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x}_k|\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial\mathcal{L}}{\partial\theta_1} \\ \vdots \\ \frac{\partial\mathcal{L}}{\partial\theta_p} \end{bmatrix} = \mathbf{0}.$$

The MLE owns some important properties [TK08]. First it is asymptotically unbiased, which means the estimate converges in the mean to the true value of the unknown parameter. And it is asymptotically consistent, i.e. the mean square of the estimates tends to zero, which provides high confidence in the result. The Cramér–Rao bound of the lowest possible value of variance is satisfied too and thus it is also asymptotically efficient.

However, note that those desirable properties of MLE are valid only for large values of n and if the data is incomplete, it would be difficult to get explicit solution for the problem.

2.1.2 Expectation-Maximization

In the last section we already know that in case of incomplete sample data the MLE is not suitable for solving the problem. For those situations, the Expectation-Maximization (EM) algorithm is a proper maximum likelihood technique for probabilistic models with missing features, called latent variables. The algorithm was initially named and explained by Dempster et al. in [DLR77] and their basic idea is to iteratively estimate the likelihood with the data that is present.

Considering a full sample $\mathcal{D}$ from a probabilistic model consisting of observed data $\mathbf{X}$ and hidden data $\mathbf{Z}$ where $\mathcal{D} = \mathbf{X} \cup \mathbf{Z}$, along with a vector of unknown parameters $\boldsymbol{\theta}$, the likelihood is maximized via

$$p(\mathbf{X}|\boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}).$$

Because here under the assumption that a straightforward optimization of $p(\mathbf{X}|\boldsymbol{\theta})$ is either impossible or difficult to be done, the likelihood function $\mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})$ with full data is employed. However, the complete dataset $\{\mathbf{X}, \mathbf{Z}\}$ is not available, only the incomplete data $\mathbf{X}$ instead. As for the latent variables $\mathbf{Z}$, values are solely possible to be estimated by the posterior distribution $p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$, which appears in the E-step (expectation step) of the EM algorithm as a replacement of the latent variables. Subsequently, in the M-step (maximization step), this expected value of log-likelihood function is maximized. This expectation is given by

$$\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^i) = \mathbb{E}_{\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^i}[\ln \mathcal{L}(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})].$$

Note that in the definition of $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$, it is a function of $\boldsymbol{\theta}$ with the old estimate $\boldsymbol{\theta}^i$ being fixed, which is the best estimate for the full distribution of the current iteration. Given this parameter estimate, the unknown data $\mathbf{Z}$ can also be marginalized and described by it. Next the parameter $\boldsymbol{\theta}^i$ with the maximum likelihood $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$ is chosen as the new value $\boldsymbol{\theta}^{i+1}$. The EM algorithm executes continuously until a certain convergence criterion is reached. The complete algorithm is summarized in Algorithm 2.1.

Algorithm 2.1 Expectation-Maximization algorithm

```
Initialize $\boldsymbol{\theta}^0$, $i = 0$.
repeat
    E-step: compute $\mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$.
    M-step: $\boldsymbol{\theta}^{i+1} \leftarrow \arg\max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}|\boldsymbol{\theta}^i)$.
    $i \leftarrow i + 1$.
until Convergence.
return $\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^i$.
```

Like many other iterative methods, the initial parameter values have a significant impact on the speed of the convergence of the EM procedure and on the quality of the final estimates. Furthermore, the algorithm also suffers from the local maximum problem for multimodal distributions. Originally the algorithm was designed to find the MLE, but with proper modification it is also capable of solving Maximum A Posteriori (MAP) problems, which differs from the MLE in that prior knowledge of the parameter in the form of a priori probability is taken into consideration, too. In this case, the posterior in the M-step becomes $\mathcal{Q}_{\mathrm{MAP}}(\boldsymbol{\theta}|\boldsymbol{\theta}^i) = \mathcal{Q}_{\mathrm{MLE}}(\boldsymbol{\theta}|\boldsymbol{\theta}^i) + \ln p(\boldsymbol{\theta})$, where a priori term $\ln p(\boldsymbol{\theta})$ is added.

A key property of the EM algorithm is that it guarantees the monotone increase of the log-likelihood of the known data $\mathbf{X}$, while the unobserved data $\mathbf{Z}$ is marginalized. To illustrate this, the log-likelihood function can also be rewritten into the following decomposition

$$\ln p(\mathbf{X}|\boldsymbol{\theta}) = \mathcal{L}(q, \boldsymbol{\theta}) + \mathrm{KL}(q||p), \tag{2.1}$$

if $q(\mathbf{Z})$ is a distribution over the latent variables and

$$\begin{aligned} \mathcal{L}(q, \boldsymbol{\theta}) &= \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta})}{q(\mathbf{Z})}, \\ \mathrm{KL}(q||p) &= -\sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \frac{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})}, \end{aligned}$$

where $\mathcal{L}(q, \boldsymbol{\theta})$ is both a functional of $q(\mathbf{Z})$ and a function of $\boldsymbol{\theta}$.

The EM algorithm is an iterative method to optimize maximum likelihood problems. The alternative description makes it possible to prove that the log-likelihood is indeed maximized. The property of the Kullback-Leibler divergence guarantees $\mathrm{KL}(q||p)$ is always non-negative, hence from Equation (2.1) we know that $\mathcal{L}(q, \boldsymbol{\theta}) \leq \ln p(\mathbf{X}|\boldsymbol{\theta})$ is satisfied. Equality is valid if and only if $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta})$. This initial state of the EM algorithm is illustrated in Figure 2.1a.

If the current parameter value is denoted as $\boldsymbol{\theta}^{\mathrm{old}}$, which is kept fixed in the E-step, the lower bound $\mathcal{L}(q, \boldsymbol{\theta}^{\mathrm{old}})$ is maximized if a proper distribution $q(\mathbf{Z})$ is found. This is based on the observation that the log-likelihood $\ln p(\mathbf{X}|\boldsymbol{\theta}^{\mathrm{old}})$ has no dependent relation with

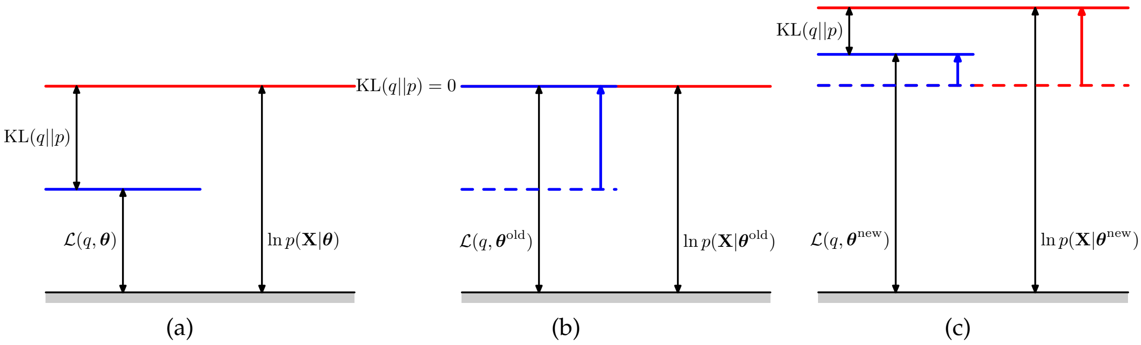

KL$(q||p)$
$\mathcal{L}(q,\boldsymbol{\theta})$
$\ln p(\mathbf{X}|\boldsymbol{\theta})$
KL$(q||p) = 0$
$\mathcal{L}(q,\boldsymbol{\theta}^{\text{old}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\text{old}})$
KL$(q||p)$
$\mathcal{L}(q,\boldsymbol{\theta}^{\text{new}})$
$\ln p(\mathbf{X}|\boldsymbol{\theta}^{\text{new}})$
(a)
(b)
(c)

Figure 2.1: Illustraion of the EM algorithm in an alternative description. (a) shows the initial state of the EM decomposition, where the KL divergence $\mathrm{KL}(q||p) > 0$ and $\mathcal{L}(q,\boldsymbol{\theta})$ sets the lower bound on the log-likelihood function $\ln p(\mathbf{X}|\boldsymbol{\theta})$. (b) reveals the E-step of the EM algorithm. Maximization of $q(\mathbf{Z})$ and the fixed parameter $\boldsymbol{\theta}^{\text{old}}$ make the lower bound approach the log-likelihood function while the KL divergence vanishing. (c) illustrates the M-step of the EM algorithm. Maximizing $\mathcal{L}(q,\boldsymbol{\theta})$ and fixing $q(\mathbf{Z})$ cause both the lower bound and the log-likelihood to go up. Because the KL divergence is not zero any more, the log-likelihood moves higher and so forth with the next EM iteration. [Bis07]

$q(\mathbf{Z})$ so when the Kullback-Leibler divergence vanishes, i.e. $q(\mathbf{Z}) = p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{old}})$, the lower bound will equal the log-likelihood, as shown in Figure 2.1b.

In the following M-step, the distribution $q(\mathbf{Z})$ is kept unmodified instead and the old parameter vector $\boldsymbol{\theta}^{\text{old}}$ is updated to $\boldsymbol{\theta}^{\text{new}}$. Maximization of the lower bound $\mathcal{L}(q,\boldsymbol{\theta})$ will result in the increase of the log-likelihood function $\ln p(\mathbf{X}|\boldsymbol{\theta})$ as well. Because $q(\mathbf{Z})$ stays fixed, it will no more equal the posterior distribution $p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{\text{new}})$ and the equality condition of the corresponding Kullback-Leibler divergence is not satisfied, either. Thus $\mathrm{KL}(q||p)$ is nonzero and there is a greater increase in the log-likelihood function than in the lower bound, as illustrated in Figure 2.1c.

Although the EM algorithm brilliantly breaks the barrier of solving some difficult MLE problems, some others still remain intractable in E-step, M-step, or even both steps. In these cases the Generalized EM algorithm is born with a bit more lax requirement than the normal one. It demands only a better $\boldsymbol{\theta}^{i+1}$ in the M-step, not necessarily the optimal value. According to the alternative representation in Equation (2.1), an improved lower bound $\mathcal{L}(q,\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ ensures the increase of the log-likelihood anyway, as seen in Figure 2.1c, unless the parameters have already reached a maximum value. It is natural that convergence of the Generalized EM algorithm will drop, but with the greater freedom it offers, much more straightforward step can be employed. And similarly, the partial update technique can also be applied to the E-step.

2.2 Probabilistic Principal Component Analysis

Principal Component Analysis is a well-established mathematical tool for data analysis and processing, which is covered in the first part of this section. In this section, we demonstrate how it arises as MLE solutions in a latent variable model to get over the absence of an associated probabilistic scheme in the original approach.

2.2.1 Principal Component Analysis

Principal Component Analysis (PCA), also known as the Karhunen–Loève transform, was invented by Pearson in [Pea01] and is nowadays extensively used in feature generation, dimensionality reduction and multivariate analysis. It has been employed for learning linear subspace models in almost all kinds of applications in computer vision, e.g. for recognition, tracking and reconstruction [CT01, SK87, TP91]. PCA is actually an orthogonal linear transformation that projects the data onto new coordinate axes according to the variance in descending order [Jol02]. After the projection, the variance of the projected data on the lower dimensional principal subspace is maximized and mutually uncorrelated [Hot33].

Let $\{\mathbf{t}_n\}$ be a set of D-dimensional input observations where $n \in \{1, \ldots, N\}$, the objective of the PCA is to find a lower Q-dimensional projection subspace with maximum variance. Let $\mathbf{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_Q\}$ be a projection matrix and the projected data

$$\mathbf{x}_n = \mathbf{W}^\top(\mathbf{t}_n - \bar{\mathbf{t}}),$$

given $\bar{\mathbf{t}}$ is the sample mean. By generating mutually uncorrelated feature vectors, PCA also appears to have some other important properties, e.g. minimum mean squared projection error. To prove this, assume that we have a set of D-dimensional orthonormal basis vectors $\{\mathbf{w}_i\}$, where $i \in \{1, \ldots, D\}$. Due to the orthonormality property, the product of these vectors satisfies the Kronecker delta

$$\delta_{ij} = \mathbf{w}_i^\top \mathbf{w}_j. \tag{2.2}$$

Then each feature vector can be represented as a linear transformation of the given basis vectors by

$$\mathbf{t}_n = \sum_{i=1}^{D} \alpha_{ni}\mathbf{w}_i, \tag{2.3}$$

where the linear transformation can be obtained with the help of Equation (2.2) as

$$\alpha_{ni} = \mathbf{t}_n^\top \mathbf{w}_i.$$

Hence Equation (2.3) can be rewritten into the form

$$\mathbf{t}_n = \sum_{i=1}^{D} \left(\mathbf{t}_n^\top \mathbf{w}_i\right) \mathbf{w}_i. \tag{2.4}$$

As we know that PCA projects the original data onto a lower dimensional space and reduces the dimensionality to $Q < D$, we can correspondingly separate the representation above to the sum of the first Q basis vectors and the rest as

$$\tilde{\mathbf{t}}_n = \sum_{i=1}^{Q} z_{ni}\mathbf{w}_i + \sum_{i=Q+1}^{D} b_i\mathbf{w}_i.$$

Our goal is to choose proper $\{\mathbf{w}_i\}$, $\{z_{ni}\}$ and b_i to minimize the mean squared error between the original data $\mathbf{t}_n$ and the approximation $\tilde{\mathbf{t}}_n$

$$J = \frac{1}{N}\sum_{n=1}^{N} ||\mathbf{t}_n - \tilde{\mathbf{t}}_n||^2. \tag{2.5}$$

By minimizing with respect to $\{z_{ni}\}$ and $\{b_i\}$ successively, it gives

$$\begin{aligned} z_{ni} &= \mathbf{t}_n^\top \mathbf{w}_i, \\ b_i &= \bar{\mathbf{t}}^\top \mathbf{w}_i. \end{aligned}$$

Again with the help of the substitution similar in Equation (2.4), we have

$$\mathbf{t}_n - \tilde{\mathbf{t}}_n = \sum_{i=Q+1}^{D} \left\{ (\mathbf{t}_n - \bar{\mathbf{t}})^\top \mathbf{w}_i \right\} \mathbf{w}_i.$$

Therefore the error measure J defined in Equation (2.5) can be further expanded as a function of $\mathbf{w}_i$ as

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=Q+1}^{D} \left(\mathbf{x}_n^\top \mathbf{w}_i - \bar{\mathbf{x}}_n^\top \mathbf{w}_i \right)^2 = \sum_{i=Q+1}^{D} \mathbf{w}_i^\top \mathbf{S} \mathbf{w}_i.$$

If J is minimized directly, degenerate solution of $\mathbf{w}_i = 0$ will occur. This can be avoided by applying a Lagrange multiplier λ_i to the additional term $\mathbf{w}_i^\top \mathbf{w}_i = 1$, which reveals

$$\widetilde{J} = \mathbf{w}_i^\top \mathbf{S} \mathbf{w}_i + \lambda_i (1 - \mathbf{w}_i^\top \mathbf{w}_i).$$

The minimum error measure can be obtained by setting the derivative with respect to $\mathbf{w}_i$ to zero, which corresponds to

$$\mathbf{S}\mathbf{w}_i = \lambda_i \mathbf{w}_i,$$

where $i \in \{1, \ldots, D\}$. Then, the mean squared error is denoted by the sum of the eigenvalues of the remaining eigenvectors vertical to the projection subspace

$$J = \sum_{i=Q+1}^{D} \lambda_i.$$

Hence the error measure is purely relevant to the selection of the eigenvalues. In other words, PCA projects the original data $\mathbf{t}_n$ onto the principal subspace spanned by the Q greatest eigenvalues, and minimizes the related reprojection error to the $D - Q$ smallest eigenvalues.

In Figure 2.2, the two principal axes are found provied the two-dimensional dataset where the variance of the projected data points reaches the maximum. When used as a dimensionality reduction technique, the multivariate samples in Figure 2.2 can be projected onto the first principal axis $\mathbf{w}_1$ with the largest variance, whereas the second principal axis $\mathbf{w}_2$ is discarded. Note that although the mentioned properties provide an excellent tool to select a reduced number of the most dominant and uncorrelated features out of the original data, consideration on the best class separability is not taken. Such case is outside the scope of this thesis and the reader is referred to Linear Discriminant Analysis (LDA) [MK01] for details.

2.2.2 Probabilistic Principal Component Analysis

The PCA has achieved a lot of successes in many fields of applications. However, a notable drawback of it is the lack of an associated probabilistic model for the observed data. In fact, Tipping and Bishop [TB99] and Roweis [Row98] have both given a probabilistic formulation of PCA, known as Probabilistic Principal Component Analysis (PPCA), which brings several appealing advantages over the conventional PCA:

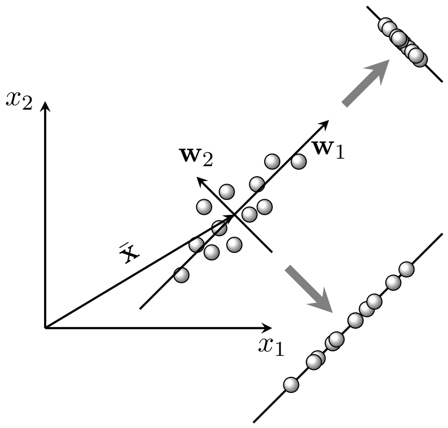


Figure 2.2: PCA seeks to project data onto a principal subspace of lower dimensionality, where the variance of the projected data maximizes.

- PPCA links PCA to a probabilistic representation and shows a constrained form of the Gaussian distribution by limiting the number of free parameters while still allowing to model the dominant correlations in the dataset.
- An efficient EM algorithm can be derived for solving PPCA iteratively, without the cost of computing the data covariance matrix, especially for large-scale applications.
- PPCA is capable of handling missing data when using the EM algorithm.
- It is more easily to generalize the single model to the mixture model case.
- The introduction of the likelihood function makes possible to fit into other probabilistic density models.

PPCA has a simple linear probabilistic assumption that all marginal and conditional distributions are Gaussian. The formulation of PPCA is closely related to factor analysis [Bar87, Bas94], in which a statistical model is used to describe the relation between a D-dimensional observed vector $\mathbf{T}$ and the corresponding Q-dimensional latent variables $\mathbf{X}$. With a $D \times Q$ projection matrix $\mathbf{W}$, a mean vector $\boldsymbol{\mu}$ and an additive isotropic Gaussian noise term $\boldsymbol{\epsilon}$ being defined, the formal definition is given by

$$\mathbf{T} = \mathbf{W}\mathbf{X} + \boldsymbol{\mu} + \boldsymbol{\epsilon},$$

where the following distributions over the latent variables $\mathbf{X}$ and the Gaussian noise process $\boldsymbol{\epsilon}$ are assumed

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \end{aligned} \tag{2.6}$$

Based on the properties of the Gaussian distribution, the conditional distribution of $\mathbf{T}$ given $\mathbf{X}$ is

$$\mathbf{T}|\mathbf{X} \sim \mathcal{N}(\mathbf{W}\mathbf{X} + \boldsymbol{\mu}, \sigma^2\mathbf{I}).$$

Using the Bayes' rule, by integrating out the latent variables $\mathbf{X}$, the marginal distribution of $\mathbf{T}$ can be obtained

$$\mathbf{T} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}),$$

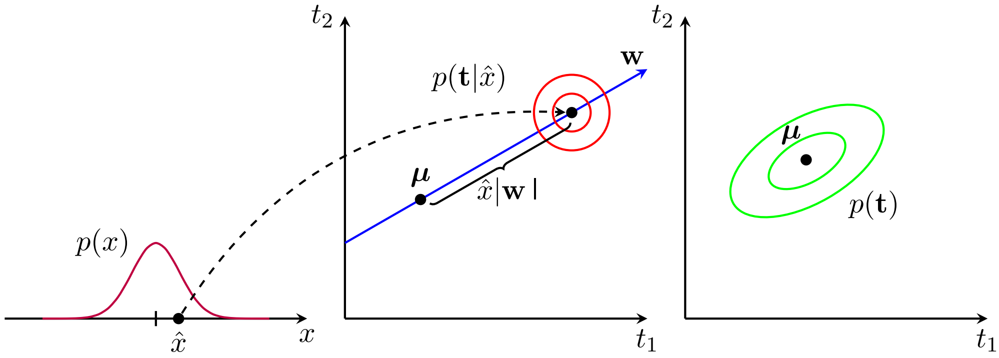


Figure 2.3: An illustration of the PPCA process with a two-dimensional data space and a one-dimensional latent space. A value of the latent variables $\hat{x}$ is drawn from the prior distribution $p(x)$. Given this value, $\mathbf{t}$ is drawn from a isotropic Gaussian distribution with mean $\mathbf{w}\hat{x} + \boldsymbol{\mu}$ and covariance $\sigma^2\mathbf{I}$, which is shown by the red circles. The marginal distribution $p(\mathbf{t})$ is illustrated by the green ellipses. [Bis07]

where the covariance model is later on replaced by $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}$ for simplicity reason. Then the log-likelihood of the observation dataset $\mathbf{T} = \{\mathbf{t}_n\}$ is

$$\begin{aligned}\mathcal{L}_{\mathrm{ML}} &= \ln p(\mathbf{T}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) \\ &= \sum_{n=1}^{N} \ln p(\mathbf{x}_n|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) \\ &= -\frac{N}{2}\left(D\ln(2\pi) + \ln|\mathbf{C}| + \mathrm{tr}(\mathbf{C}^{-1}\mathbf{S})\right),\end{aligned} \tag{2.7}$$

where

$$\mathbf{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{t}_n - \boldsymbol{\mu})(\mathbf{t}_n - \boldsymbol{\mu})^\top. \tag{2.8}$$

Figure 2.3 illustrates how PPCA maps an one-dimensional latent space onto a two-dimensional data space given the prior distribution of the latent variables. The conditional isotropic Gaussian distribution and the marginal distribution of the samples are shown by the red circles and the green ellipses respectively.

The model parameters can be determined using different methods. First, MLE can be employed to estimate $\mathbf{W}$ and σ^2 in closed form by maximization of $\mathcal{L}_{\mathrm{ML}}$ respectively

$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_Q(\boldsymbol{\Lambda}_Q - \sigma^2\mathbf{I})^{\frac{1}{2}}\mathbf{R}, \tag{2.9}$$

where the Q columns in the $D \times Q$ matrix $\mathbf{U}_Q$ are the principal eigenvectors of $\mathbf{S}$. Their counterpart eigenvalues $\lambda_1, \ldots, \lambda_Q$ form the $Q \times Q$ diagonal matrix $\boldsymbol{\Lambda}_Q$. Note that $\mathbf{R}$ is an arbitrary $Q \times Q$ orthogonal matrix, which, in practice, can be effectively ignored. Similar to the conventional PCA case, the global maximum is only possible when the greatest Q eigenvalues are in matrix $\boldsymbol{\Lambda}_Q$. In this case, if these eigenvectors are so organized that their eivenvalues are in descending order, $\mathbf{W}$ will exactly span the principal subspace of the standard PCA. When setting $\mathbf{W} = \mathbf{W}_{\mathrm{ML}}$, the MLE of the noise variance σ^2 is obtained by

$$\sigma^2_{\mathrm{ML}} = \frac{1}{D-Q}\sum_{i=Q+1}^{D}\lambda_i, \tag{2.10}$$

which has a natural interpretation associated with the discarded information in the extra dimensions.

With a huge size of dataset or an extreme high dimensionality, the closed-form method provided by the MLE is no longer suitable. Furthermore, in case of the factor analysis model without closed-form solution, or in the presence of missing data, the EM algorithm can be employed to handle these situations. The general process of the EM algorithm described in Section 2.1.2 can be applied here. In PPCA the observation data $\mathbf{T}$ is modeled over a continuous Gaussian latent space $\mathbf{X}$. In the EM approach for maximizing the likelihood estimates of the model parameters, the "missing" latent variables $\mathbf{X}$ together with the "good" data samples $\mathbf{T}$ are put together as the complete data. Because of the assumption of independent data, the log-likelihood takes the form

$$\mathcal{L}_{\text{EM}} = \ln p(\mathbf{T}, \mathbf{X}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^{N} \{\ln p(\mathbf{t}_n|\mathbf{x}_n) + \ln p(\mathbf{x}_n)\} .$$

In the E-step, taking the expectation with respect to this posterior distribution using the "old" parameter values gives

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{EM}}] = -\sum_{n=1}^{N} \Bigg\{ & \frac{D}{2}\ln(2\pi\sigma^2) + \frac{1}{2}\operatorname{tr}\left(\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top]\right) + \frac{1}{2\sigma^2}||\mathbf{t}_n - \boldsymbol{\mu}||^2 \\ & -\frac{1}{\sigma^2}\mathbb{E}[\mathbf{x}_n]^\top\mathbf{W}^\top(\mathbf{t}_n - \boldsymbol{\mu}) + \frac{1}{2\sigma^2}\operatorname{tr}\left(\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top]\mathbf{W}^\top\mathbf{W}\right)\Bigg\} .\end{aligned}$$

Note that the expectation is done with respect to the distribution $p(\mathbf{X}|\mathbf{T}, \mathbf{W}, \sigma^2)$ and terms independent of the model parameters are ignored. Then the following evaluations are made

$$\mathbb{E}[\mathbf{x}_n] = \mathbf{M}^{-1}\mathbf{W}^\top(\mathbf{t}_n - \bar{\mathbf{t}}), \tag{2.11}$$

$$\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top] = \operatorname{cov}[\mathbf{x}_n] = \sigma^2\mathbf{M}^{-1} + \mathbb{E}[\mathbf{x}_n]\mathbb{E}[\mathbf{x}_n]^\top, \tag{2.12}$$

in which $\mathbf{M}$ is defined as

$$\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}.$$

In the M-step, by keeping the posterior statistics fixed, $\mathcal{L}_{\text{EM}}$ is maximized with respect to $\mathbf{W}$ and σ^2 yields the "new" parameters

$$\begin{aligned}\mathbf{W}_{\text{new}} &= \left[\sum_{n=1}^{N}(\mathbf{t}_n - \bar{\mathbf{t}})\mathbb{E}[\mathbf{x}_n]^\top\right]\left[\sum_{n=1}^{N}\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top]\right]^{-1} \\ &= \mathbf{S}\mathbf{W}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^\top\mathbf{S}\mathbf{W})^{-1},\end{aligned} \tag{2.13}$$

$$\begin{aligned}\sigma^2_{\text{new}} &= \frac{1}{ND}\sum_{n=1}^{N}\Big\{||\mathbf{t}_n - \bar{\mathbf{t}}||^2 - 2\mathbb{E}[\mathbf{x}_n]^\top\mathbf{W}_{\text{new}}^\top(\mathbf{t}_n - \bar{\mathbf{t}}) \\ &\quad + \operatorname{tr}\left(\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top]\mathbf{W}_{\text{new}}^\top\mathbf{W}_{\text{new}}\right)\Big\} \\ &= \frac{1}{D}\operatorname{tr}(\mathbf{S} - \mathbf{S}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}_{\text{new}}^\top).\end{aligned} \tag{2.14}$$

After initialization, the EM algorithm for PPCA alternates between E-step, which evaluates the expectations over the latent space posterior using Equation (2.11) and Equation

(2.12), and M-step, which revises the parameter values in Equation (2.13) and Equation (2.14). These two steps are repeated until certain convergence criteria are satisfied. Actually these four equations can further be merged so that the appearances of $\mathbb{E}[\mathbf{x}_n]$ and $\mathbb{E}[\mathbf{x}_n\mathbf{x}_n^\top]$ are replaced by the estimates in the E-step, which gives

$$\begin{aligned}\mathbf{W}_{\text{new}} &= \mathbf{SW}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^\top\mathbf{SW})^{-1},\\ \sigma^2_{\text{new}} &= \frac{1}{D}\operatorname{tr}\left(\mathbf{S} - \mathbf{SWM}^{-1}\mathbf{W}^\top_{\text{new}}\right),\end{aligned}$$

where $\mathbf{S}$ is the covariance matrix in Equation (2.8).

2.2.3 Probabilistic Relational Principal Component Analysis

Based on the probabilistic formulation of PCA in Section 2.2.2, a lot of appealing features are made possible with the introduction of the Gaussian latent variables. Both PCA and PPCA are only valid on the assumption that the data samples are independent and identically distributed. It is true that for certain cases this assumption suffices, for many real-world applications, though, it is usually unreasonable for relational data and some intrinsic links between the data are already lost in this modeling phase [GT07]. For example, if research papers and their references are analyzed for classification in subfields, it is rational to assert that papers belonging to the same category have more cross-references between them, which can be modeled as additional knowledge of significant importance. With the iid assumption, however, this information is discarded. For the Non-Rigid Structure from Motion (NRSFM) problem, if we exploit the internal relations within temporally nearby frames, they are more similar in comparison with the frames with a large time interval between them. Li et al. [LYZ09] proposed an novel extension of PPCA, called Probabilistic Relational Principal Component Analysis (PRPCA), for relational data analysis.

Remember that in PPCA, the latent variable matrix $\mathbf{X}$ is denoted

$$\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Phi}),$$

where the covariance $\mathbf{\Phi}$ is defined as the identity matrix $\mathbf{I}$ for iid variables in Equation (2.6). Actually $\mathbf{\Phi}$ reflects on the semantics within the data, so if it is substituted by a non-identity matrix, the iid assumption is easily eliminated. Hence, one of the essential tasks of PRPCA are to figure out a reasonable covariance matrix $\mathbf{\Phi}$ that represents the measure of relation between the latent variables so that if there exists a high relation between them, it should be modeled as close as possible.

Suppose that the links between two instances are always positively correlated. Euclidean distance can be employed to define the gap. With the observation that the larger the retained variance of the latent variables $\mathbf{X}$ are, the lower the probability density at $\mathbf{X}$ with respect to the prior is, the density function should be given a low value if the link between the instances is close. Thus when a symmetric $Q \times Q$ matrix $\mathbf{A}$ is defined to indicate the positive link, a decent relational matrix $\mathbf{\Delta}$ can be given as

$$\mathbf{\Delta} = \gamma\mathbf{I} + (\mathbf{I} + \mathbf{A})^\top(\mathbf{I} + \mathbf{A}),$$

where γ is typically a very small value solely to keep the relation matrix $\mathbf{\Delta}$ to be positive. Because commonly if there is a link from point i to j, the link in the inverse direction from point j to i also stands. That says $\mathbf{A}_{ij} = \mathbf{A}_{ji}$. Thus due to the symmetric property of matrix $\mathbf{A}$, the relational matrix $\mathbf{\Delta}$ can be written as

$$\mathbf{\Delta} = \gamma\mathbf{I} + (\mathbf{I} + \mathbf{A})(\mathbf{I} + \mathbf{A}) \tag{2.15}$$

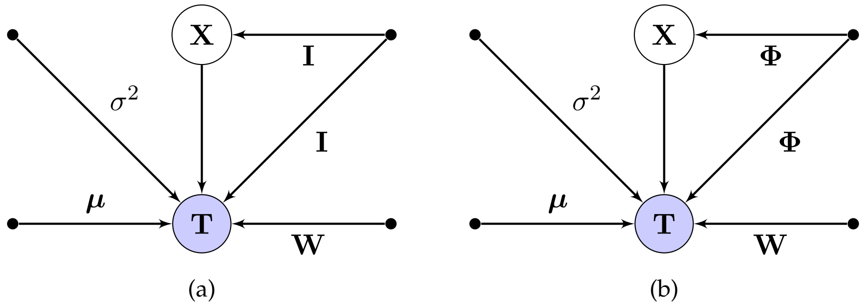


Figure 2.4: Graphical models of PPCA in (a) and PRPCA in (b), in which the observation matrix $\mathbf{T}$ can be expressed as a directed graph associated with the latent variable matrix $\mathbf{X}$, while parameter values of $\boldsymbol{\mu}$, $\mathbf{W}$ and σ^2 are learned.

as well. Hence if let the covariance matrix $\boldsymbol{\Phi}$ be the inverse of the relational matrix $\boldsymbol{\Delta}$

$$\boldsymbol{\Phi} = \boldsymbol{\Delta}^{-1},$$

the prior for $\mathbf{X}$ is indeed set to a lower value if the relation values in $\mathbf{A}$ as well as in $\boldsymbol{\Delta}$ appear to be large. A detailed proof is out of the scope of this thesis, so it is not provided here. The reader is referred to [LYZ09].

With an appropriate covariance matrix $\boldsymbol{\Phi} = \boldsymbol{\Delta}^{-1}$ being given, the general PRPCA model is defined as follows:

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}) \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}) \\ \mathbf{T} &= \mathbf{W}\mathbf{X} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \end{aligned}$$

Further distributions very close to those in PPCA can be obtained

$$\mathbf{T}|\mathbf{X} \sim \mathcal{N}(\mathbf{W}\mathbf{X} + \boldsymbol{\mu}, \sigma^2\mathbf{I}), \tag{2.16}$$

$$\mathbf{T} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}\mathbf{W}^\top\boldsymbol{\Phi} + \sigma^2\mathbf{I}). \tag{2.17}$$

The graphical models of PPCA and PRPCA are illustrated in Figure 2.4a and Figure 2.4b. The difference between both algorithms is quite small and both can be expressed as a directed graph associated with the latent variable matrix $\mathbf{X}$ and the observation matrix $\mathbf{T}$, with only $\mathbf{I}$ being replaced by $\boldsymbol{\Phi}$. That again proves that the data here is correlated compared to those independent samples in PPCA. Actually if the iid assumption applies, i.e. $\mathbf{A} = \mathbf{0}$, the covariance matrix $\boldsymbol{\Phi}$ approximates the identity matrix $\mathbf{I}$, thus PRPCA degenerates to PPCA in this case, as we may derive from the equations above.

If the covariance matrix of the marginal distribution of the observation matrix $\mathbf{T}$ is denoted $\mathbf{C} = \mathbf{W}\mathbf{W}^\top + \sigma^2\mathbf{I}$, the log-likelihood function is derived similar as in Equation (2.7) for PPCA in the form of

$$\begin{aligned} \mathcal{L}_{\mathrm{ML}} &= \ln p(\mathbf{T}|\boldsymbol{\mu}, \mathbf{W}, \sigma^2) \\ &= -\frac{N}{2}\left(D\ln(2\pi) + \ln|\mathbf{C}| + \mathrm{tr}(\mathbf{C}^{-1}\mathbf{H})\right), \end{aligned} \tag{2.18}$$

where terms irrelevant to the parameters are discarded and $\mathbf{H}$ is defined as

$$\mathbf{H} = \frac{1}{N}(\mathbf{T} - \boldsymbol{\mu})\boldsymbol{\Delta}(\mathbf{T} - \boldsymbol{\mu})^\top. \tag{2.19}$$

If we make a comparison of Equation (2.7) and Equation (2.18), the difference is only between $\mathbf{S}$ and $\mathbf{H}$ as the relational matrix $\boldsymbol{\Delta}$ is in the place of the iid matrix $\mathbf{I}$. Therefore, the existing learning methods for PPCA may still be used with little modification. For the closed-form solutions using MLE, it is even in the same form for the projection matrix $\mathbf{W}_{\text{ML}}$ and the noise variance σ^2_{ML}, which are provided in Equation (2.9) and Equation (2.10) respectively.

In the EM formulation, the observation data $\mathbf{T}$ and the latent variables $\mathbf{X}$, seen as missing data, are treated together as the complete dataset, whereas $\mathbf{W}$ and σ^2 as parameters. The complete log-likelihood is

$$\mathcal{L}_{\text{EM}} = \ln p(\mathbf{T}, \mathbf{X} | \boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \ln p(\mathbf{T}|\mathbf{X}) + \ln p(\mathbf{X}).$$

With the help of the Bayes' rule, the posterior distribution of the latent variables $p(\mathbf{X}|\mathbf{T})$ can be derived from Equation (2.16) and Equation (2.17). In the E-step, the estimates of

$$\mathbb{E}[\mathbf{X}] = \mathbf{M}^{-1}\mathbf{W}^\top(\mathbf{T} - \boldsymbol{\mu}) \tag{2.20}$$

$$\mathbb{E}[\mathbf{X}\boldsymbol{\Delta}\mathbf{X}^\top] = N\sigma^2\mathbf{M}^{-1} + \mathbb{E}[\mathbf{X}]\boldsymbol{\Delta}\mathbb{E}[\mathbf{X}]^\top \tag{2.21}$$

are calculated, where

$$\mathbf{M} = \mathbf{W}^\top\mathbf{W} + \sigma^2\mathbf{I}$$

and the expectation of the complete log-likelihood function is given as

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{\text{EM}}] = -\frac{ND}{2}\ln\sigma^2 - \frac{1}{2\sigma^2}\Big\{&\operatorname{tr}\Big((\mathbf{T}-\boldsymbol{\mu})\boldsymbol{\Delta}(\mathbf{T}-\boldsymbol{\mu})^\top\Big) \\ &-2\operatorname{tr}\Big((\mathbf{T}-\boldsymbol{\mu})\boldsymbol{\Delta}\mathbb{E}[\mathbf{X}]^\top\mathbf{W}^\top\Big) + \operatorname{tr}\Big(\mathbf{W}^\top\mathbf{W}\mathbb{E}[\mathbf{X}\boldsymbol{\Delta}\mathbf{X}^\top]\Big)\Big\}.\end{aligned}$$

In the next M-step to maximize the complete log-likelihood, the parameters $\{\mathbf{W}, \sigma^2\}$ are updated to the new values

$$\begin{aligned}\mathbf{W}_{\text{new}} &= (\mathbf{T}-\boldsymbol{\mu})\boldsymbol{\Delta}\mathbb{E}[\mathbf{X}]^\top\mathbb{E}[\mathbf{X}\boldsymbol{\Delta}\mathbf{X}^\top]^{-1} \\ &= \mathbf{H}\mathbf{W}(\sigma^2\mathbf{I} + \mathbf{M}^{-1}\mathbf{W}^\top\mathbf{H}\mathbf{W})^{-1},\end{aligned}$$

$$\sigma^2_{\text{new}} = \frac{1}{D}\operatorname{tr}(\mathbf{H} - \mathbf{H}\mathbf{W}\mathbf{M}^{-1}\mathbf{W}^\top_{\text{new}}),$$

where $\mathbf{S}$ is defined in Equation (2.19).

2.3 Manifold Optimization

The optimization problem with constraints seeks to maximize or minimize a function, while regular constraints terms are to be satisfied. The conventional approach for this problem is to impose weighted cost to the constraints and solves the optimization problem of the sum of the objective function and the cost functions. Examples of such numerical optimization techniques are the Lagrange multiplier, or more generalized Karush–Kuhn–Tucker conditions. However, the optimization community has long been aware of the fact that linear and quadratic functions with some specific constraints, e.g. the orthonormality constraints, have special structure to exploit. In fact, Stiefel (or Grassmann) manifold have also the geometric meaning representing these constraints. On the other hand, the Newton's method has been widely used for hundreds of years as a nonlinear analysis tool to find good approximations to the maximum or the minimum of functions. Hence in this section, geometric insights of the underlying constraints for optimization have been addressed. At first, the Newton's method on the Euclidean space and a brief introduction to the manifold geometry are given. In the next part, a generalization of the Newton's method on the manifold is described.

2.3.1 The Newton's Method

The Newton's method, also known as the Newton–Raphson method, named after Isaac Newton and Joseph Raphson, is originally a nonlinear technique to iteratively approximate the roots of the functions. Starting from an initial guess, the method calculates the tangent line and moves the iteration point to its intercept of the x-axis, which is usually a better approximation and can be used for the next iteration. The Newton's method is also an ideal approach to find the stationary points of differentiable functions.

Assuming that $f(x)$ is a twice-differentiable function on $\mathbb{R}^n$, a necessary (and sometimes sufficient) condition for a minimum at point $x^* \in \mathbb{R}^n$ is that

$$\nabla f(x^*) = 0.$$

If $f(x)$ is continuously differentiable up to second order for every point $x \in \mathbb{R}^n$, the update sequence x^n can be approximated by the Taylor series expansion up to the second order, which yields

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = 0.$$

Suppose that the Hessian $\nabla^2 f(x^k)$ is non-degenerate, thus invertible, the previous equation can be solved with the answer x^{k+1} as

$$x^{k+1} = x^k - \frac{\nabla f(x^k)}{\nabla^2 f(x^k)}. \tag{2.22}$$

Provided a good initial point is given, the Newton's method owns some outstanding properties [Avr03]:

- Knowledge up to only second order of the function at the current point required.
- Locally quadratic rate of convergence to a local minimum in general.
- Convergence in a single iteration for quadratic functions.

The Newton's method on the Euclidean space simply updates the current iteration point by subtracting the gradient vector multiplied by the inverse of the Hessian. Only knowledge of the first and second order derivatives are required in this case. In the remaining part of this section, insights of its generalization on manifolds are given.

2.3.2 Geometric Foundation of Manifolds

Manifold is a topological space, which is locally $\mathbb{R}^n$. That means, a small enough scale of the manifold resembles the Euclidean space. And the dimension of that scale represents the dimension of the manifold. The simplest manifold is the Euclidean space itself. Other examples such as circle (one-dimensional) and sphere (two-dimensional) are also familiar to us.

However, in most real-world applications some more specific kinds of manifolds with calculus are required. For differentiable geometry, a smooth manifold is a differentiable manifold if all orders of derivatives exist. Another example is that in order to measure distances and angles of a differentiable manifold, a metric, or in inner product $\langle \cdot, \cdot \rangle$ must be endowed for each tangent space. This is named as Riemannian manifold. Various notions like volumes and curvature can be defined. For instance, the Euclidean space with the Euclidean distance as its metric is the most general case of a Riemannian manifold.

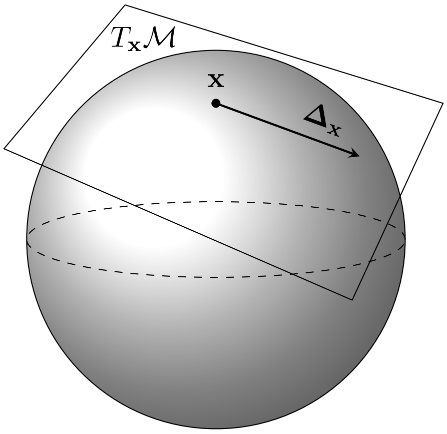


Figure 2.5: Tangent space $T_{\mathbf{x}}\mathcal{M}$ at the point $\mathbf{x}$ on the standard 2D sphere $\mathbb{S}^2$. $\boldsymbol{\Delta}_{\mathbf{x}} \in T_{\mathbf{x}}\mathcal{M}$ is a tangent vector passing through $\mathbf{x}$.

In this thesis, the manifold of the three-dimensional special orthogonal group $SO(3)$ is extensively studied, which contains orthogonal matrices $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ with determinant 1, so formally

$$SO(3) = \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} : \mathbf{R}^\top\mathbf{R} = \mathbf{I}, \det(\mathbf{R}) = 1\right\}. \tag{2.23}$$

Actually it is a special instance of Stiefel manifold

$$V_k(\mathbb{R}^n) = \left\{\mathbf{A} \in \mathbb{R}^{n\times k} : \mathbf{A}^\top\mathbf{A} = \mathbf{I}\right\},$$

which generalizes to $O(n)$ when $k = n$.

The tangent space at the point $\mathbf{x}$ on the differentiable manifold is the unique tangent plane to the submanifold at that point. Informally it contains all tangent vectors that pass through $\mathbf{x}$. For example, Figure 2.5 illustrates the tangent space $T_{\mathbf{x}}\mathcal{M}$ at the point $\mathbf{x}$ on the sphere $\mathbb{S}^2$, which is a two-dimensional manifold. Because the dimension of the tangent space is the same as the dimension of the manifold, we observe that both the tangent plane and the sphere are two-dimensional. $\boldsymbol{\Delta}_{\mathbf{x}}$ is one of the tangent vectors that constitute the tangent space. On the sphere it is perpendicular to the radii. If we make a deeper understanding of the tangent vectors on the manifold of orthogonal groups, e.g. Stiefel manifold, differentiating $\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}$ reveals

$$\mathbf{Y}^\top\boldsymbol{\Delta} + \boldsymbol{\Delta}^\top\mathbf{Y} = \mathbf{0}, \tag{2.24}$$

which leads to the verdict that $\mathbf{Y}^\top\boldsymbol{\Delta}$ is a skew-symmetric matrix, a matrix whose transpose is its negative. In case of the rotation group $SO(3)$, this can be investigated from another perspective. Since it is a Lie subgroup as well as the general linear group $GL(3)$ [Lee03], the Lie algebra $\mathfrak{so}(3)$ associated with $SO(3)$ covers all skew-symmetric 3×3 matrices. Note that unlike the case on the Euclidean space, where vectors can be moved in parallel straightforwardly by merely changing the base point of the arrow. But on the embedded manifold, if we still move the tangent vector at the point $\mathbf{Y}(0)$ to a new location $\mathbf{Y}(\epsilon)$ in the same way, it usually does not guarantee that the vector is still a tangent vector at $\mathbf{Y}(\epsilon)$. By subtracting the component in the direction of the normal vector, the direction of the new tangent vector is shown in Figure 2.6

Metric is the distance between two points on the space. On the space $\mathcal{M}$, the metric function is as follows:

$$g : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$$

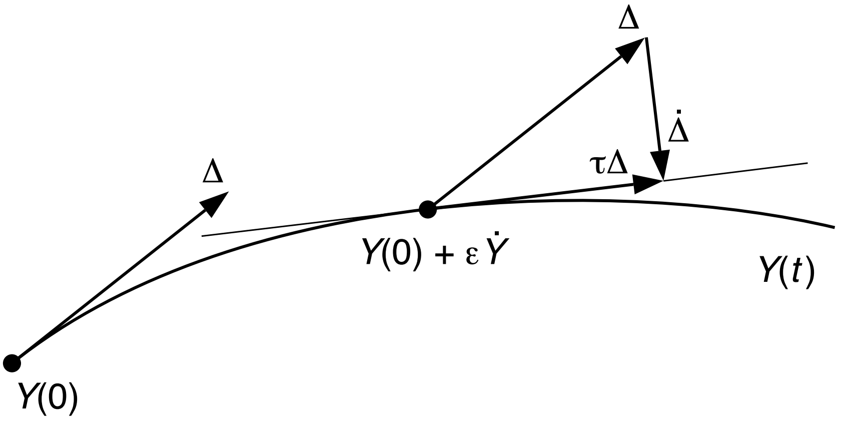


Figure 2.6: Parallel transport of a tangent vector at the point $\mathbf{Y}(0)$ to a new location $\mathbf{Y}(\epsilon)$. The direction of the new tangent vector can be obtained by removing the component in the direction of the normal vector. [EAS99]

For $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{M}$, a well-defined metric is required to be

- positive definite, which means $g(\mathbf{x}, \mathbf{y}) \geq 0$ and the equality holds only when $\mathbf{x} = \mathbf{y}$,
- symmetry, i.e. $g(\mathbf{x}, \mathbf{y}) = g(\mathbf{y}, \mathbf{x})$, and
- triangle inequality with $g(\mathbf{x}, \mathbf{z}) \leq g(\mathbf{x}, \mathbf{y}) + g(\mathbf{y}, \mathbf{z})$.

For Stiefel manifold, if the equality of all points on the manifold is taken into account, the canonical metric is given by Edelman et al. [EAS99] varies in accordance with the location $\mathbf{Y}$ in the form of

$$g_c(\mathbf{\Delta}, \mathbf{\Delta}) = \operatorname{tr}\left(\mathbf{\Delta}^\top(\mathbf{I} - \frac{1}{2}\mathbf{Y}\mathbf{Y}^\top)\mathbf{\Delta}\right).$$

In case of the rotation group $SO(3)$ where $k = n$, the canonical metric is simply

$$g_c(\mathbf{\Delta}, \mathbf{\Delta}) = \frac{1}{2}\operatorname{tr}(\mathbf{\Delta}^\top\mathbf{\Delta}).$$

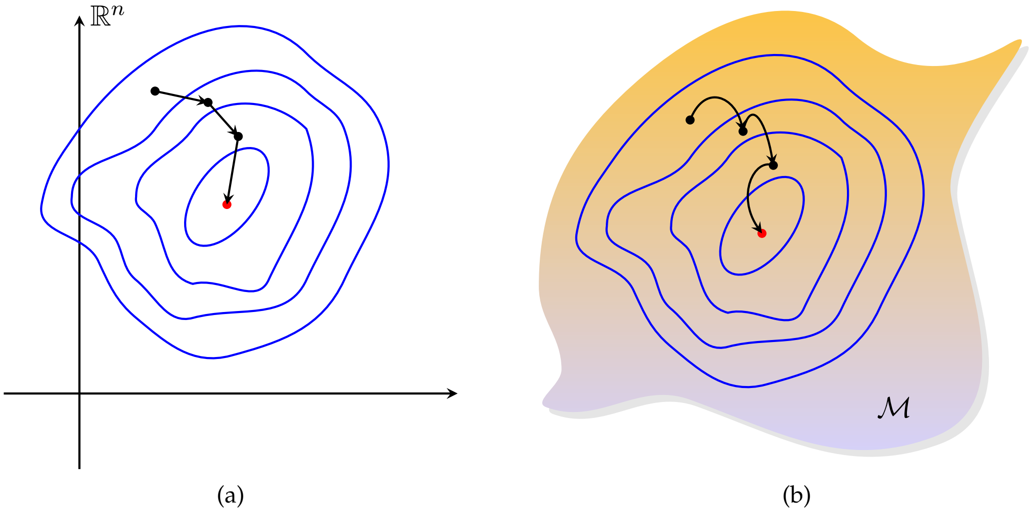
As we all know, the shortest path between two points on the Euclidean space is straight line. On the manifold instead, the notion should be generalized to curved path, namely the geodesic, which gets its name from the old science for measurement study of the earth, geodesy. To start with, let's consider the case on the sphere. If we keep the acceleration constant, the acceleration vector is normal to the radius, and hence the path is the great circle of the sphere. To calculate the geodesic, from the definition point of view, it is the same to minimize the curve length at $\mathbf{Y}(t)$

$$L = \int \sqrt{g_c(\dot{\mathbf{Y}}, \dot{\mathbf{Y}})}\,\mathrm{d}\,t.$$

After some steps of derivations [EAS99], the geodesic function for orthogonal group is given by

$$\mathbf{Y}(t) = \mathbf{Q}\exp(\mathbf{A}t), \tag{2.25}$$

where $\mathbf{Q}$ is the starting point at $t = 0$ and $\mathbf{A}$ is a skew-symmetric matrix related to the tangent vector.



(a) (b)

Figure 2.7: Comparison between nonlinear optimization schemes on the Euclidean space and on the manifold.

2.3.3 Generalization of Newton's Method on Manifold

The Newton's method for optimizing a function $f(x)$ introduced in Section 2.3.1 is only valid when x belongs to an open subset of the Euclidean space. If the variable is subject to some special constraints, e.g. the orthonormality constraints, many conventional studies are unaware of the geometric meanings of the underlying manifold space and the manifold is embedded as a submanifold into the Euclidean space $\mathbb{R}^n$ of a higher dimension and an impose additional constraints to approximate the maximum or minimum. However, if carrying out the Newton steps on the proper manifold, the problem can be eventually turned into an unconstrained solution. That is why the Newton's method is generalized on manifold [Man04]. Figure 2.7 illustrates a demonstrative example of the similarity and disparities of the non linear optimization schemes on the Euclidean space and on the manifold. In Figure 2.7a, the update steps are straight lines, while in Figure 2.7b, the updates must be done on the manifold along the geodesics.

Formulated in Equation (2.22), the Newton's method updates the current location by subtracting the gradient vector multiplied by the inverse of the Hessian. The calculation of the gradient and the Hessian depends on the choice of the metric. The gradient on the manifold is in fact a tangent vector, in which direction the objective function value increases the fastest. Hence at point $\mathbf{Y}$ on the manifold, the gradient vector ∇F for function $F(\mathbf{Y})$ is defined as

$$\mathrm{tr}(F_{\mathbf{Y}}^{\top}\boldsymbol{\Delta}) = g_c(\nabla F, \boldsymbol{\Delta})$$

for an arbitrary tangent vector $\boldsymbol{\Delta}$, where $F_{\mathbf{Y}}$ is taken as the directional derivative of F with respect to all components in $\mathbf{Y}$. Alternatively, if $\mathbf{Y}(t)$ is seen as a moment on the geodesic, the gradient as well as the Hessian may be written as follows:

$$\nabla F(\boldsymbol{\Delta}) = \left.\frac{\mathrm{d}\, F(\mathbf{Y}(t))}{\mathrm{d}\, t}\right|_{t=0} \tag{2.26}$$

$$\mathrm{Hess}\, F(\boldsymbol{\Delta}, \boldsymbol{\Delta}) = \left.\frac{\mathrm{d}^2\, F(\mathbf{Y}(t))}{\mathrm{d}\, t^2}\right|_{t=0} \tag{2.27}$$

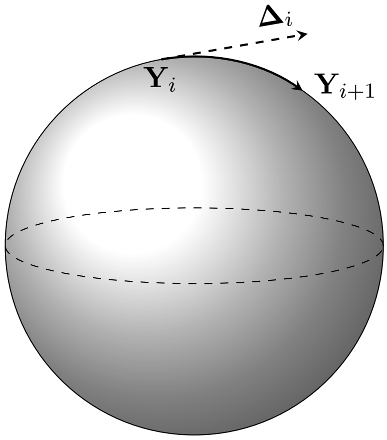


Figure 2.8: Generalization of the Newton's method on manifold. Current approximation is updated in the direction of the optimal update vector $\mathbf{\Delta}_i \in T_{\mathbf{Y}_i}\mathcal{M}$ by a distance of $\sqrt{g_c(\mathbf{\Delta}_i, \mathbf{\Delta}_i)}$. Applying the update on the geodesic reveals the new point $\mathbf{Y}_{i+1}$.

For the Newton's method, assuming that the Hessian is invertible, the optimal update vector is the tangent vector that satisfies $\mathbf{\Delta} = -\operatorname{Hess}^{-1}\mathbf{G}$, or equivalently

$$\operatorname{Hess} F(\mathbf{\Delta}, \mathbf{X}) = g_c(-\mathbf{G}, \mathbf{X})$$

for all tangent vectors $\mathbf{X}$, where $\mathbf{G} = \nabla F$ as the gradient.

Now that we have accomplished all the required elements for the Newton's method, to be specific, the gradient, the Hessian and the update path on the geodesic. The whole process remains unchanged mostly with only a few modifications, as shown in Figure 2.8. In each iteration, an optimal update vector $\mathbf{\Delta}_i \in T_{\mathbf{Y}_i}\mathcal{M}$ is computed using the canonical metric at $\mathbf{Y}_i$. Then the current approximation is updated in this direction on the geodesic by a distance of $\sqrt{g_c(\mathbf{\Delta}_i, \mathbf{\Delta}_i)}$ to $\mathbf{Y}_{i+1}$. The algorithm is summarized in Algorithm 2.2.

Algorithm 2.2 Newton's method for optimization $F(\mathbf{Y})$ on manifold

```
repeat
    At the point $\mathbf{Y}$, compute the optimal update vector $\mathbf{\Delta}$.
        2(i).   Compute the gradient $\mathbf{G}$.
        2(ii).  Compute the Hessian $\operatorname{Hess}^{-1}$.
        2(iii). Obtain $\mathbf{\Delta} = -\operatorname{Hess}^{-1}\mathbf{G}$.
    Move from $\mathbf{Y}$ in direction $\mathbf{\Delta}$ along the geodesic using Equation (2.25) by a distance
    of $\sqrt{g_c(\mathbf{\Delta}_i, \mathbf{\Delta}_i)}$.
until Convergence.
```

3. Methodology

Having given an introduction to the theoretical principles of this work, this chapter describes the design and implementation for the NRSFM task. First, we start with our general NRSFM formulation. The model initialization technique is described next. In the subsequent sections, the PPCA framework and its modification PRPCA are presented. Last but not least, our generalization of the Newton's method on the Riemannian manifold for the camera rotation update is given.

3.1 Problem Formulation

NRSFM seeks to reconstruct three-dimensional structure of the deformable object and camera motion from a series of two-dimensional monocular image tracks. In our setup, we assume that the dataset consists of N frame of image sequence. J landmarks are present over all frames. At each frame $i \in \{1, \dots, N\}$, rigid motion is applied onto the 3D key points of the object and the 2D camera measurements under orthographic projection are represented as

$$\underbrace{\mathbf{p}_{ji}}_{2\times 1} = \underbrace{\mathbf{R}_i}_{2\times 3}\underbrace{\mathbf{s}_{ji}}_{3\times 1} + \underbrace{\mathbf{t}_i}_{2\times 1}, \tag{3.1}$$

where $\mathbf{s}_{ji} = [X_{ji}, Y_{ji}, Z_{ji}]^\top$ and $\mathbf{p}_{ji} = [x_{ji}, y_{ji}]^\top$ are the 3D coordinates and the 2D projection of point j at time i respectively. $\mathbf{R}_i \in \mathbb{R}^{2\times 3}$ is the orthonormal rotation matrix and $\mathbf{t}_i$ is the translation vector, which together comprise the motion field to be recovered. If the vectors of the J points are stacked up in rows, Equation (3.1) can be changed to

$$\underbrace{\mathbf{p}_i}_{2J\times 1} = \underbrace{\mathbf{G}_i}_{2J\times 3J}\underbrace{\mathbf{s}_i}_{3J\times 1} + \underbrace{\mathbf{T}_i}_{2J\times 1}, \tag{3.2}$$

where $\mathbf{R}_i$ is duplicated J times on the diagonal of $\mathbf{G}_i$. The objective of the NRSFM problem is to recover the 3D structure $\mathbf{s}_i$ as well as the camera motion $\{\mathbf{R}_i, \mathbf{T}_i\}$, which may be factorized from the 2D observation matrix, although there exists an infinite number of possibilities if no extra constraints are imposed. Figure 3.1 illustrates the main process of the NRSFM factorization [TK92, Bra05, TR05]. Note that in case of $\mathbf{s}_i$ being constant over all frames, this degenerates to the rigid Structure from Motion (SFM) study by Tomasi and Kanade [TK92].

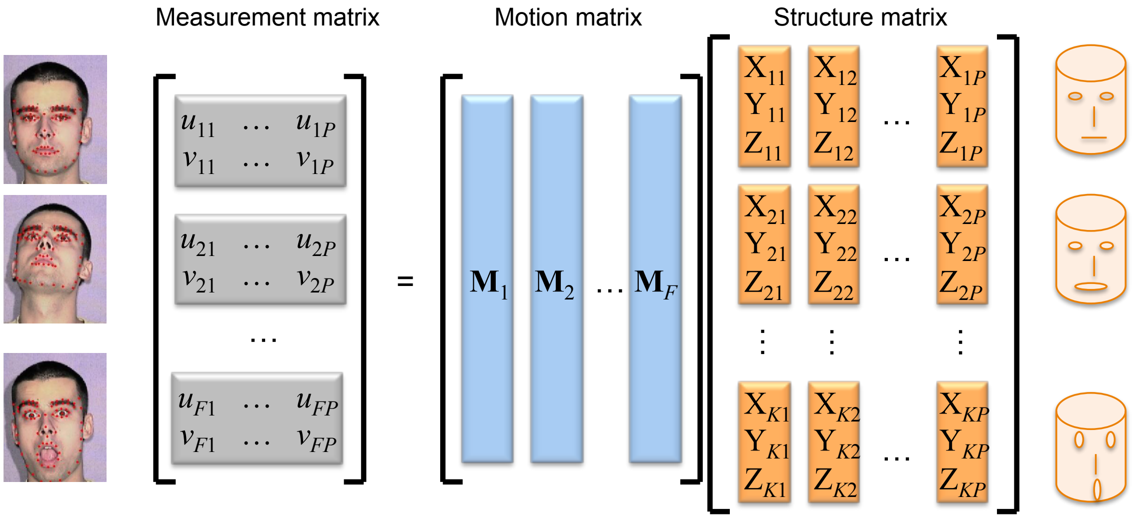

Measurement matrix
Motion matrix
Structure matrix
u_{11} … u_{1P}
v_{11} … v_{1P}
u_{21} … u_{2P}
v_{21} … v_{2P}
u_{F1} … u_{FP}
v_{F1} … v_{FP}
=
$\mathbf{M}_1$ $\mathbf{M}_2$ … $\mathbf{M}_F$
X_{11} Y_{11} Z_{11}
X_{12} Y_{12} Z_{12}
X_{1P} Y_{1P} Z_{1P}
X_{21} Y_{21} Z_{21}
X_{22} Y_{22} Z_{22}
X_{2P} Y_{2P} Z_{2P}
X_{K1} Y_{K1} Z_{K1}
X_{K2} Y_{K2} Z_{K2}
X_{KP} Y_{KP} Z_{KP}

Figure 3.1: Factorization technique for NRSFM. The left images are samples of input frames with landmarks. The goal is to factorize those 2D frames into correct 3D motion and structure matrices.

Previous researches [XCK06, THB08] have already shown that if the shape matrix $\mathbf{s}_i$ is deformed arbitrarily, the NRSFM problem is inherently underconstrained. Hence a shape model must be properly defined. Recently a common method to model shapes is used for 2D shape reconstruction [BV99, BB98] and computer graphics [Par72], which considers the 3D shape as a linear combination of a dominant rigid body [KsH05] and other deformation bases. Even many physical systems can be accurately represented [BJ05]. To be more specific on our case, the face model of a specific person may be modeled as a mean shape plus other bases representing facial expressions, blinking, talking etc. In our work, this widely accepted assumption is adopted. Let the $3J \times 1$ matrix $\bar{\mathbf{s}}$ be the mean shape, the $3J \times K$ matrix $\mathbf{V}$ and the K-dimensional vector $\mathbf{z}_i$ be the remaining basis shapes and their weights respectively, where K is the number of articulation shapes apart from the mean shape, the 3D shape of the ith frame is given as

$$\underbrace{\mathbf{s}_i}_{3J\times 1} = \underbrace{\bar{\mathbf{s}}}_{3J\times 1} + \underbrace{\mathbf{V}}_{3J\times K}\underbrace{\mathbf{z}_i}_{K\times 1}. \tag{3.3}$$

Note that shapes are stacked in matrix $\mathbf{V}$ so that each column represents a basis shape. With the above setup, if we align the images to the center and drop the translations, our NRSFM model is derived by combining Equation (3.2) and Equation (3.3) as

$$\mathbf{p}_i = \mathbf{R}_i(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_i). \tag{3.4}$$

Since the choice of the shape coefficients $\mathbf{z}_i$ determines the contribution of the shape bases and the recovery of the corresponding shape as well, it is studied by various of papers. An intuitive and popular approach is to embed them into the K-dimensional linear subspace. But in Section 3.3, we will show that we actually place a Gaussian prior onto the latent variables $\mathbf{z}_i$, which endows the linear subspace model with a probabilistic formulation.

3.2 Initialization

The nature of the EM algorithm introduced in Section 2.1 determines that the initial parameter values play a significant role on the speed of the convergence and the quality of the final estimates. In order not to get in a local maximum or minimum, a good and efficient initialization is desired. Because under the assumption of the shape subspace model in Section 3.1, the major contribution for shape modeling is the dominant mean shape component, which is rigid throughout the image sequence, the modified Tomasi-Kanade algorithm [TK92] for rigid SFM is used as initial motion and structure parameters in our work, which is initially employed by Torresani et al. in [THB04, THB08].

In [TK92], the rank theorem of rigid SFM is studied. With the help of the additional orthonormality constraints, the complete factorization algorithm of the 2D camera matrix is developed. The first step is to compute the Singular Value Decomposition (SVD) [GR70] of the registered measurement matrix $\widetilde{\mathbf{W}}$ with the mean of each row removed as

$$\widetilde{\mathbf{W}} = \mathbf{O}_1 \mathbf{\Sigma} \mathbf{O}_2.$$

Let $\mathbf{O}_1'$, $\mathbf{\Sigma}'$ and $\mathbf{O}_2'$ refer to the first three columns of $\mathbf{O}_1$, the first 3×3 submatrix of $\mathbf{\Sigma}$ and the first three rows of $\mathbf{O}_2$. Define

$$\hat{\mathbf{R}} = \mathbf{O}_1'(\mathbf{\Sigma}')^{\frac{1}{2}},$$
$$\hat{\mathbf{S}} = (\mathbf{\Sigma}')^{\frac{1}{2}}\mathbf{O}_2'.$$

Since there exists a non-degenerate 3×3 matrix $\mathbf{Q}$ so that the true rotation matrix $\mathbf{R}$ and the true shape matrix $\mathbf{S}$ are

$$\mathbf{R} = \hat{\mathbf{R}}\mathbf{Q},$$
$$\mathbf{S} = \mathbf{Q}^{-1}\hat{\mathbf{S}}.$$

Because the true rotation matrix $\mathbf{R}$ is subject to the orthonormality constraints, it results in the following over-constrained system:

$$\hat{\mathbf{i}}_f^\top \mathbf{Q}\mathbf{Q}^\top \hat{\mathbf{i}}_f = 1$$
$$\hat{\mathbf{j}}_f^\top \mathbf{Q}\mathbf{Q}^\top \hat{\mathbf{j}}_f = 1$$
$$\hat{\mathbf{i}}_f^\top \mathbf{Q}\mathbf{Q}^\top \hat{\mathbf{j}}_f = 0$$

In this equation, $\mathbf{i}_f$ and $\mathbf{j}_f$ are mutually orthogonal unit vectors that satisfy

$$|\mathbf{i}_f| = |\mathbf{j}_f| = 1$$

and

$$\mathbf{i}_f^\top \mathbf{j}_f = 0.$$

By applying the orthonormality constraints, the correct linear transformation matrix $\mathbf{Q}$ is found. Thus, the mean shape $\bar{\mathbf{S}}$ and the rigid motion $\mathbf{R}$ are successfully initialized.

To recover the remaining articulated shape bases $\mathbf{V}$, subtract $\widetilde{\mathbf{W}}$ with the mean shape and motion estimate

$$\widetilde{\mathbf{W}} = \mathbf{W} - \mathbf{R}\bar{\mathbf{S}}.$$

The residual is fitted separately at each frame i for $\mathbf{v}_{ki}$ so that

$$\mathbf{v}_{ki} = (\mathbf{R}_i)^{-1}\widetilde{\mathbf{W}}_i.$$

Finally, PCA is applied so that the first principal component of $\mathbf{v}_{ki}$ is selected for $\mathbf{V}_k$. The fitting process is proceeded iteratively for the remaining residual until the entire shape bases is initialized. The whole NRSFM initialization algorithm for our work is listed in Algorithm 3.1.

Algorithm 3.1 NRSFM initialization algorithm

```
1: Initialize the mean shape and rigid motion using [TK92].
     1(i). Compute the SVD of the registered measurement matrix $\widetilde{\mathbf{W}} = \mathbf{O}_1\mathbf{\Sigma}\mathbf{O}_2$.
     1(ii). Compute the pseudo shape and motion parameters $\hat{\mathbf{R}} = \mathbf{O}'_1(\mathbf{\Sigma}')^{\frac{1}{2}}$ and $\hat{\mathbf{S}} = (\mathbf{\Sigma}')^{\frac{1}{2}}\mathbf{O}'_2$ deviated by a linear transformation $\mathbf{Q}$.
     1(iii). Compute $\mathbf{Q}$ with the orthonormality constraints.
     1(iv). Obtain the true rotation matrix $\mathbf{R} = \hat{\mathbf{S}}\mathbf{Q}$ and shape matrix $\mathbf{S} = \mathbf{Q}^{-1}\hat{\mathbf{S}}$.
2: Compute the residual $\widetilde{\mathbf{W}} = \widetilde{\mathbf{W}} - \mathbf{R}\bar{\mathbf{S}}$.
3: for $k = 1$ to $K$ do
4:     for $i = 1$ to $N$ do
5:         Compute separately $\mathbf{v}_{ki} = (\mathbf{R}_i)^{-1}\widetilde{\mathbf{W}}_i$.
6:     end for
7:     Apply PCA to $\mathbf{v}_{ki}$ and select the first principal component as $\mathbf{V}_k$.
8: end for
```

3.3 PPCA Shape Model

In Section 3.1 the basic idea of defining the deformation shapes as a linear combination of the mean shape and other articulated bases. Although with computational convenience in mind, a K-dimensional linear subspace is endowed successfully in many applications [BV99, TP91], for solving the NRSFM problem, there are some noticeable limitations and drawbacks. For example, Xiao and Kanade [XK04] proved that even when imposing proper constraints, the linear model could still tend to degeneracy when there is bases not of full rank three. Another problem is that this approach is very sensitive to the manual selection of the number of shapes K. On the one hand, if K is set too small, the linear model cannot span the required space of the deformation so that not all variations of the real-world object can be represented. On the other hand, a large K could not only cause extra degrees of freedom with noise, but also the NRSFM problem becomes underconstrained with the increase of K. When $K = 2J$, the arbitrary articulation of all key points makes the NRSFM problem totally unconstrained. Torresani et al. [THB08] employs a PPCA deformation shape model with unknown priors, which is actually known as the hierarchical prior in Bayesian statistics [GCSR03]. The shape coefficients are assumed to come from a normally distributed probability function, while the exact parameters are not clear in prior. With those unobserved, or latent variables, the EM algorithm solves the maximum likelihood problems iteratively. During the iterations, one of the parameters of the distribution and the shape model is kept fixed and the other is fitted to maximize the likelihood alternately. This probabilistic framework generates the shape model very well on the fly. It also has an extraordinary ability to handle noisy data, therefore we adopt this implementation in our work.

PPCA introduced in Section 2.2.2 is a probabilistic enhancement for PCA, a well-established tool for exploratory data analysis, dimensionality reduction, factor analysis, etc. Here we use PPCA to describe the distribution over shapes. Remember that in Equation (3.3), we define $\mathbf{z}_i$ as the weights of the shape bases. In PPCA, we place a zero-mean Gaussian prior distribution on this latent variables

$$\mathbf{z}_i \sim \mathcal{N}(0; \mathbf{I}), \tag{3.5}$$

where the covariance matrix $\mathbf{I}$ is modeled with the iid assumption through the frames. Due to the inevitable presence of internal and external noise in image tracks and labeling, a zero-mean Gaussian noise term

$$\mathbf{n}_i \sim \mathcal{N}(0; \sigma^2 \mathbf{I})$$

with variance σ^2 is also added to the initial linear subspace modeled in Equation (3.4). So the new factorization of the 2D measurement matrix is as follows

$$\mathbf{p}_i = \mathbf{R}_i(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_i) + \mathbf{n}_i.$$

In terms of PPCA, the latent variables $\mathbf{z}_i$ is seen as the "projected" data points, which are marginalized out in the following EM algorithm. Correspondingly, the "sample" data points $\mathbf{p}_i$ is a linear combination of Gaussian distributed variables, and it is also a normal distribution. It is clear that the conditional distribution with respect to $\mathbf{z}_i$ gives

$$\mathbf{p}_i | \mathbf{z}_i \sim \mathcal{N}(\mathbf{R}_i(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_i); \sigma^2 \mathbf{I}).$$

Its exact distribution can be obtained by marginalizing over $\mathbf{z}_i$ on the "complete" dataset $\{\mathbf{p}_i, \mathbf{z}_i\}$ as

$$p(\mathbf{p}_i) = \int p(\mathbf{p}_i, \mathbf{z}_i)\, \mathrm{d}\,\mathbf{z}_i = \int p(\mathbf{p}_i | \mathbf{z}_i) p(\mathbf{z}_i)\, \mathrm{d}\,\mathbf{z}_i,$$

which yields

$$\mathbf{p}_i \sim \mathcal{N}(\mathbf{R}_i \bar{\mathbf{s}}; \mathbf{R}_i \mathbf{V}\mathbf{V}^\top \mathbf{R}_i^\top + \sigma^2 \mathbf{I}). \tag{3.6}$$

Using PPCA model, the problem of NRSFM is turned into the same as estimating the Gaussian distribution of the shape weights $\mathbf{z}_i$, while the motion and the non-rigid shapes are learned on the fly. In particular, the joint negative log-likelihood of $\{\mathbf{p}_i, \mathbf{z}_i\}$

$$\begin{aligned} \mathcal{L} = \frac{1}{2} \sum_i (\mathbf{p}_i - \mathbf{R}_i \bar{\mathbf{s}})^\top (\mathbf{R}_i \mathbf{V}\mathbf{V}^\top \mathbf{R}_i^\top + \sigma^2 \mathbf{I})(\mathbf{p}_i - \mathbf{R}_i \bar{\mathbf{s}}) && (3.7) \\ + \frac{1}{2} \sum_i \log |\mathbf{R}_i \mathbf{V}\mathbf{V}^\top \mathbf{R}_i^\top + \sigma^2 \mathbf{I}| + JT \log(2\pi) && (3.8) \end{aligned}$$

is maximized, which will be discussed in detail in the upcoming part. One of the most crucial assumptions we discussed before that make the NRSFM solvable is the non-arbitrariness of the shape deformation. The zero-mean unified Gaussian over the shape weights $\mathbf{z}_i$ actually makes the shape $\mathbf{s}_i$ at the ith each frame more or less confined to the dominant mean shape, which means, each pose is not unconstrained. Therefore no additional regularization terms are necessary.The other advantage of the model lies in that since the shape weights $\mathbf{z}_i$ are ultimately marginalized out in the EM iterations, the previous concern for overfitting with a large number of basis shapes K with a linear subspace model does not occur here.

A closed-form MLE optimization over the log-likelihood function in Equation (3.7) is not feasible regarding the latent variables and high dimensionality of the datasets. The

EM algorithm is a powerful tool to handle maximum likelihood problems with latent variables. In Section 2.1.2, we have given an introduction of its basic idea of iteratively estimating the likelihood with the data that is present and its applications in related applications e.g. factor analysis [GH96] and PPCA [Row98]. Our formulation of the specific EM algorithm is as follow: Since until now the probabilistic distribution of the measurement matrix $\mathbf{p}_i$ is given individually for each frame i, we need a joint distribution over all frames for the EM estimates $p(\mathbf{p}_{1:N})$, which may simply be a multiplication of the probability in every single frame in Equation (3.6) by

$$p(\mathbf{p}_{1:N}|\mathbf{R}_{1:N}, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2) = \prod_i p(\mathbf{p}_i|\mathbf{R}_i, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2).$$

EM is an iterative algorithm by alternating two phases: computing the distribution over the latent variables in the E-step, and updating the expected log likelihood function in the M-step.

E-step: In the first phase of the EM algorithm, we begin by obtaining the posterior distribution over $\mathbf{z}_i$ with respect to the old parameter estimates. Denote $q(\mathbf{z}_i)$ as this distribution and we get

$$\begin{aligned} q(\mathbf{z}_i) &= p(\mathbf{z}_i|\mathbf{p}_i, \mathbf{R}_i, \mathbf{T}_i, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2) \\ &= \mathcal{N}(\mathbf{z}_i|\boldsymbol{\beta}(\mathbf{p}_i - \mathbf{R}_i\bar{\mathbf{s}}); \mathbf{I} - \boldsymbol{\beta}\mathbf{R}_i\mathbf{V}), \end{aligned}$$

where $\boldsymbol{\beta}$ is in the form of

$$\boldsymbol{\beta} = \mathbf{V}^\top \mathbf{R}_i^\top (\mathbf{R}_i \mathbf{V}\mathbf{V}^\top \mathbf{R}_i^\top + \sigma^2 \mathbf{I})^{-1}. \tag{3.9}$$

Note that for computational efficiency, the matrix inversion lemma [Woo50]

$$(\mathbf{A} + \mathbf{UCV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1}$$

is employed, which reveals

$$\boldsymbol{\beta} = \mathbf{V}^\top \mathbf{R}_i^\top \left(\sigma^{-2}\mathbf{I} - \mathbf{R}_i\mathbf{V}(\mathbf{I} + \sigma^{-2}\mathbf{V}^\top\mathbf{R}_i^\top\mathbf{R}_i\mathbf{V})^{-1}\mathbf{V}^\top\mathbf{R}_i^\top\sigma^{-4}\right).$$

According to the posterior distribution $q(\mathbf{z}_i)$, the moments of the latent variables

$$\boldsymbol{\mu}_i \equiv \mathbb{E}[\mathbf{z_i}] = \boldsymbol{\beta}(\mathbf{p}_i - \mathbf{R}_i\bar{\mathbf{s}}) \tag{3.10}$$

$$\boldsymbol{\phi}_i \equiv \mathbb{E}[\mathbf{z}_i\mathbf{z}_i^\top] = \mathbf{I} - \boldsymbol{\beta}\mathbf{R}_i\mathbf{V} + \boldsymbol{\mu}_i\boldsymbol{\mu}_i^\top \tag{3.11}$$

are taken.

M-step: In the following M-step, the expected negative log-likelihood function

$$\begin{aligned} Q &\equiv \mathbb{E}[-\log p(\mathbf{p}_{1:N}|\mathbf{R}_{1:N}, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2)] \\ &= \mathbb{E}[-\sum_i \log p(\mathbf{p}_i|\mathbf{R}_i, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2)] \\ &= \frac{1}{2\sigma^2}\sum_i \mathbb{E}[||\mathbf{p}_i - \mathbf{R}_i(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z}_i)||^2] + JT\log(2\pi\sigma^2) \end{aligned} \tag{3.12}$$

is to be minimized and the shape and motion parameters are updated simultaneously. Note that this function may not be optimized in closed form, so the parameters are computed individually to make a better approximation for the log-likelihood function. Thus this is essentially a generalized EM algorithm.

Among the unknown parameters $\{\mathbf{R}_i, \bar{\mathbf{s}}, \mathbf{V}, \sigma^2\}$, the camera rotation matrix $\mathbf{R}_i$ is subject to the orthonormality constraints, which makes it impossible to have a closed-form solution. Aside from the rotation, the other three parameters can be solved directly. At first, the mean shape and the remaining shape bases can be recomposed together as a single matrix to make a more compact update. Accordingly, the shape weights vector $\mathbf{z}_i$ can be expanded from K rows to $K+1$ rows to insert the unit weight for the mean shape $\bar{\mathbf{s}}$:

$$\begin{aligned} \widetilde{\mathbf{V}} &\equiv [\bar{\mathbf{s}}, \mathbf{V}] \\ \tilde{\mathbf{z}}_i &\equiv [1, \mathbf{z}_i^\top]^\top \end{aligned}$$

On the basis of this modification, the first moment defined in Equation 3.10 is naturally changed to

$$\tilde{\boldsymbol{\mu}}_i \equiv [1, \boldsymbol{\mu}_i^\top]^\top,$$

while the second moment in Equation 3.10 can be approximated as

$$\tilde{\boldsymbol{\phi}}_i \equiv \begin{bmatrix} 1 & \boldsymbol{\mu}_i^\top \\ \boldsymbol{\mu}_i & \boldsymbol{\phi}_i \end{bmatrix}.$$

Using the formulae above, the expected negative log-likelihood function in Equation (3.12) becomes

$$Q = \frac{1}{2\sigma^2} \sum_i \mathbb{E}[||\mathbf{p}_i - \mathbf{R}_i \widetilde{\mathbf{V}} \tilde{\mathbf{z}}_i||^2] + JT \log(2\pi\sigma^2). \tag{3.13}$$

The updates for each parameter is done by solving for minimizing the value of Q with respect to itself, and holding the other parameters fixed. We start with the update of the shape bases $\widetilde{\mathbf{V}}$ by setting the partial derivative to zero

$$\begin{aligned} \frac{\partial Q}{\partial \widetilde{\mathbf{V}}} &= -\frac{1}{2\sigma^2} \sum_i \mathbb{E}[\mathbf{R}_i^\top (\mathbf{p}_i - \mathbf{R}_i \widetilde{\mathbf{V}} \tilde{\mathbf{z}}_i) \tilde{\mathbf{z}}_i^\top] \\ &= -\frac{1}{2\sigma^2} \sum_i \mathbf{R}_i^\top \mathbf{p}_i \tilde{\boldsymbol{\mu}}_i^\top + \frac{1}{2\sigma^2} \sum_i \mathbf{R}_i^\top \mathbf{R}_i \widetilde{\mathbf{V}} \tilde{\boldsymbol{\phi}}_i. \end{aligned}$$

By means of the vec and the Kronecker product $\otimes$, the following rule [Hor86] is given:

$$\operatorname{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A}) \operatorname{vec}(\mathbf{B})$$

By applying the vec operator to both sides of the equation, we have

$$\begin{aligned} \operatorname{vec} \frac{\partial Q}{\partial \widetilde{\mathbf{V}}} &= \frac{\partial Q}{\partial \operatorname{vec}(\widetilde{\mathbf{V}})} \\ &= -\frac{1}{2\sigma^2} \operatorname{vec} \left(\sum_i \mathbf{R}_i^\top \mathbf{p}_i \tilde{\boldsymbol{\mu}}_i^\top \right) + \frac{1}{2\sigma^2} \sum_i (\tilde{\boldsymbol{\phi}}_i^\top \otimes (\mathbf{R}_i^\top \mathbf{R}_i)) \operatorname{vec}(\widetilde{\mathbf{V}}). \end{aligned}$$

The stationary point is obtained by setting this partial derivative to zero, which yields the new shape bases

$$\operatorname{vec}(\widetilde{\mathbf{V}}) \leftarrow \left(\sum_i (\tilde{\boldsymbol{\phi}}_i^\top \otimes (\mathbf{R}_i^\top \mathbf{R}_i)) \right)^{-1} \operatorname{vec} \left(\sum_i \mathbf{R}_i^\top \mathbf{p}_i \tilde{\boldsymbol{\mu}}_i^\top \right).$$

The same approach can be applied to solve for the noise variance update by setting the partial derivative of Q with respect to σ^2

$$\frac{\partial Q}{\partial \sigma^2} = -\frac{1}{\sigma^3}\sum_i \mathbb{E}[||\mathbf{p}_i - \mathbf{R}_i\widetilde{\mathbf{V}}\tilde{\mathbf{z}}_i||^2] + \frac{2JT}{\sigma}$$

to zero, which gives

$$\begin{aligned}\sigma^2 &= \frac{1}{2JT}\sum_i \mathbb{E}[||\mathbf{p}_i - \mathbf{R}_i\widetilde{\mathbf{V}}\tilde{\mathbf{z}}_i||^2] \\ &= \frac{1}{2JT}\sum_i \left(||\mathbf{p}_i||^2 - 2\mathbf{p}_i^\top\mathbf{R}_i^\top\widetilde{\mathbf{V}}\tilde{\boldsymbol{\mu}}_i + \mathbb{E}[\tilde{\mathbf{z}}_i^\top\widetilde{\mathbf{V}}^\top\mathbf{R}_i^\top\mathbf{R}_i\widetilde{\mathbf{V}}\mathbf{z}_i]\right) \\ &= \frac{1}{2JT}\sum_i \left(||\mathbf{p}_i||^2 - 2\mathbf{p}_i^\top\mathbf{R}_i^\top\widetilde{\mathbf{V}}\tilde{\boldsymbol{\mu}}_i + \mathrm{tr}(\mathbf{V}^\top\mathbf{R}_i^\top\mathbf{R}_i\widetilde{\mathbf{V}}\mathbb{E}[\tilde{\mathbf{z}}_i\tilde{\mathbf{z}}_i^\top])\right),\end{aligned}$$

where the last term is derived by the fact that the expression is a scalar, and by the commutativity property of the tr function

$$\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$$

Then, the final noise variance update is

$$\sigma^2 \leftarrow \frac{1}{2JT}\sum_i \left(||\mathbf{p}_i||^2 - 2\mathbf{p}_i^\top\mathbf{R}_i^\top\widetilde{\mathbf{V}}\tilde{\boldsymbol{\mu}}_i + \mathrm{tr}(\mathbf{V}^\top\mathbf{R}_i^\top\mathbf{R}_i\widetilde{\mathbf{V}}\tilde{\boldsymbol{\phi}}_i)\right).$$

However, the camera rotation parameter $\mathbf{R}_i$ is subject to orthonormality constraints, hence closed-form update like the other parameters is not possible. In the initial paper, Torresani et al. [THB08] approximates the solution with a single Gauss-Newton step on the Euclidean space, which is inaccurate and has a theoretically low convergence rate. In Section 3.5, we propose our optimization technique on the manifold.

3.4 PRPCA Shape Model

In the last section, the NRSFM problem is modeled within a probabilistic framework. Our observation of the PPCA formulation is that with the iid assumption of the latent variables $\mathbf{z}$ for the weights of shape bases, some relational information between the frames with the same or similar deformations may be lost. For example, two consequent frames are more likely to have close relation in the weighs of the shape bases than the frames with a large time interval between them. So we present a probabilistic relational approach to the PPCA algorithm for solving the NRSFM problem.

Remember that in Section 3.3, the prior distribution of the latent variables $\mathbf{z}$ is given in Equation (3.5), where the covariance matrix is set as the identity matrix. Here we substitute $\mathbf{I}$ with the inverse of the relational matrix $\boldsymbol{\Delta}$ defined in Equation (2.15). In the EM iterations, the first moment in the E-step is the same as in Equation (3.10) according to Equation (2.20). The change lies in the second moment that subject to (2.21), the relational matrix in added to the initial PPCA Equation (3.11) so that

$$\boldsymbol{\phi}_i \equiv \mathbb{E}[\mathbf{z}_i\boldsymbol{\Delta}\mathbf{z}_i^\top] = \mathbf{I} - \boldsymbol{\beta}\mathbf{R}_i\mathbf{V} + \boldsymbol{\mu}_i\boldsymbol{\Delta}\boldsymbol{\mu}_i^\top,$$

where $\boldsymbol{\beta}$ is given by Equation (3.9). In the following M-step, based on the above estimates, the closed-form updates for the shape bases $\widetilde{\mathbf{V}}$ and noise variance σ^2 should also be modified to

$$\begin{aligned}\mathrm{vec}(\widetilde{\mathbf{V}}) &\leftarrow \left(\sum_i (\boldsymbol{\phi}_i^\top \otimes (\mathbf{R}_i^\top \mathbf{R}_i))\right)^{-1} \mathrm{vec}\left(\mathbf{R}^\top (\mathbf{p} - \mathbf{R}\bar{\mathbf{s}}) \boldsymbol{\Delta} \boldsymbol{\mu}^\top\right), \\ \sigma^2 &\leftarrow \frac{1}{2JT} \left(\sum_i \left(\|\mathbf{p}_i - \mathbf{R}_i \bar{\mathbf{s}}\|^2 + \mathrm{tr}(\mathbf{V}^\top \mathbf{R}_t^\top \mathbf{R}_i \mathbf{V} \boldsymbol{\phi}_i)\right)\right. \\ &\qquad \left. - \mathrm{tr}\left(2(\mathbf{p} - \mathbf{R}\bar{\mathbf{s}})^\top \mathbf{R} \mathbf{V} \boldsymbol{\mu} \boldsymbol{\Delta}\right)\right).\end{aligned}$$

In order to make the most of the PRPCA model, a reasonable relational matrix $\boldsymbol{\Delta}$ for our specific use remains to be found. Intuitively, the statistics of the geometric variation of the input image tracks should be exploited. Unfortunately, since various pose changes are present in the original image sequence, the analysis is not possible directly without further processing. The Point Distribution Model (PDM) algorithm, which is employed by Cootes et al. for Active Shape Model (ASM) [CTCG95] and Active Appearance Model (AAM) [CET98], is a powerful tool in computer vision to statistically study the shape of objects. It calculates the average positions and several aspects in which each sample tends to vary from the mean. PDM requires a set of landmark points to provide sufficient detail and identify the object precisely. In practice, our face datasets are labeled on the contour of the cheeks, eyes and noses, which is in general adequate for the algorithm and the geometry of the face shape is well approximated.

The first and most important step of the PDM algorithm is to align the training set so that the landmarks are positioned equivalently and the shapes are as closely related spatially as possible. The generalized Procrustes analysis [Gow75] aims to minimize the weighted squared error through the image sequence by scaling, rotating and translating the training shapes. Let

$$\mathbf{x}_i = [x_{i1}, y_{i1}, x_{i2}, y_{i2}, \ldots, x_{iN}, y_{iN}]^\top$$

denote the vector for ith shape consisting of N landmark points and

$$M(s, \theta)[\mathbf{x}] = \begin{bmatrix} s\cos(\theta)x_{i1} - s\sin(\theta)y_{i1} \\ s\sin(\theta)x_{i1} + s\cos(\theta)y_{i1} \\ \vdots \\ s\cos(\theta)x_{iN} - s\sin(\theta)y_{iN} \\ s\sin(\theta)x_{iN} + s\cos(\theta)y_{iN} \end{bmatrix}$$

be the operation of scaling by s times and rotating by θ. To start with, consider two shapes $\mathbf{x}_i$ and $\mathbf{x}_j$. If proper $M(s_j, \theta_j)$ and translation $[t_{xj}, t_{yj}]$ are applied onto the second shape $\mathbf{x}_j$, the weighted squared error is given by

$$\mathcal{E}_j = (\mathbf{x}_i - M(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j)^\top \mathbf{W} (\mathbf{x}_i - M(s_j, \theta_j)[\mathbf{x}_j] - \mathbf{t}_j),$$

where the diagonal of the matrix $\mathbf{W}$ is made of the weights for each point. It is utilized to give more emphasis to the the points which are more "stable" over the frames. That means, those points which move less in comparison with other points gain higher weight. Thus, a distance matrix R_{kl} is defined to hold the distance between the kth point and the lth point in a single shape. The variance of the distance R_{kl} over all shapes in the image sequence $V_{R_{kl}}$ should mean the inverse of the weights according to the above assumption. Then the weight w_k for the kth point is

$$w_k = \left(\sum_{l=1}^{N} V_{R_{kl}} \right)^{-1}.$$

This weight formulation is rational, because if the distance of a certain point to the others remains unchanged or changes not much for all images, the variance of the distance tends to be small. Therefore the weight is set to a large value and vice versa. At the end, least squares approach is used and the resulting linear equations are solved for the alignment parameters. The generalized Procrustes alignment algorithm is run several iterations until preset threshold value is reached. Algorithm 3.2 summarizes the alignment approach.

Algorithm 3.2 Generalized Procrustes algorithm for image alignment

```
1: Align each shape with the first one in the dataset with rotation, scaling and translation.
2: repeat
3:     Compute the mean shape of the aligned shapes.
4:     Normalize the current mean.
5:     Align each shape with the mean shape.
6: until Convergence.
```

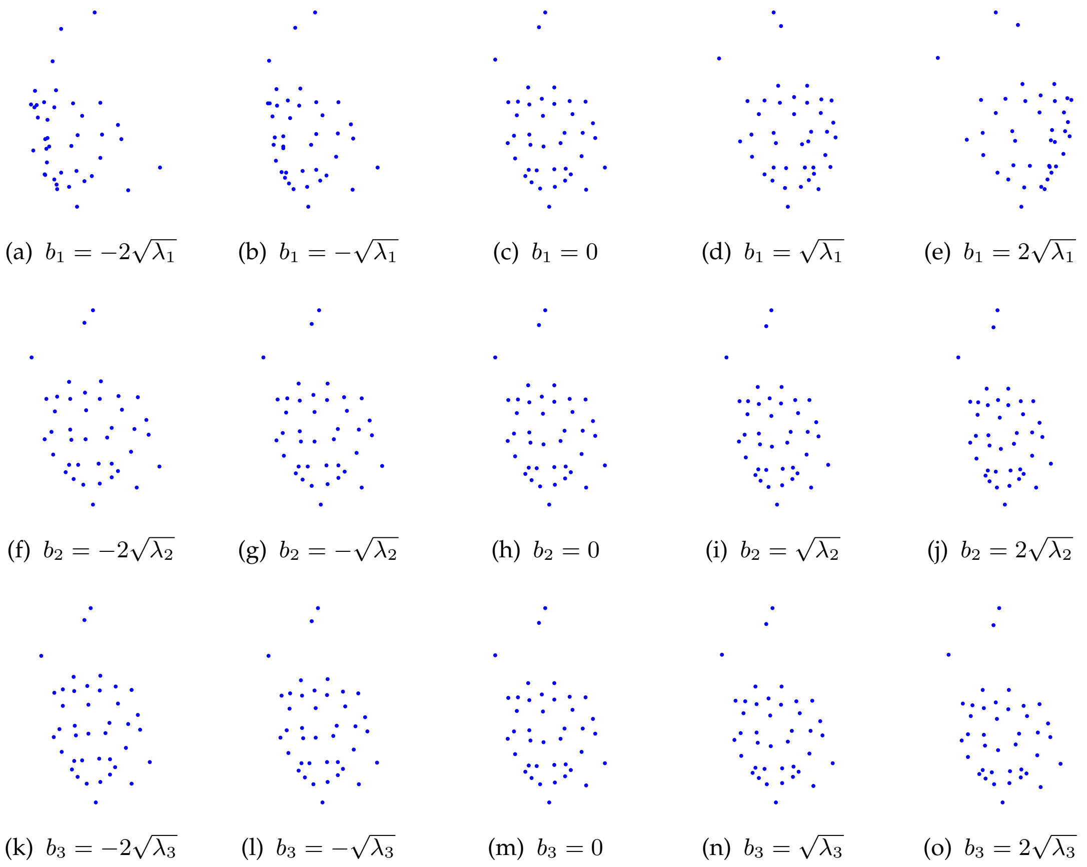
After the alignment process, all shapes are normalized and aligned to the centroid. Assuming the scattering is a Gaussian in the space, PCA is capable of retrieving the variations of the shapes and the eigenvectors and eigenvalues of the covariance matrix is obtained. Each principal axis represents a mode of variation and the corresponding variance σ_i^2 is given by the eigenvalue λ_i. From Section 2.2.1, we know that largest eigenvectors conform to the longest axes of the Gaussian ellipsoid and any shape $\mathbf{x}$ can be approximated by the first few eigenvectors if they are sorted in descending order:

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{i=1}^{D} b_i \mathbf{p}_i \tag{3.14}$$

In the equation, $\bar{\mathbf{x}}$ is the aligned mean shape, while $\mathbf{p}_i$ and b_i are the ith eigenvector and its weight. Because the eigenvectors obtained by PCA is linearly independent, Equation (3.14) is able to generate new shapes within the span subspace, if suitable ranges of the weights b_i are selected. Empirically, it is considered to be safe to allow the weights to vary within ±3 standard deviations

$$-3\sqrt{\lambda_i} \leq b_i \leq 3\sqrt{\lambda_i}.$$

A descriptive example of tuning the PCA parameters is illustrated in Figure 3.2 for our Vicon face dataset described in Section 4.2.1. In this plot, the largest three eigenvalues and the corresponding eigenvectors are analyzed and the shape variation is ranged from -2 to $+2$ times of the standard deviation. Note that in the middle column, without any variation in pose and expression, all five plots represent the mean shape of the dataset. In the first row, it is clear to figure out that this principal axis describes the pose change referring to the horizontal movements, which is totally understandable because even without doing alignment and PCA, the panning action of the head is easy to recognize as the



(a) $b_1 = -2\sqrt{\lambda_1}$ (b) $b_1 = -\sqrt{\lambda_1}$ (c) $b_1 = 0$ (d) $b_1 = \sqrt{\lambda_1}$ (e) $b_1 = 2\sqrt{\lambda_1}$

(f) $b_2 = -2\sqrt{\lambda_2}$ (g) $b_2 = -\sqrt{\lambda_2}$ (h) $b_2 = 0$ (i) $b_2 = \sqrt{\lambda_2}$ (j) $b_2 = 2\sqrt{\lambda_2}$

(k) $b_3 = -2\sqrt{\lambda_3}$ (l) $b_3 = -\sqrt{\lambda_3}$ (m) $b_3 = 0$ (n) $b_3 = \sqrt{\lambda_3}$ (o) $b_3 = 2\sqrt{\lambda_3}$

Figure 3.2: PDM result on the Vicon dataset. The effects of the first three eigenvectors applied to the mean shape are illustrated in each row. From left to right, the weight increases from -2 to $+2$ times of the standard deviation with respect to the eigenvalue.

largest variation of the 2D camera data. The second and the third eigenvectors control the shape variations. The face model becomes wider when negative weight of the second component is added, and vice versa. The third row consists of the expression of opening and closing the mouth. As a continuous action of the mouth movement, eyes and eyebrows naturally moves up and down when the mouth opens and closes, which is also well modeled by PDM. We observe that starting from the fourth PCA axis, no more valuable shape information is given and the model starts to fit noise. The normalized sum of the eigenvalues also suggest that the first three eigenvalues add up to more than 95% of the total sum.

After all, by applying the PDM to our face datasets, a set of D-dimensional eigenvectors are computed, which are orthogonal and linearly independent. So the 2D shape in each frame can be approximately represented by the feature vector $\mathbf{b}$ of the coefficients b_i using Equation (3.14). Since the ordered eigenvalues of the eigenvectors $\mathbf{p}_i$ fall off very quickly, b_i should be normalized to eliminate the difference of the magnitudes. We implement this normalization by dividing each b_i with the variance of the corresponding eigenvectors, which reveals a new feature vector $\mathbf{c}$. Based on the observation in Figure 3.2 that the first one or two eigenvectors normally indicate pan or tilt of the head orientation, which are irrelevant to the face shape, those coefficients are excluded from $\mathbf{c}$. The symmetric matrix

$\mathbf{A}$ of PRPCA is then obtained by calculating the framewise distance of $\mathbf{c}_i$ and $\mathbf{c}_j$:

$$\mathbf{A}_{ij} = ||\mathbf{c}_i - \mathbf{c}_j||_2 = \sqrt{\sum_{k=1}^{N} |\mathbf{c}_{i_k} - \mathbf{c}_{j_k}|^2} \tag{3.15}$$

Finally, the relational matrix $\mathbf{\Delta}$ is given by Equation (2.15) with γ typically being a very small positive value to guarantee the positive definiteness of $\mathbf{\Delta}$.

In practice, PDM may also be used to determine the the number of the non-rigid shapes. This is achieved by setting a threshold of the sum of the sorted normalized eigenvalues in the PCA process. The number of the selected eigenvalues is considered to be a reliable guess for the number of the shape bases.

3.5 Rotation Update on Manifold

In Torresani et al.'s approach [THB08] of solving the NRSFM problem with PPCA, the partial derivative of the negative log-likelihood Q in Equation (3.13) with respect to the rotation $\mathbf{R}$

$$\mathrm{vec}\,\frac{\partial Q}{\partial \mathbf{R}} \approx \mathbf{A}\,\mathrm{vec}(\hat{\boldsymbol{\xi}}) + \mathbf{B},$$

where $\mathrm{vec}(\hat{\boldsymbol{\xi}})$ is the twist vector for the equation and $\hat{\boldsymbol{\xi}}$ is the skew-symmetric matrix in the form of

$$\hat{\boldsymbol{\xi}} = \begin{bmatrix} 0 & -\xi_3 & \xi_2 \\ \xi_3 & 0 & -\xi_1 \\ -\xi_2 & \xi_1 & 0 \end{bmatrix}.$$

Minimizing

$$||\mathbf{A}\,\mathrm{vec}(\hat{\boldsymbol{\xi}}) + \mathbf{B}||_F$$

with respect to $\boldsymbol{\xi}$ reveals

$$\mathrm{vec}(\hat{\boldsymbol{\xi}}) \leftarrow -\mathbf{A}^{+}\mathbf{B}.$$

Note that the operator $\mathbf{A}^{+}$ denotes the Moore–Penrose pseudoinverse of $\mathbf{A}$. In the presence of the orthonormality constraints, the incremental rotation update must be performed by means of the exponential map with the skew-symmetric matrix $\hat{\boldsymbol{\xi}}$. Expansion of Taylor series yields

$$\mathbf{\Delta} = \exp(\boldsymbol{\xi}) = \mathbf{I} + \hat{\boldsymbol{\xi}} + \frac{\hat{\boldsymbol{\xi}}^2}{2!} + \dots .$$

Thus the new rotation matrix is obtained by dropping the nonlinear terms as

$$\mathbf{R}_{\text{new}} = (\mathbf{I} + \hat{\boldsymbol{\xi}})\mathbf{R}.$$

We notice that the exponential map of the skew-symmetric matrix $\hat{\boldsymbol{\xi}}$ is employed to form the step of a single Gauss-Newton step. Note that without defining an appropriate metric on the manifold, a manually selected and fixed updating step length is implemented, which declines the performance obviously when faced complex setups. Moreover, the Gauss-Newton optimization has a theoretically low convergence rate.

The rotation matrix $\mathbf{R}$ is orthogonal matrix with determinant 1, which lies exactly on the manifold of the special orthogonal group $SO(3)$ defined in Equation (2.23). Hence instead of trying to put an approximate algebraic or numeric constraint on the Euclidean space $\mathbb{R}^N$ and projecting them back onto the $SO(3)$ manifold, an unconstrained optimization on the manifold is a natural generalization and is expected to perform better. In Section 2.3, we have already introduced the fundamental of the canonical Riemannian structure of those orthogonal manifolds in order to generalize a Riemannian Newton method on them. Remember that besides the gradient and Hessian, definition of the update along the geodesic of the manifold must be known to ensure that the update is valid, because unlike on the Euclidean space, update path is no longer a straight line but rather a geodesic curve, which stays on the surface of the manifold all the time.

We define the objective function F with respect to the rotation matrix $\mathbf{R}$ for the manifold optimization as follows

$$F(\mathbf{R}) = \mathbb{E}[||\mathbf{p} - \mathbf{R}(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z})||_F^2].$$

Note that in our NRSFM formulation in Equation (3.4), the camera rotation matrix $\mathbf{R}$ is in fact a 2×3 matrix. Fortunately, we can parameterize it by multiplying a 2×3 identity projection matrix

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

so that

$$\mathbf{R}^{2\times 3} = \mathbf{\Pi}\mathbf{R}^{3\times 3}.$$

In this way, the special orthogonal group $SO(3)$ is still applicable to our camera rotation parameter.

Since $\mathbf{R} \in SO(3)$, its tangent vector $\mathbf{\Delta} \in T(SO(3))$ is given by

$$\mathbf{\Delta} = \mathbf{R}\hat{\mathbf{u}},$$

where $\hat{\mathbf{u}}$ is the skew-symmetric matrix of vector $\mathbf{u}$. For the Riemannian manifold, the canonical metric can simply be induced from the Euclidean metric as

$$g_c(\mathbf{\Delta}, \mathbf{\Delta}) = \frac{1}{2}\operatorname{tr}(\mathbf{\Delta}^\top \mathbf{\Delta}).$$

The explicit formula for geodesics [MKS99] on $SO(3)$ at $\mathbf{R}$ in direction $\mathbf{\Delta}$ is then

$$\begin{aligned} \mathbf{R}(t) &= \exp(\mathbf{R}, \mathbf{\Delta}t) \\ &= \mathbf{R}\exp(\hat{\boldsymbol{\omega}}t) \\ &= \mathbf{R}\left(\mathbf{I} + \hat{\boldsymbol{\omega}}\sin(t) + \hat{\boldsymbol{\omega}}^2(1 - \cos(t))\right), \end{aligned}$$

where $t \in \mathbb{R}$, $\omega = \mathbf{R}^\top\mathbf{\Delta} \in \mathfrak{so}(3)$ and $\mathfrak{so}(3)$ is the Lie algebra associated with the $SO(3)$ group. The last equation is called the Rodrigues' rotation formula [MSZ94].

To obtain the gradient and Hessian, we first derive the first and second order derivative for the geodesic $\mathbf{R}(t)$ with respect to t:

$$\begin{aligned}
\left.\frac{\mathrm{d}\,\mathbf{R}(t)}{\mathrm{d}\,t}\right|_{t=0} &= \left.\mathbf{R}\hat{\boldsymbol{\omega}}\cos(t) + \mathbf{R}\hat{\boldsymbol{\omega}}^2\sin(t)\right|_{t=0} \\
&= \mathbf{R}\hat{\boldsymbol{\omega}} \\
&= \mathbf{R}(\mathbf{R}^\top\boldsymbol{\Delta}) \\
&= \boldsymbol{\Delta} \\
\left.\frac{\mathrm{d}^2\,\mathbf{R}(t)}{\mathrm{d}\,t^2}\right|_{t=0} &= \left.-\mathbf{R}\hat{\boldsymbol{\omega}}\sin(t) + \mathbf{R}\hat{\boldsymbol{\omega}}^2\cos(t)\right|_{t=0} \\
&= \mathbf{R}\hat{\boldsymbol{\omega}}^2 \\
&= \mathbf{R}(\mathbf{R}^\top\boldsymbol{\Delta})(\mathbf{R}^\top\boldsymbol{\Delta}) \\
&= \boldsymbol{\Delta}(\mathbf{R}^\top\boldsymbol{\Delta}) \\
&= -\boldsymbol{\Delta}\boldsymbol{\Delta}^\top\mathbf{R}
\end{aligned}$$

Note that the last equation is derived from the property of tangent space on the Stiefel manifold in Equation (2.24) that $\mathbf{R}^\top\boldsymbol{\Delta}$ is a skew-symmetric matrix. The gradient and Hessian in direction $\boldsymbol{\Delta} \in T(SO(3))$ can be derived given the geodesic definition of a function in Equation (2.26) and Equation (2.27) and the estimates in the M-step of PPCA in Equation (3.10) and Equation (3.11) of Section 3.3:

$$\begin{aligned}
\mathrm{d}\,F(\boldsymbol{\Delta}) &= \left.\frac{\mathrm{d}\,F(\mathbf{R}(t))}{\mathrm{d}\,t}\right|_{t=0} \\
&= \mathbf{R}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\dot{\mathbf{R}}^\top - \mathbf{p}\boldsymbol{\mu}^\top\mathbf{V}^\top\dot{\mathbf{R}}^\top|_{t=0} \\
&= \mathbf{R}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\boldsymbol{\Delta}^\top - \mathbf{p}\boldsymbol{\mu}^\top\mathbf{V}^\top\boldsymbol{\Delta}^\top \\
\operatorname{Hess}F(\boldsymbol{\Delta},\boldsymbol{\Delta}) &= \left.\frac{\mathrm{d}^2\,F(\mathbf{R}(t))}{\mathrm{d}\,t^2}\right|_{t=0} \\
&= \dot{\mathbf{R}}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\dot{\mathbf{R}}^\top + \mathbf{R}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\ddot{\mathbf{R}}^\top - \mathbf{p}\boldsymbol{\mu}^\top\mathbf{V}^\top\ddot{\mathbf{R}}^\top|_{t=0} \\
&= \boldsymbol{\Delta}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\boldsymbol{\Delta}^\top - \mathbf{R}\mathbf{V}\boldsymbol{\phi}\mathbf{V}^\top\mathbf{R}^\top\boldsymbol{\Delta}\boldsymbol{\Delta}^\top + \mathbf{p}\boldsymbol{\mu}^\top\mathbf{V}^\top\mathbf{R}^\top\boldsymbol{\Delta}\boldsymbol{\Delta}^\top
\end{aligned}$$

For any arbitrary pair of vectors $\mathbf{X}, \mathbf{Y} \in T(SO(3))$, polarization helps compute $\operatorname{Hess}F(\mathbf{X},\mathbf{Y})$ with

$$\begin{aligned}
\operatorname{Hess}F(\mathbf{X},\mathbf{Y}) &= \frac{1}{4}\left(\operatorname{Hess}F(\mathbf{X}+\mathbf{Y},\mathbf{X}+\mathbf{Y}) - \operatorname{Hess}F(\mathbf{X}-\mathbf{Y},\mathbf{X}-\mathbf{Y})\right) \\
&= \frac{1}{2}\Big(\mathbf{X}\mathbf{V}\phi\mathbf{V}^\top\mathbf{Y}^\top + \mathbf{Y}\mathbf{V}\phi\mathbf{V}^\top\mathbf{X}^\top \\
&\quad - \mathbf{R}\mathbf{V}\phi\mathbf{V}^\top\mathbf{R}^\top\mathbf{X}\mathbf{Y}^\top - \mathbf{R}\mathbf{V}\phi\mathbf{V}^\top\mathbf{R}^\top\mathbf{Y}\mathbf{X}^\top \\
&\quad + \mathbf{p}\mu^\top\mathbf{V}^\top\mathbf{R}^\top\mathbf{X}\mathbf{Y}^\top + \mathbf{p}\mu^\top\mathbf{V}^\top\mathbf{R}^\top\mathbf{Y}\mathbf{X}^\top\Big).
\end{aligned}$$

With the requirements for generalizing Newton's method being ready, the optimal updating vector on the manifold can be found by modifying the original Newton Equation (2.22) to

$$\boldsymbol{\Delta} = -\operatorname{Hess}^{-1}\mathbf{G},$$

assuming that the Hessian is non-degenerate. It is the same as finding a vector $\boldsymbol{\Delta}$ that satisfies for all vector fields $\mathbf{Y}$

$$\operatorname{Hess}F(\mathbf{Y},\boldsymbol{\Delta}) = g_c(-\mathbf{G},\mathbf{Y}) = -\mathrm{d}\,F(\mathbf{Y}),$$

where $\mathbf{G} = \nabla F$ stands for the gradient. The Hessian can be uniquely determined by using an orthonormal basis $\{\mathbf{E}^k\}, k = 1, 2, 3$ into the equation above as

$$\operatorname{Hess} F(\mathbf{E}^k, \mathbf{\Delta}) = -\mathrm{d}\, F(\mathbf{E}^k).$$

For simplicity, the standard basis $\mathbf{e}_k$ for $\mathbb{R}^3$ is chosen so that $\mathbf{E}^k = \mathbf{R}\hat{\mathbf{e}}_k \in T(SO(3))$. Thus, the 3×3 Hessian matrix $\mathbf{H}$ and the three-dimensional gradient vector $\mathbf{g}$ can be obtained:

$$\begin{aligned} \mathbf{H}_{kl} &= \operatorname{Hess} F(\mathbf{E}^k, \mathbf{E}^l), \\ \mathbf{g}_k &= \mathrm{d}\, F(\mathbf{E}^k), \ \ k, l = 1, 2, 3 \end{aligned}$$

Then solving for the vector $\mathbf{u} = [u_1, u_2, u_3]^\top \in \mathbb{R}^3$ using

$$\mathbf{u} = -\mathbf{H}^{-1}\mathbf{g}.$$

Finally, the desired updating vector $\mathbf{\Delta} = \mathbf{R}\hat{\mathbf{u}}$ is done. The last step is to update the current rotation along the geodesic in the direction of this vector. The algorithm is summarized in Algorithm 3.3.

Algorithm 3.3 Minimization for the objective function $F(\mathbf{R}) = \mathbb{E}[||\mathbf{p} - \mathbf{R}(\bar{\mathbf{s}} + \mathbf{V}\mathbf{z})||_F^2]$

```
1: At the point $\mathbf{R} \in SO(3)$, compute the optimal updating vector $\mathbf{\Delta} = -\operatorname{Hess}^{-1}\mathbf{G}$.
    1(i). Choose basis tangent vectors $\mathbf{E}^k = \mathbf{R}\hat{\mathbf{e}}_k \in T(SO(3))$, with $\mathbf{e}_k$ for $1 \le k \le 3$ the standard basis for $\mathbb{R}^3$.
    1(ii). Compute the $3 \times 3$ matrix $\mathbf{H}_{kl} = \operatorname{Hess} F(\mathbf{E}^k, \mathbf{E}^l)$, $1 \le k, l \le 3$.
    1(iii). Compute the three-dimensional vector $\mathbf{g}_k = \mathrm{d}\, F(\mathbf{E}^k)$, $1 \le k \le 3$.
    1(iv). Compute the vector $\mathbf{u} = (u_1, u_2, u_3)^\top \in \mathbb{R}^3$ such that $\mathbf{u} = -\mathbf{H}^{-1}\mathbf{g}$.
    1(v). The optimal updating vector $\mathbf{\Delta} = -\operatorname{Hess}^{-1}\mathbf{G} = \mathbf{R}\hat{\mathbf{u}}$.
2: Update the rotation $\mathbf{R}$.
    2(i). Move $\mathbf{R}$ in the direction $\mathbf{\Delta}$ along the geodesic to
          $\exp(\mathbf{R}, \mathbf{\Delta}t) = \mathbf{R}\left(\mathbf{I} + \hat{\boldsymbol{\omega}}\sin(t) + \hat{\boldsymbol{\omega}}^2\,(1 - \cos(t))\right)$,
          where $t = \sqrt{\frac{1}{2}\operatorname{tr}(\mathbf{\Delta}^\top\mathbf{\Delta})}$ and $\boldsymbol{\omega} = \frac{\mathbf{R}^\top\mathbf{\Delta}}{t}$.
```

4. Experiments

This chapter demonstrates the performance of our NRSFM algorithm with the experiments conducted on the real-world face datasets, which are described first in this chapter. Thereafter, results of PDM for relational matrix construction are given. In the following sections, studies on different numbers of non-rigid shapes are presented. In order to test the robustness of our algorithm and the state-of-the-art algorithm, Gaussian noise is manually added to the original data with various noise levels.

4.1 Experimental Setup

Three types of experiments are conducted on real-world face datasets to qualitatively and quantitatively evaluate the performance of our work in comparison with the state-of-the-art algorithm proposed by Torresani et al. in [THB08]. In the first experiment, generalized Procrustes analysis is applied to the 2D camera measurement data to first align the faces from different pose changes, and to obtain the statistical relations between each frame using PCA. A number of tests are made based on the number of deformation shapes K to see the choice of K has how much impact on the overall performance. Another experiment scenario is to understand the robustness of the 3D reconstruction in the presence of noise.

We compare the following models and algorithms overall:

- **PPCA**. The baseline algorithm based on PPCA using the EM algorithm proposed by Torresani et al. in [THB08], which is described in Section 3.3.
- **PRPCA**. Our extended PPCA algorithm embedded with the relational shape information described in Section 3.4 to embed relational information between the frames.
- **Manifold PPCA**. Our solution of the orthonormality constraints with the generalized Newton's method on Manifold, which is described in Section 3.5.

In the remainder of this chapter, **PPCA**, **PRPCA** and **Manifold PPCA** are called to represent the approaches above.

Because the EM algorithm is employed in PPCA to maximize the data likelihood, no training phase is required in prior and the face model is learned on the fly. After aligning the image frames to the centroid, a total of 50 EM iterations are run and the relevant parameters for shape and motion recovery are computed. Torresani et al. [THB08] found that when the noise variance σ^2 is forced to remain a large value in the initial EM iterations, the NRSFM algorithm is more likely to converge to a better solution. An annealing parameter is applied to σ^2 for iterations in range of $1 \leq n \leq \frac{N}{2}$ so that

$$\sigma^2 \leftarrow \sigma^2 \left(1 + N \left(1 - \frac{n}{\frac{N}{2}}\right)\right),$$

where n is the current iteration count and N is the total iteration number. Thus, the annealing parameter decreases from a relatively large value in the first iteration and reaches 1 at the middle of the EM iterations. We also adopt this tweak and keep the setup of our PRPCA and Manifold PPCA algorithms the same as the baseline algorithm. All the algorithms in this work are implemented using the numerical computing environment MATLAB R2010a [Mat]. The PPCA source code is provided by Torresani et al. [THBa].

Our evaluation criteria is the same as in the previous papers, i.e. the sum of squared differences between estimated 3D shapes to ground truth depth with camera rotation also being applied

$$||\hat{\mathbf{s}}_{1:T} - \mathbf{s}_{1:T}||_F^2,$$

where the lowering F denotes the Frobenius norm [GVL96]:

$$||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij}|^2}$$

In the experiments with posterior zero-mean Gaussian additive noise added, the noise level is plotted as the ratio of the noise variance to the norm of the 2D measurements:

$$\frac{JT\sigma^2}{||\mathbf{p}_{1:T}||_F}.$$

The noise levels range from 0% to 30% with 2% step and the trials for each level of noise were averaged over 10 runs.

4.2 Experimental Data

Experiments are conducted on two different real-world face datasets. They are described in detail in this section.

4.2.1 Vicon Dataset

The first dataset is made public available for evaluation purposes by Torresani et al. [THBb], which is first employed in [THB08]. The image sequence is captured with a Vicon optical motion capture system. This dataset contains only a single subject with 40 markers attached to the face. The 3D position of the markers are estimated using triangulation. In the totally 316 frames long sequence sampled at 15 Hz, the subject made a limited range of facial expressions and head pose changes. Note that the tracking is very accurate using the markers with little noise. An illustration of he mean shape and the other deformation shapes applied to the mean shape is shown in Figure 4.1. Note that the lines are not present in the original dataset. They are shown for the sake of a better visualization. The mean shape represents the neutral expression, whereas the deformation bases model the open and close actions of the mouse and eyes.

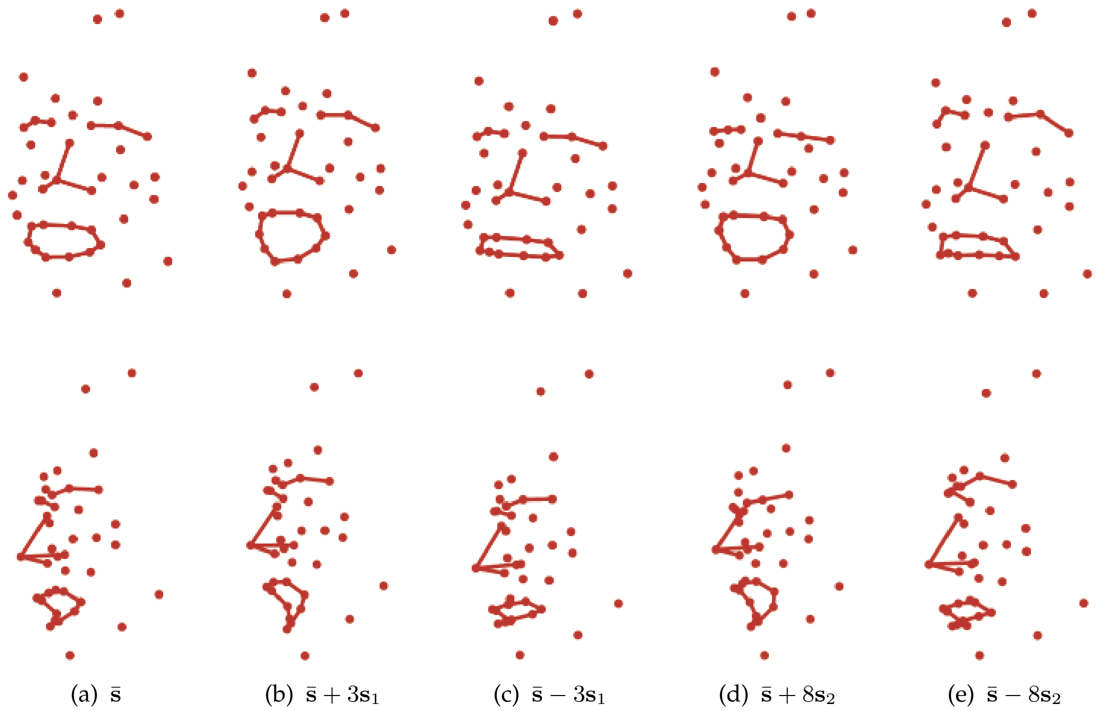

(a) $\bar{s}$ (b) $\bar{s} + 3s_1$ (c) $\bar{s} - 3s_1$ (d) $\bar{s} + 8s_2$ (e) $\bar{s} - 8s_2$

Figure 4.1: The Vicon face dataset with the first two shape bases applied to the mean shape. [THB08]

4.2.2 BU-3DFE Dataset

The second dataset is a subset of the BU-3DFE dataset, which is created by the Binghamton University for 3D facial expression analysis. The complete dataset is very large, consisting of 100 subjects, 56 females and 44 males, covering a wide age range and different ethnic groups. Seven expressions, neutral, happiness, disgust, fear, angry, surprise and sadness, are performed by each subject. Frontal-view textures are also provided. We randomly select 300 frames for our test. Since only 3D face feature points are present, random pose changes are applied and projected to the 2D input data. Note that the hand-labeled 83 marker points make noticeable noise in the original measurements. The purpose of this test is to learn how good universal face model can be generated using NRSFM algorithms. Four sample subjects of the dataset are shown in Figure 4.2. In the first column, the 83 feature points are labeled with white dots.

4.3 Experimental Results

This section presents a detailed description of the experiments. Possible causes of performance improvements and decreases are analyzed.

4.3.1 Results of Relational Information

The goal of this experiment is to assess how PDM works with various pose and facial expression variations and to give an illustration of the constructed relational matrix for PRPCA. The experiment is done on both of our face datasets. PDM is carried out directly on the 2D camera measurements. After aligning the feature points to be able to compare equivalent points from different frames, the statistical relational information is extracted by PCA. A descriptive introduction of this approach is given in Section 3.4.

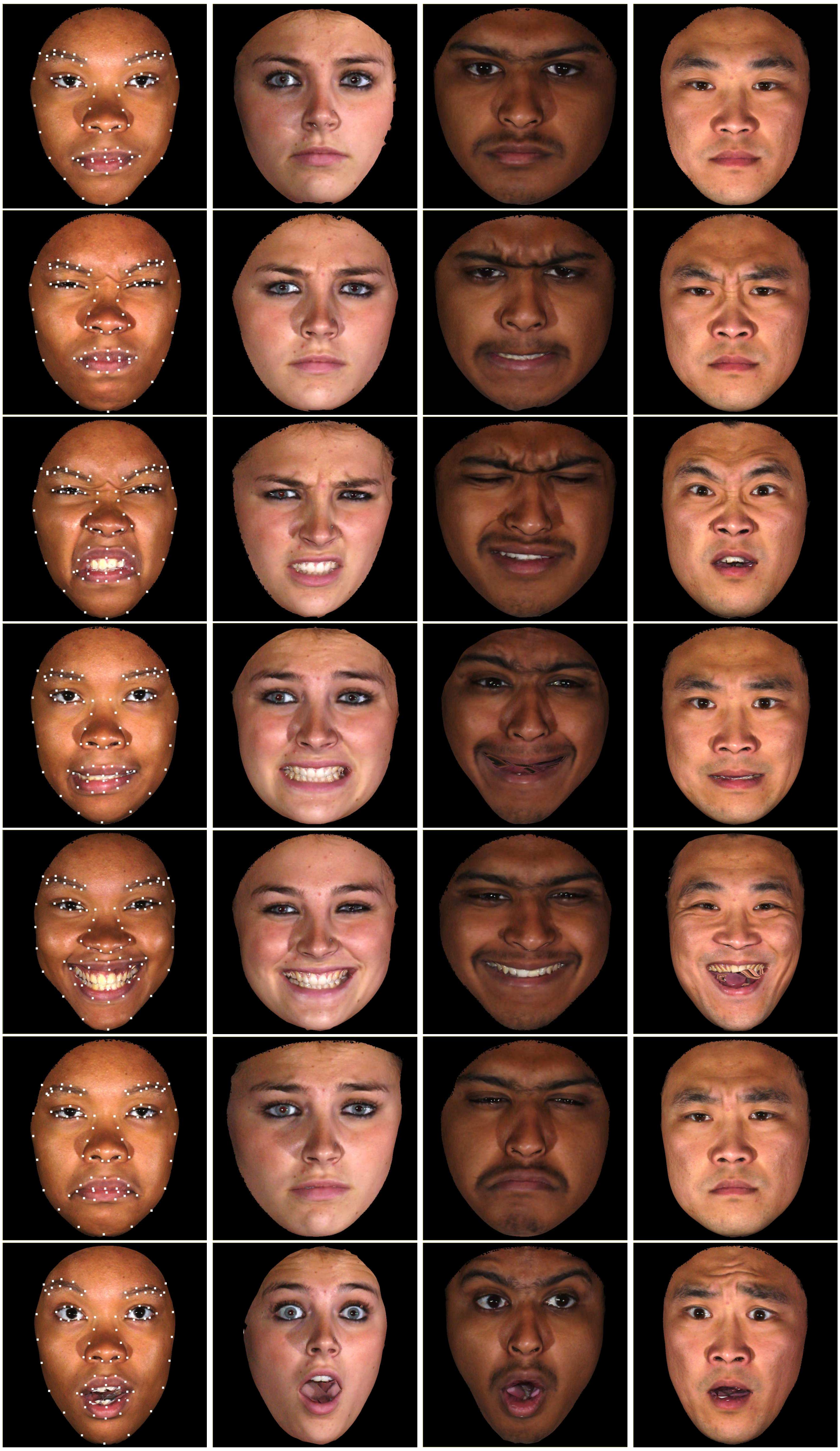


Figure 4.2: Four sample subjects of the BU-3DFE dataset showing various expressions. In the first column, the 83 feature points are marked with white dots. [YWS+06]

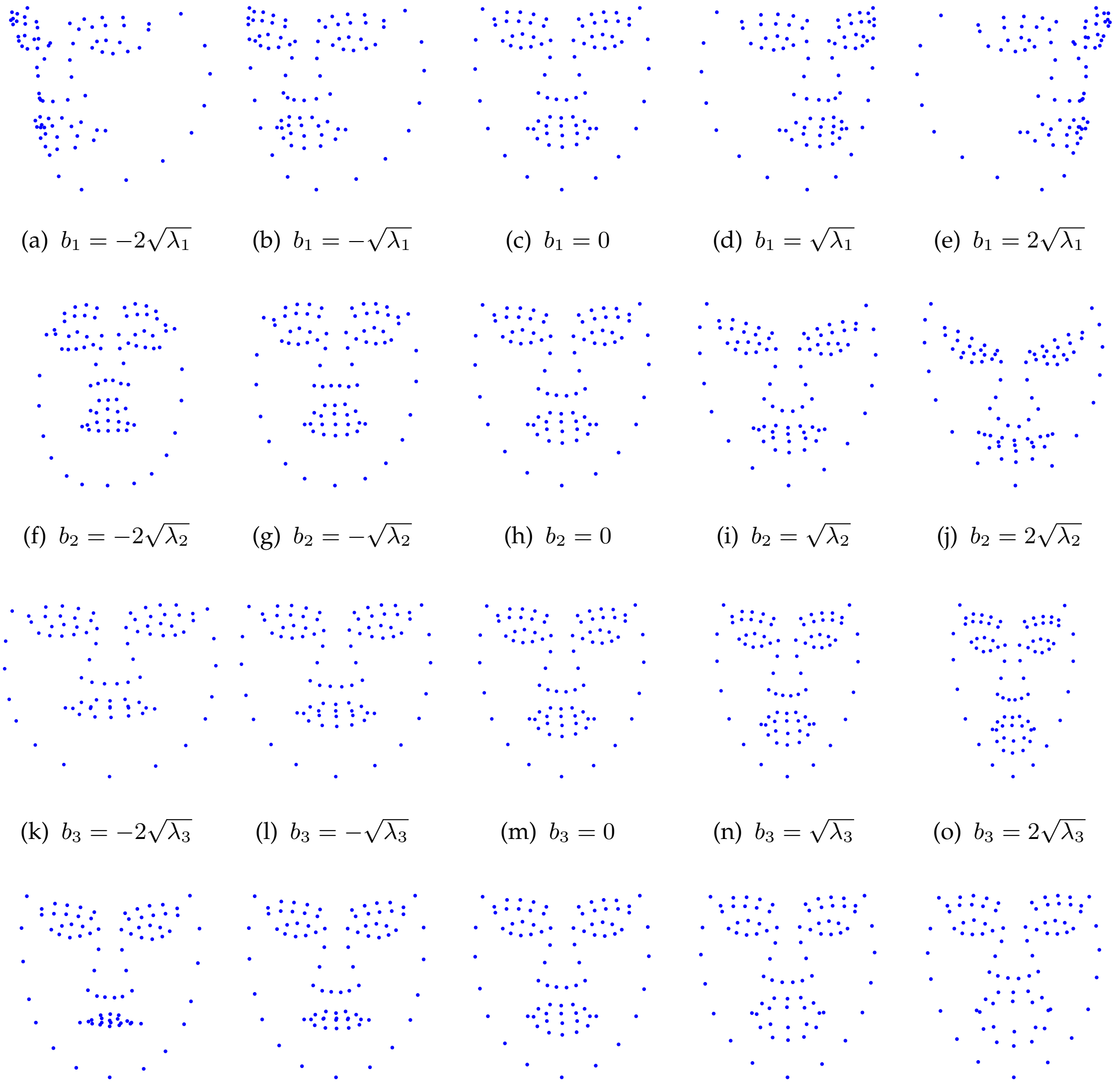
In Section 3.4, PDM result on the Vicon dataset is already shown in Figure 3.2. On our second dataset, the BU-3DFE face dataset, the PDM result is given in Figure 4.3. Because our original feature points only contain frontal view, zero-mean Gaussian pan and tilt are added as the rotation of the face and projected to the 2D camera measurements. The effects of the pose variations are seen in the first two rows of the plot. The behavior of the first row is somewhat similar to the experiment on the Vicon dataset. The tilt movement, which means to rotate in a vertical plane, is modeled with the second principal component. If a weight of $-2\sqrt{\lambda_2}$ of that shape basis is added, the new synthesized face model faces up. And with a positive weight imposed, the face model looks down. More notable changes in the width of the face is seen by applying the same amount of weight of the third eigenvector. This is because the Vicon dataset only consists of one subject, while in BU-3DFE, more subjects with a large variety of face shapes are available. As a result, more variation in this aspect is utilized to make a good model. In the final row, the mouth movement is also modeled as is seen in the previous experiment. Since the first two PCA component correspond to rigid movements, we take from the relational information from the third parameter to calculate the relationship between shapes using Equation (3.15).

A partial relational map constructed using PDM on the BU-3DFE dataset is illustrated in Figure 4.4. The color in the relational matrix plot is the same as in a "heat map", which reveals a higher value with warm colors and a lower value with cold colors. Thus, higher relation between frame 20 and frame 22, and lower relation between frame 6 and frame 20 are seen in the plot. From the profile view of these frames on the right side of the figure, we see that the shape of the cheek in frame 6 is obviously different than those in frame 20 and frame 22. Moreover, from the positions of the facial features we can also judge that the faces in the last two frames are the same one, while in frame 6, it is from another subject. Despite the fact that the first two faces both have a neutral expression, their shape difference is properly modeled in the relational matrix. From the result, we notice that although the BU-3DFE dataset covers a lot of subjects, its statistical information can still be effectively explained by PDM. This gives confidence of our 3D reconstruction algorithm based on PRPCA, which indeed yields a good performance on this dataset.

4.3.2 NRSFM Results with Different Numbers of Basis Shapes

The experiments conducted in this section intend to assess the performance of our NRSFM algorithms with different basis shape numbers. Since with the linear shape model, the choice of the shape number K is very sensitive in the previous work. Problems with insufficient shape number or overfitting may occur. Therefore, we test with a range from one deformation shape up to ten, in order to evaluate if the probabilistic approach really solves these problems.

The first Vicon dataset is a quite simple scenario. Since it is tracked using precise markers, we are able to evaluate on almost noise-free input data. The PDM analysis learned in Section 4.3.1 also suggest the number of the deformation bases be quite small. From the quantitative results in Figure 4.5, none of the evaluated algorithms exceed a reconstruction error of 3% except for the model with only one deformation shape basis, while the baseline PPCA performs slightly better. Unlike the conventional linear subspace model, which faces the overfitting problem with the basis shape number K growing up [THB08], here the probabilistic models learn the distribution parameters without fitting noise, which is seen from relatively constant error rates. Manifold PPCA has a small jump from seven shape bases to eight, which reaches about 3% error. But starting from $K = 9$, it stops growing up and the error rate re-stabilizes.



(a) $b_1 = -2\sqrt{\lambda_1}$ (b) $b_1 = -\sqrt{\lambda_1}$ (c) $b_1 = 0$ (d) $b_1 = \sqrt{\lambda_1}$ (e) $b_1 = 2\sqrt{\lambda_1}$

(f) $b_2 = -2\sqrt{\lambda_2}$ (g) $b_2 = -\sqrt{\lambda_2}$ (h) $b_2 = 0$ (i) $b_2 = \sqrt{\lambda_2}$ (j) $b_2 = 2\sqrt{\lambda_2}$

(k) $b_3 = -2\sqrt{\lambda_3}$ (l) $b_3 = -\sqrt{\lambda_3}$ (m) $b_3 = 0$ (n) $b_3 = \sqrt{\lambda_3}$ (o) $b_3 = 2\sqrt{\lambda_3}$

(p) $b_4 = -2\sqrt{\lambda_4}$ (q) $b_4 = -\sqrt{\lambda_4}$ (r) $b_4 = 0$ (s) $b_4 = \sqrt{\lambda_4}$ (t) $b_4 = 2\sqrt{\lambda_4}$

Figure 4.3: PDM result on the BU-3DFE dataset. The effects of the first four eigenvectors applied to the mean shape are illustrated in each row. From left to right, the weight increases from -2 to $+2$ times of the standard deviation with respect to the eigenvalue.

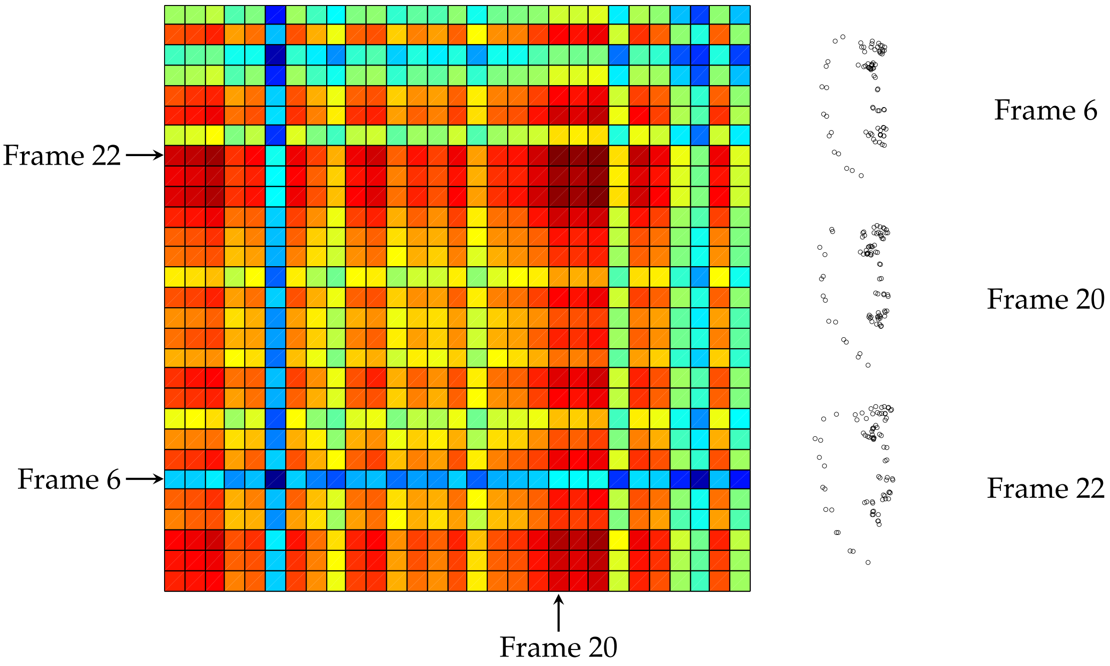


Figure 4.4: Plot of the partial relational matrix from the BU-3DFE dataset on the left. The heat color map reveals a higher relation between frame 20 and frame 22, and a lower relation between frame 6 and frame 20. These sample frames are plotted on the right respectively.

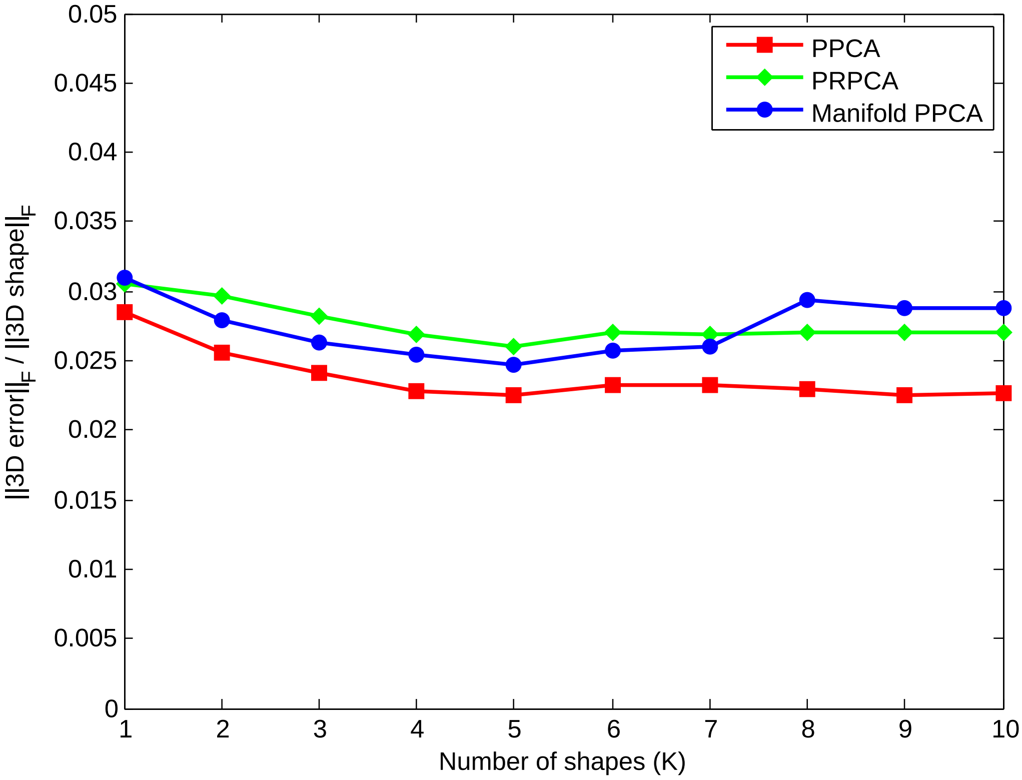


Figure 4.5: Reconstruction error as a function of the number of basis shapes on the Vicon dataset.

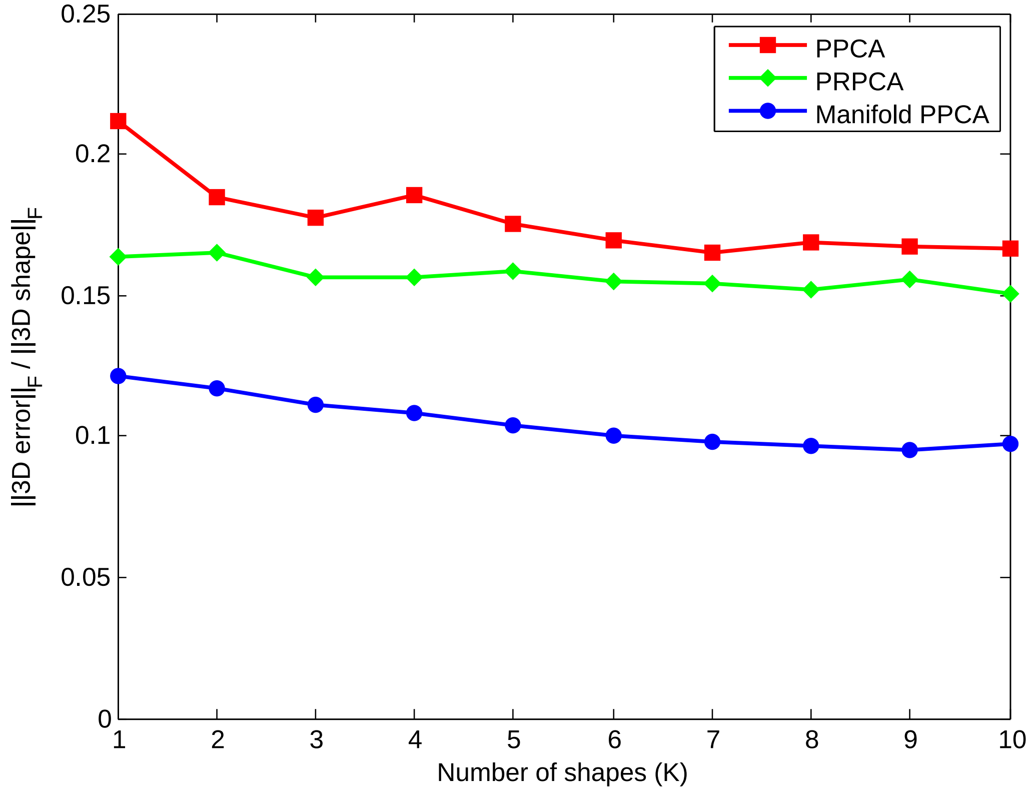


Figure 4.6: Reconstruction error as a function of the number of basis shapes on the BU-3DFE dataset.

The BU-3DFE dataset contains a variety of diverse facial expressions, which make it an ideal dataset for evaluating deformation shapes. Moreover, as many as 100 subjects are included in the dataset. So this is in a mostly different situation as with the Vicon data and the goal of this test is to learn how good a universal face model can be generated using the NRSFM algorithms. In Figure 4.6, our Manifold PPCA outperforms the baseline algorithm with a 6% to 8% absolute error gap in overall, regardless of the choice of K, which demonstrates a huge relative performance rise of 30% to 40% equivalently. It is also interesing to observe that with the help of the additional relational knowledge between frames and shapes, PRPCA is able to beat PPCA in this experiment with ca. 10% relative performance gain. Since the BU-3DFE dataset with a lot of subjects is more complex, all approaches need a larger K to model the shape parameters, but tuning this number does not cause much difference in the performance.

We also give the graphical reconstructions with five sample frames of all three algorithms for visualization. The knowledge from the results above revealing an insensitive impact of the choice of K, we select the median of five basis shapes. The first row shows the 2D tracks as inputs and the following rows give the plot of the baseline as well as our reconstructions in colored dots juxtaposed with the ground truth in black circles. Note that in order to emphasize the effect of the 3D depth recovery, these plots are shown from a different viewpoint, which is perpendicular to the original 2D inputs. On the Vicon dataset in Figure 4.7, all the three algorithms yield good structure and motion recovery and most of the feature points are almost perfectly positioned. Although a few misplaced features are visible, the reconstruction results are pretty satisfying.

In Figure 4.8, the qualitative results of the reconstructions is plotted. Unsurprisingly, the huge improvement of Manifold PPCA is also seen in the qualitative results. As we can

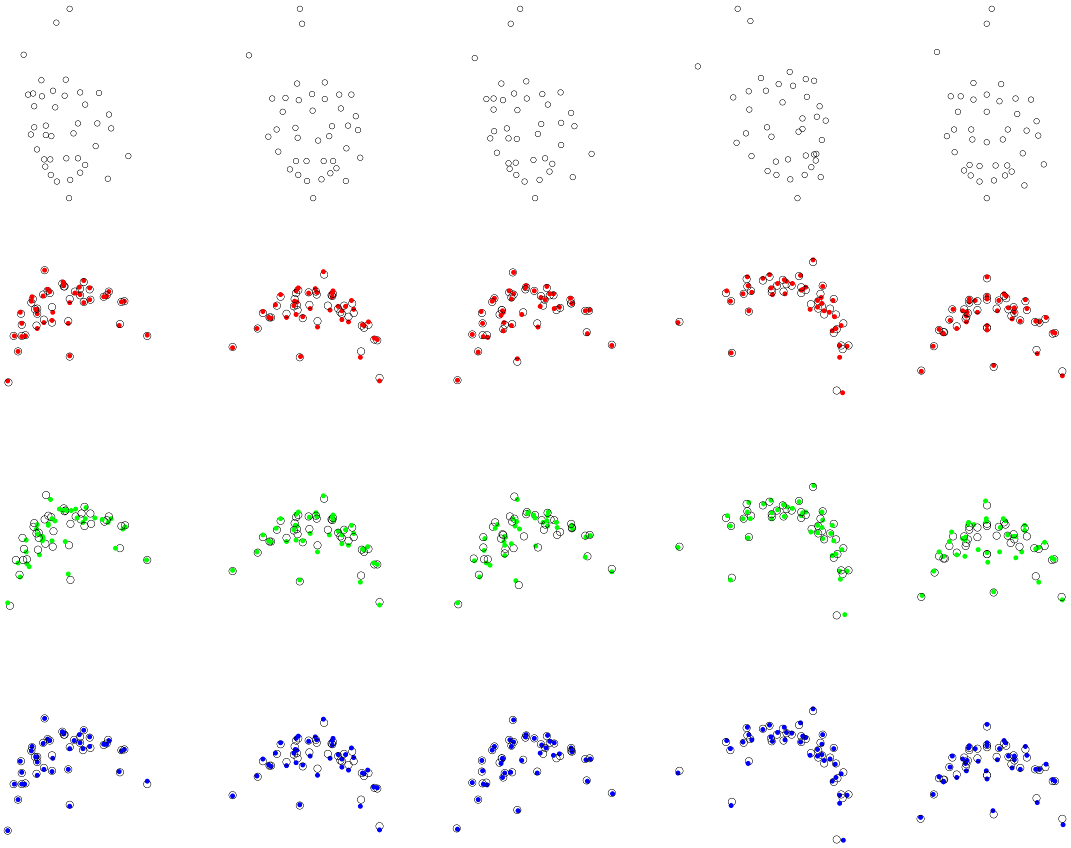


Figure 4.7: Vicon 2D tracks in the upper row, reconstruction results of PPCA, PRPCA and Manifold PPCA in the second, third and fourth row respectively. Images are captured at frame number 50, 100, 150, 200 and 250 respectively. Ground truth features are illustrated in black circles and reconstructions are colored dots.

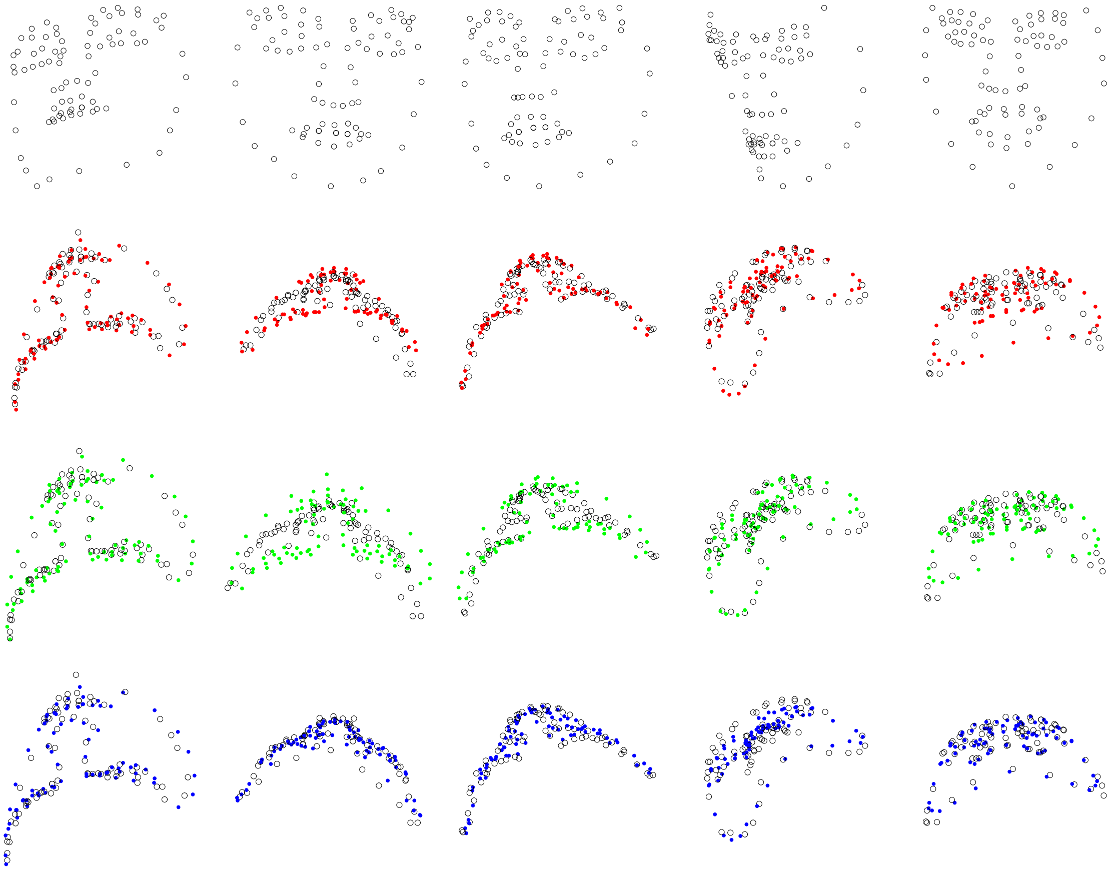


Figure 4.8: BU-3DFE 2D tracks in the upper row, reconstruction results of PPCA, PRPCA and Manifold PPCA in the second, third and fourth row respectively. Images are captured at frame number 50, 100, 150, 200 and 250 respectively. Ground truth features are illustrated in black circles and reconstructions are colored dots.

see in the first row, all of the frames contain different poses and facial expressions. In consequence, the recovered models in the following rows are hard to fit all shape instances for all of the three approaches. Once again, the PPCA and the PRPCA algorithms generate similar outcomes with hardly visible distinctions. For example, in frame 100, PPCA recovers the 3D motion better than PRPCA. On the other hand, PRPCA gets slightly better shape reconstructions in frame 200 and 250. However, our Manifold PPCA approach achieves clearly better results than the state-of-the-art. It is obvious that PPCA's inaccurate rotation approximation limits its result to getting better rotation estimate in frame 100. It also has difficulties to recover the contours of the faces correctly. Especially in frame 250, both PPCA and PRPCA lose the 3D depth information to some extent, whereas the Manifold PPCA method clearly performs better and recovers most of the key points correctly.

4.3.3 NRSFM Results with Noise

The real-world situation distinguishes itself from the image sequence of the experiments in that instead of tracking the feature points with markers, they are more likely outputs from a certain tracking algorithm. That means, the presence of noise must not be overlooked when evaluating the NRSFM algorithms. For this purpose, another experiment is conducted in order to investigate how the performance drops with additive noise. To

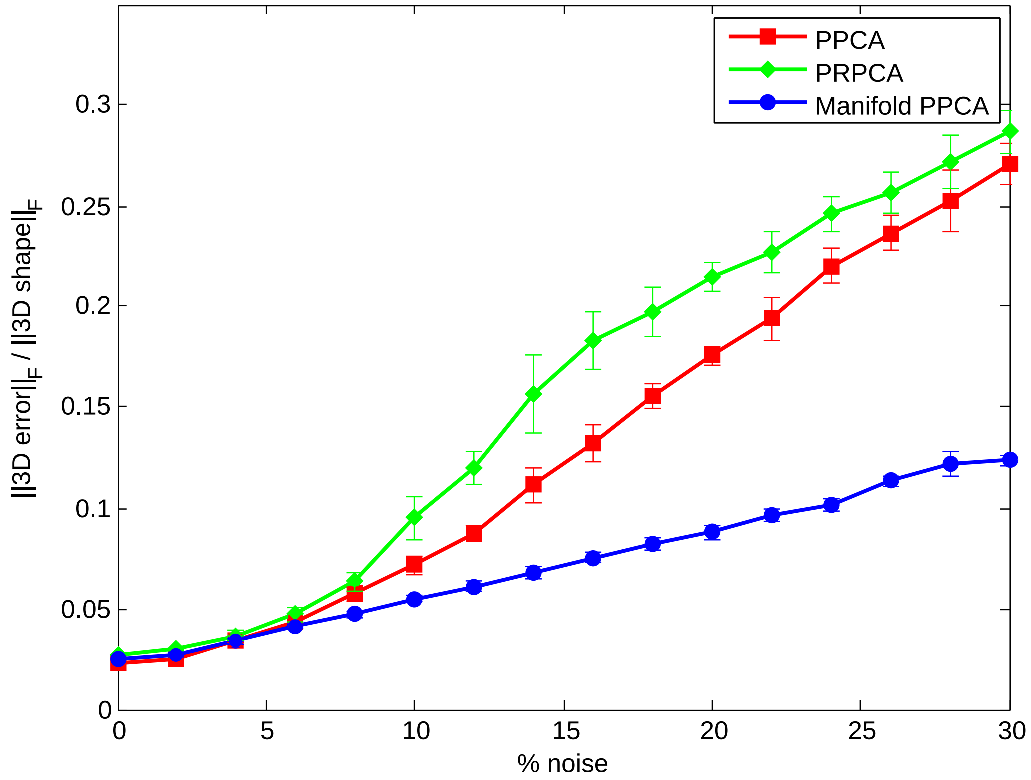


Figure 4.9: Reconstruction error with additive Gaussian noise up to 30% on the Vicon dataset. Results are averaged over 10 runs and error bars for standard deviations are also plotted.

eliminate the affect of random extremes at some noise level, each trial is run 10 times and the results are averaged. The noise levels span from 0% (no noise) to 30% with 2% step.

As can be observed from Figure 4.9, which plots the reconstruction errors on the Vicon dataset with additional noise, all algorithms continue their good outputs at the beginning in the noise-free experiment in Figure 4.5 and have practically the same error rate up to 6% noise. With more added noise, PPCA and PRPCA start to degenerate much more significantly than the Manifold PPCA approach. Starting from 20% noise level, the result gets 50% lower error rate than PPCA. That is most likely because in severe cases of noise, it is difficult for PPCA's rotation approximation in finding the updating direction and projecting back to the original manifold than with less noise. Unfortunately, PRPCA is still unable to outperform PPCA in this case, which is on average 2% to 5% inferior to the baseline algorithm. It also worth mentioning that Shaji and Chandran [SC08] also made evaluation on the Vicon dataset with additive noise. From their plot the performance degrades very quickly with noise level over 20%. However, our probabilistic approach on manifold does not suffer from this problem. Error bars are also plotted to demonstrate the standard deviations of each measurement. The longer error bars for PPCA and PRPCA show the instability of their performances with additive noise. For Manifold PPCA, however, the error bars are hardly visible for most of the measurements, which indicates its robustness against noise.

Since the feature points of the BU-3DFE dataset are manually labeled, noticeable noise can already be seen from the 2D input data. The multiple subjects may also be regarded as an obstruction or interference to the NRSFM model under the assumption of the linear shape basis. Hence by adding more Gaussian noise to the data in Figure 4.10, PPCA

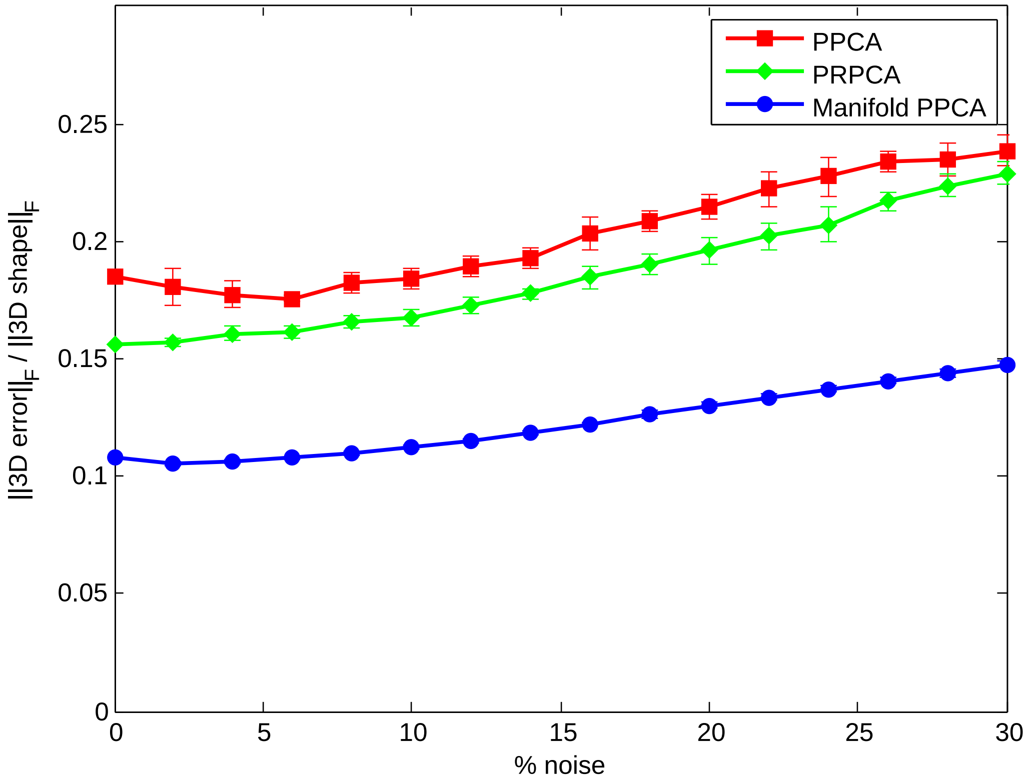


Figure 4.10: Reconstruction error with additive Gaussian noise up to 30% on the BU-3DFE dataset. Results are averaged over 10 runs and error bars for standard deviations are also plotted.

and PRPCA do not deteriorate as rapidly as on the "clean" Vicon dataset in Figure 4.9 with the increase of noise. Although the reconstructions degrade at a similar rate and the performance gap is kept throughout the experiment, Manifold PPCA is by far the best approach. Even with as much as 30% additive noise, it outperforms both PPCA and PRPCA on the noise-free data. These results reveal that to model more complicated shapes, an optimal rotation estimation using manifold optimization techniques is superior.

4.3.4 Subject Specific Analysis on the BU-3DFE Dataset

Given the improved performance on the BU-3DFE dataset over the state-of-the-art algorithm, it is obvious to conclude that using relational information and generalization of the Newton's method on manifold to solve the orthonormality constraints, both PRPCA and PPCA are capable of learning a better universal shape model on a dataset that contains more than one subject. Due to the contrary results compared to Figure 4.5, it is also interesting to see how our algorithms perform with a single subject. We make this experiment setup as close as the Vicon dataset so that five different subjects are selected. By applying zero-mean random pan and tilt as head pose, five new sub-datasets with a single subject of BU-3DFE are constructed. From the results in Table 4.1, Manifold PPCA still has the leading performance in almost each test except for subject 2 against PPCA, which has nearly the lowest reconstruction error for all algorithms. That proves again that PPCA's rotation approximation is only successful for uncomplicated motion and structure recovery. Beyond that, the results of Manifold PPCA are considerably more stable. From the fact that PRPCA is again unable to outperform the baseline algorithm, we infer that it is more appropriate for the scenario with multiple subjects.

	Subject 1	Subject 2	Subject 3	Subject 4	Subject 5	Mean
PPCA	6.28%	5.06%	10.38%	5.29%	8.97%	7.20%
PRPCA	8.94%	7.45%	11.70%	7.20%	9.43%	8.94%
Manifold PPCA	5.15%	5.35%	6.61%	4.53%	5.07%	5.34%

Table 4.1: Subject specific reconstruction results on the BU-3DFE dataset.

	Mixed average
PPCA	16.4%
PRPCA	14.4%
Manifold PPCA	8.0%

Table 4.2: Subject independent reconstruction results with five subjects on the BU-3DFE dataset.

To make further investigation on the basis of the subject specific experiment above, we make a new subset of the BU-3DFE dataset with exclusively the frames of the five subjects in Table 4.1. With the new image sequence, we intend to conduct a subject independent reconstruction test. As is seen in Table 4.2, our Manifold PPCA approach succeeds in building a generic model with the least reconstruction error increase of only ca. 2%–3% from the person specific model. In contrast, PPCA more than doubles the reconstruction error as well as PRPCA. As a conclusion, our proposed Manifold PPCA outperforms the state-of-the-art algorithm in nearly all situations, while PRPCA takes the advantage in modeling subject independent generic face model.

4.3.5 Convergence

In Section 2.3, we have learned that the generalized Newton's method employs the second order derivative, which has a theoretically better convergence property and is less possible to be stuck at a local minimum than the approximated Gauss-Newton step. In the following experiments, the 3D reconstruction error of each EM iteration is plotted in order to evaluate the convergence property of the PPCA and Manifold PPCA.

In the only case that PPCA performs slightly better than our Manifold PPCA, the convergence results are plotted in Figure 4.11. After the initial phase, both curves descend in a similar way and the Manifold PPCA reaches its minimum earlier at the 37th iteration.

On the BU-3DFE dataset without noise, the convergence results illustrated in Figure 4.12 are quite surprising. Starting from the same initialization, PPCA goes almost directly upwards. The more iterations are run, the higher 3D reconstruction error is reached. A possible cause of this phenomenon is that the rotation approximation employed by PPCA is problematic with the variation of multiple subjects in the BU-3DFE dataset. Note that the curves correspond to the reconstruction error measured by the 3D ground truth, where the ground truth is not available in the real NRSFM process. Hence with the negative log-likelihood objective function out of the 2D input data still decreasing during the EM iterations, it is not possible for PPCA to predict the failed recovery and stop the algorithm in time.

In the experiments with additional noise of 20%, the results for the Vicon dataset and the BU-3DFE dataset are shown in Figure 4.13 and Figure 4.14 respectively. In both plots, PPCA fails to make any improvements to the 3D reconstruction after the initialization. Employing the same probabilistic approach, however, Manifold PPCA continuously makes a better reconstruction by updating the rotation parameters on the manifold

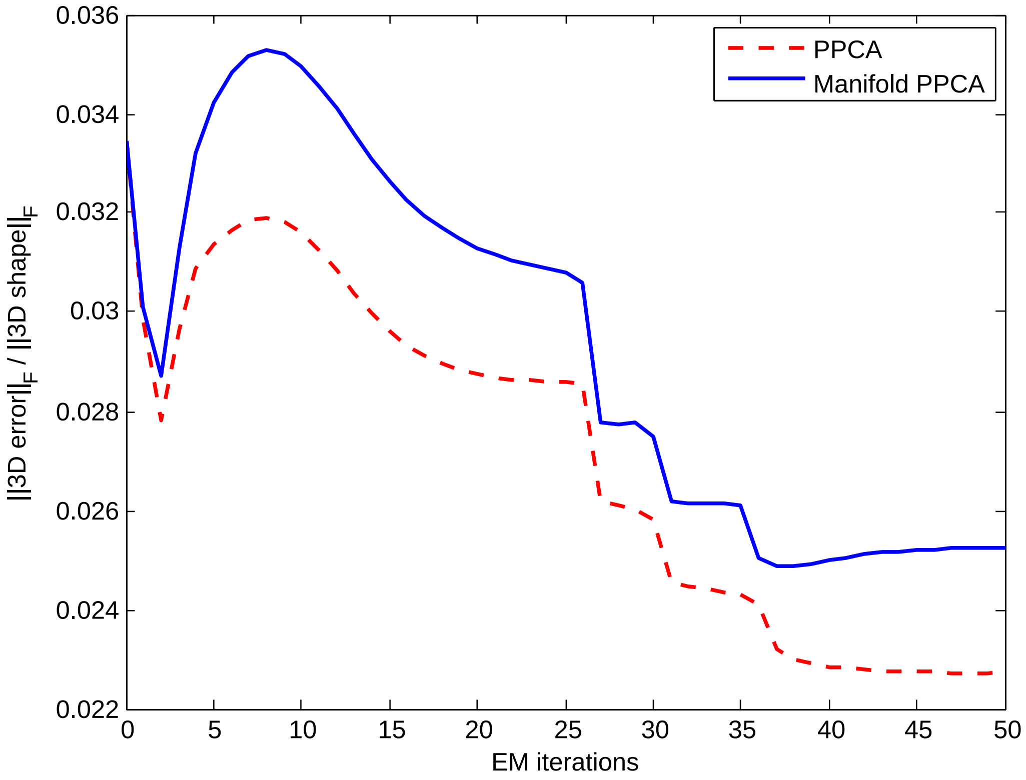


Figure 4.11: Convergence results without additive noise on the Vicon dataset.

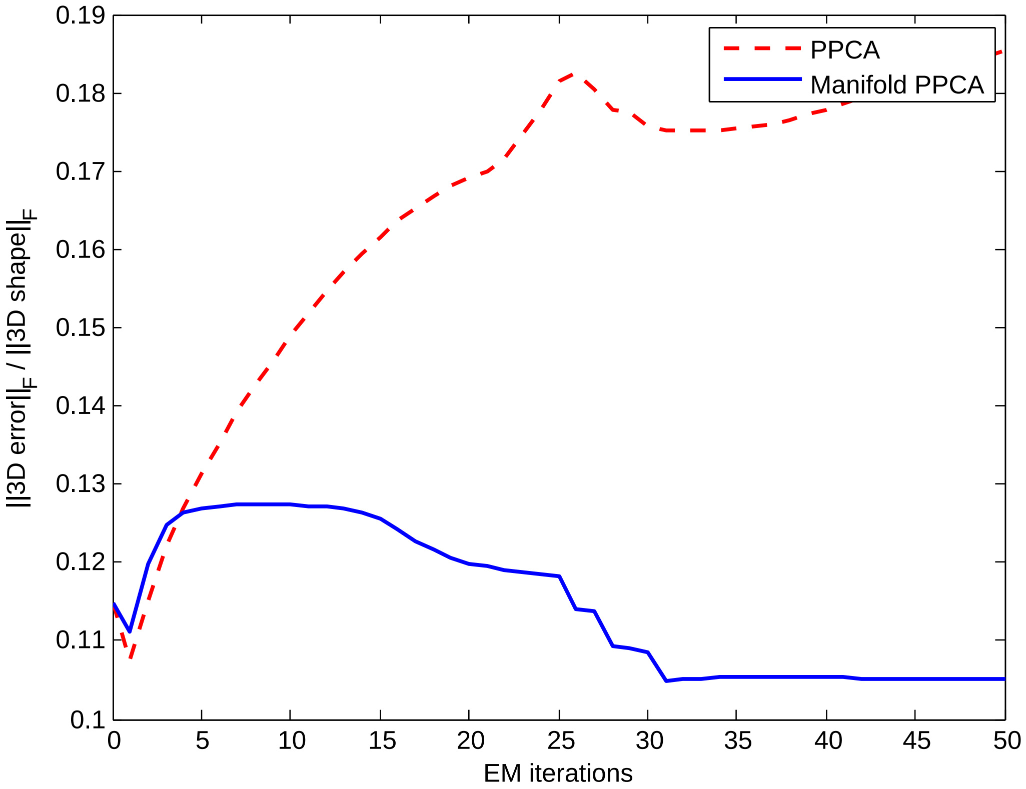


Figure 4.12: Convergence results without additive noise on the BU-3DFE dataset.

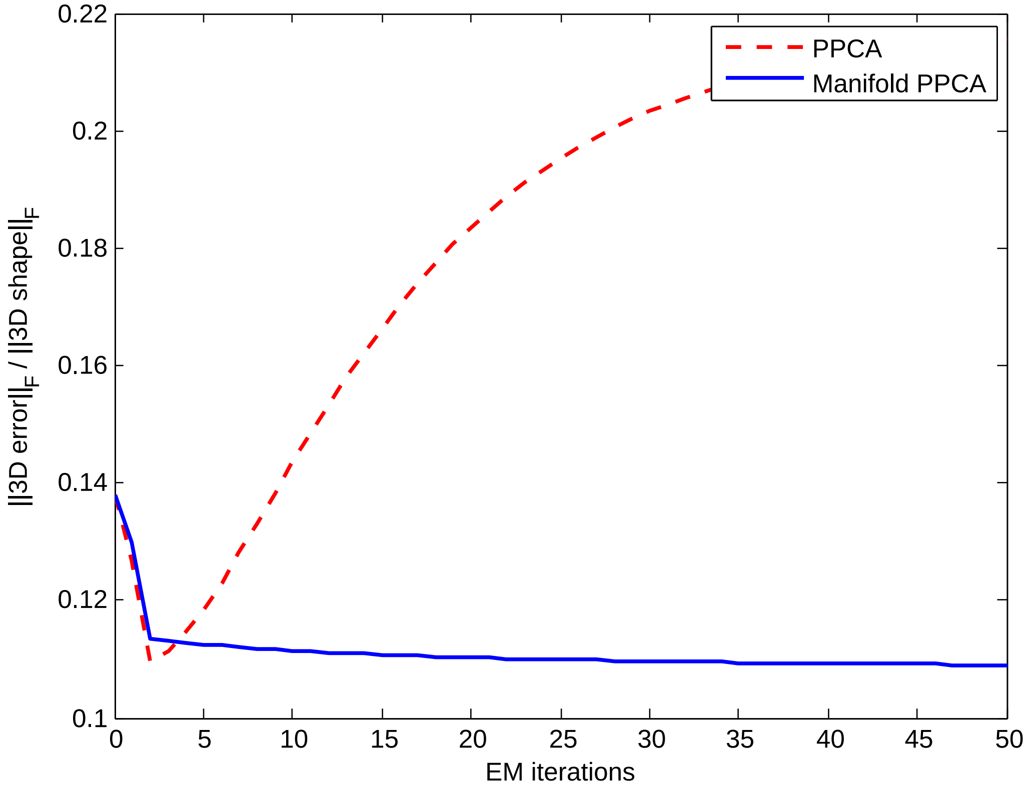


Figure 4.13: Convergence results with 20% additive noise on the Vicon dataset.

of the orthonormal group. The results clarify the fundamental problem why PPCA is increasingly affected by the noise added in Section 4.3.3.

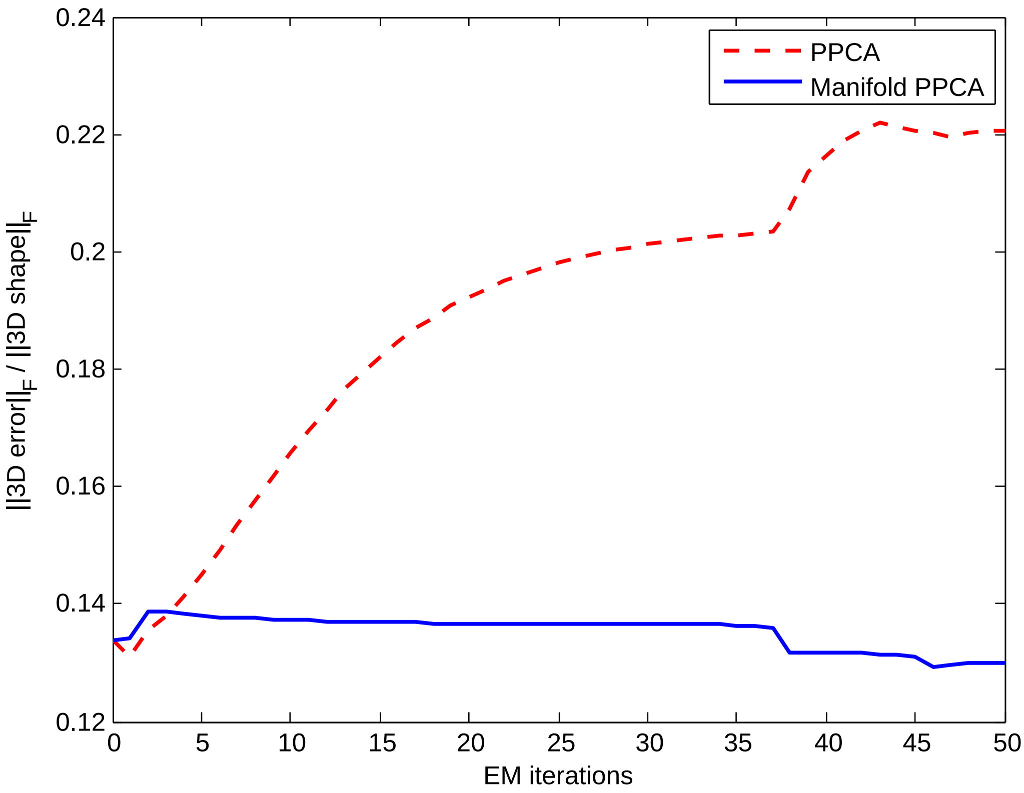


Figure 4.14: Convergence results with 20% additive noise on the BU-3DFE dataset.

5. Conclusion

Non-Rigid Structure from Motion—known as recovering three-dimensional object structure and camera motion from 2D monocular sequence of images—has become one of the most attractive and important tasks of 3D reconstruction because of its simplicity in comparison to other technical configurations. In this thesis, possible approaches of NRSFM have been intensively studied to improve the recovery performance of structures as well as rigid motions.

In order to solve this inherently underconstrained problem due to the additional degrees of freedom than in the rigid case, a low-rank subspace was employed. A probabilistic framework was built on the basis of PPCA due to its better performance than the closed-form solutions under noisy environment. With our PRPCA extension with relational shape information between frames, improved results over the baseline algorithm were obtained on the BU-3DFE dataset with multiple subjects, which met our expectation.

As a primary contribution of this work, we have presented a novel solution to unleash the orthonormality constraints of camera rotation matrix in the NRSFM problem by generalizing the Newton's method on the Riemannian manifold. Needless to conduct complex approximations, performing rotation update on the $SO(3)$ manifold implicitly ensures the validity of the constraints. In the experiments on the Vicon dataset without noise, we achieved comparable results with the state-of-the-art PPCA algorithm. With additional noise, this approach performed significantly better. Furthermore, on the BU-3DFE dataset it almost doubled the performance in all tests. As a conclusion, we have shown that the proposed approach is robust against noise, which indicates that it has more capability to deal with real-world data. Moreover, the superiority with multiple subjects also suggested the extreme importance of an optimal rotation estimation.

Based on the current system, several possibilities in future improvements are considered. We did find that although the performance was significantly improved on the BU-3DFE dataset, the linear subspace model somehow limited the modeling of multiple subjects. For example, articulated [PBS$^+$09] or nonlinear time series models [PRM01, WFH06] can provide better spatial and temporal representations. Another flaw of the current formulation is the orthographic or weak-perspective camera model, which can be replaced by the more realistic full-perspective camera model [LDBA06].

There is also some space for improvements in the core algorithms. The PRPCA-based approach failed to perform better PPCA in the single subject case. Alternative methods such as Variational PCA [Bis99] might offer better relational information. Additionally, besides manifold optimization of rotation matrices, we also plan to constrain the shape bases to orthonormal bases and consequently solve them on manifold.

As a future work, the current sparse face model can be densified with mapping or interpolation tools [GMDlTGZ10]. Then the appearance model [TH04] can be directly applied to the statistical estimation framework. Textured meshes with low-order deformation [SPIF07] is also applicable.

Bibliography

[ASK09] I. Akhter, Y. Sheikh, and S. Khan, "In defense of orthonormality constraints for nonrigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1534–1541.

[Avr03] M. Avriel, *Nonlinear Programming: Analysis and Methods*. Dover Publications, 2003.

[AWC$^+$07] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *AISTATS*, San Juan, Puerto Rico, 2007.

[Bar87] D. J. Bartholomew, *Latent Variable Models and Factor Analysis*. London: Charles Griffin & Co. Ltd., 1987.

[Bas94] A. T. Basilevsky, *Statistical Factor Analysis and Related Methods: Theory and Applications*. New York: Wiley, 1994.

[BB98] B. Bascle and A. Blake, "Separability of pose and expression in facial tracking and animation," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 323–328.

[BHB00] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2690–696, 2000.

[Bis99] C. M. Bishop, "Variational principal components," in *In Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99*, 1999, pp. 509–514.

[Bis07] C. M. Bishop, *Pattern Recognition and Machine Learning*, 1st ed. Springer, October 2007.

[BJ05] J. Barbič and D. L. James, "Real-time subspace integration for st. venant-kirchhoff deformable models," *ACM Trans. Graph.*, vol. 24, pp. 982–990, July 2005.

[Bra01] M. Brand, "Morphable 3D models from video," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 456–463, 2001.

[Bra05] M. Brand, "A direct method for 3D factorization of nonrigid motion observed in 2d," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 122–128, 2005.

[BV99] 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*, ser. SIGGRAPH '99. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[CET98] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Springer, 1998, pp. 484–498.

[CK98] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, pp. 159–179, 1998.

[CT01] T. F. Cootes and C. Taylor, "Statistical models of appearance for medical image analysis and computer vision," in *In Proc. SPIE Medical Imaging*, 2001, pp. 236–248.

[CTCG95] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models–their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, Jan. 1995.

[DHS01] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, November 2001.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[EAS99] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, pp. 303–353, April 1999.

[GCSR03] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. CRC Press, Jul. 2003.

[GH96] Z. Ghahramani and G. E. Hinton, "The em algorithm for mixtures of factor analyzers," Univ. of Toronto, Tech. Rep., 1996.

[GMDlTGZ10] J. Gonzalez-Mora, F. De la Torre, N. Guil, and E. L. Zapata, "Learning a generic 3D face model from 2D image databases using incremental structure-from-motion," *Image Vision Comput.*, vol. 28, pp. 1117–1129, July 2010.

[Gow75] J. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975, 10.1007/BF02291478.

[GR70] G. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, pp. 403–420, 1970, 10.1007/BF02163027.

[GT07] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning*. The MIT Press, 2007.

[GVL96] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[HK01] M. Han and T. Kanade, "Multiple motion scene reconstruction from uncalibrated views," *IEEE International Conference on Computer Vision*, vol. 1, pp. 163–170, 2001.

[Hor86] R. A. Horn, *Topics in matrix analysis*. New York, NY, USA: Cambridge University Press, 1986.

[Hot33] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psych.*, vol. 24, 1933.

[HZ04] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[Jol02] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. Springer, Oct. 2002.

[KsH05] T. Kim and K. sang Hong, "Estimating approximate average shape and motion of deforming objects with a monocular view," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, pp. 586–601, 2005.

[LDBA06] X. Lladó, A. Del Bue, and L. Agapito, "Euclidean reconstruction of deformable structure using a perspective camera with varying intrinsic parameters," in *Proceedings of the 18th International Conference on Pattern Recognition*, ser. ICPR '06, vol. 1. Washington, DC, USA: IEEE Computer Society, 2006, pp. 139–142.

[Lee03] J. M. Lee, *Introduction to Smooth Manifolds*. Springer-Verlag, 2003.

[LYZ09] W.-J. Li, D.-Y. Yeung, and Z. Zhang, "Probabilistic relational PCA," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, Eds., 2009, pp. 1123–1131.

[Man04] J. H. Manton, "On the various generalisations of optimisation algorithms to manifolds," *Sixteenth International Symposium on Mathematical Theory of Networks and Systems*, July 2004.

[Mat] MATLAB. [Online]. Available: http://www.mathworks.com/products/matlab/

[MK01] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 228–233, 2001.

[MKS99] Y. Ma, J. Košecká, and S. Sastry, "Optimization criteria and geometric algorithms for motion and structure estimation," *International Journal of Computer Vision*, vol. 44, pp. 219–249, 1999.

[MSZ94] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1994.

[OD07] T. Okatani and K. Deguchi, "On the wiberg algorithm for matrix factorization in the presence of missing components," *Int. J. Comput. Vision*, vol. 72, pp. 329–337, May 2007.

[Par72] F. I. Parke, "Computer generated animation of faces," in *Proceedings of the ACM annual conference - Volume 1*, ser. ACM '72. New York, NY, USA: ACM, 1972, pp. 451–457.

[PBS$^+$09] M. Paladini, A. D. Bue, M. Stošić, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 0, pp. 2898–2905, 2009.

[Pea01] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.

[PK93] C. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," Computer Science Department, Pittsburgh, PA, Tech. Rep. CMU-CS-93-219, December 1993.

[PRM01] V. Pavlović, J. M. Rehg, and J. Maccormick, "Learning switching linear models of human motion," in *Advances in Neural Information Processing Systems 13*, 2001, pp. 981–987.

[RB09] V. Rabaud and S. Belongie, "Linear embeddings in non-rigid structure from motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, 2009.

[Row98] S. Roweis, "EM algorithms for PCA and SPCA," in *in Advances in Neural Information Processing Systems*. MIT Press, 1998, pp. 626–632.

[SC08] A. Shaji and S. Chandran, "Riemannian manifold optimisation for non-rigid structure from motion," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–6.

[SK87] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Am. A*, vol. 4, no. 3, pp. 519–524, Mar 1987.

[SPIF07] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, "Surface deformation models for nonrigid 3D shape recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, pp. 1481–1487, August 2007.

[TB99] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal Of The Royal Statistical Society Series B*, vol. 61, no. 3, pp. 611–622, 1999.

[TH04] L. Torresani and A. Hertzmann, "Automatic non-rigid 3D modeling from video," in *In ECCV*, 2004, pp. 299–312.

[THBa] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure from motion MATLAB software. [Online]. Available: http://www.cs.dartmouth.edu/~lorenzo/projects/learning-nr-shape/em-sfm.zip

[THBb] L. Torresani, A. Hertzmann, and C. Bregler. Vicon face motion capture dataset. [Online]. Available: http://www.cs.dartmouth.edu/~lorenzo/Data/face.zip

[THB04] L. Torresani, A. Hertzmann, and C. Bregler, "Learning non-rigid 3D shape from 2D motion," in *Advances in Neural Information Processing Systems 16*, S. Thrun, L. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004.

[THB08] 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*, vol. 30, pp. 878–892, 2008.

[TJK10] J. Taylor, A. Jepson, and K. Kutulakos, "Non-rigid structure from locally-rigid motion," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2761–2768.

[TK92] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137–154, 1992, 10.1007/BF00129684.

[TK08] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. Academic Press, 2008.

[TP91] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, pp. 71–86, January 1991.

[TR05] P. Tresadern and I. Reid, "Articulated structure from motion by factorization," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, ser. CVPR '05, vol. 2. Washington, DC, USA: IEEE Computer Society, 2005, pp. 1110–1115.

[Tri96] B. Triggs, "Factorization methods for projective structure and motion," in *CVPR*, 1996, pp. 845–851.

[TYAB01] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 493–500, 2001.

[Ull83] S. Ullman, "Maximizing rigidity: The incremental recovery of 3-D structure structure from rigid and nonrigid motion," *Perception*, vol. 13, pp. 255–274, 1983.

[WFH06] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Advances in Neural Information Processing Systems 18*. MIT Press, 2006, pp. 1441–1448.

[Woo50] M. A. Woodbury, "Inverting modified matrices," Statistical Research Group, Memo, Princeton, N. J., Tech. Rep. 42, 1950.

[XCK06] J. Xiao, J. Chai, and T. Kanade, "A closed-form solution to non-rigid shape and motion recovery," *International Journal of Computer Vision*, vol. 67, pp. 233–246, April 2006.

[XK04] J. Xiao and T. Kanade, "Non-rigid shape and motion recovery: Degenerate deformations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2004, pp. 668–675.

[YJS06] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, December 2006.

[YKA02] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, Jan. 2002.

[YP05] J. Yan and M. Pollefeys, "A factorization-based approach to articulated motion recovery," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 815–821, 2005.

[YWS$^+$06] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3D facial expression database for facial behavior research," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition*, ser. FGR '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 211–216.

[ZCPR03] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, “Face recognition: A literature survey,” *ACM Comput. Surv.*, vol. 35, pp. 399–458, December 2003.