Ph.D. DISSERTATION

Probabilistic 3D Human Pose Recovery and Its Application to Action Recognition

확률적인 3차원 자세 복원과 행동인식

BY

JUNGCHAN CHO

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Probabilistic 3D Human Pose Recovery and Its Application to Action Recognition

확률적인 3차원 자세 복원과 행동인식

BY

JUNGCHAN CHO

FEBRUARY 2016

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Abstract

These days, computer vision technology becomes popular and plays an important role in intelligent systems, such as augment reality, video and image analysis, and to name a few. Although cost effective depth cameras, like a Microsoft Kinect, have recently developed, most computer vision algorithms assume that observations are obtained from RGB cameras, which make 2D observations. If, somehow, we can estimate 3D information from 2D observations, it might give better solutions for many computer vision problems.

In this dissertation, we focus on estimating 3D information from 2D observations, which is well known as non-rigid structure from motion (NRSfM). More formally, NRSfM finds the three dimensional structure of an object by analyzing image streams with the assumption that an object lies in a low-dimensional space. However, a human body for long periods of time can have complex shape variations and it makes a challenging problem for NRSfM due to its increased degree of freedom. In order to handle complex shape variations, we propose a Procrustean normal distribution mixture model (PNDMM) by extending a recently proposed Procrustean normal distribution (PND), which captures the distribution of non-rigid variations of an object by excluding the effects of rigid motion. Unlike existing methods which use a single model to solve an NRSfM problem, the proposed PNDMM decomposes complex shape variations into a collection of simpler ones, thereby model learning can be more tractable and accurate. We perform experiments showing that the proposed method outperforms existing methods on highly complex and long human motion sequences.

In addition, we extend the PNDMM to a single view 3D human pose estimation problem. While recovering a 3D structure of a human body from an image is important, it is a highly ambiguous problem due to the deformation of an articulated human body. Moreover, before estimating a 3D human pose from a 2D human pose, it is important to obtain an accurate 2D human pose. In order to address inaccuracy of 2D pose estimation on a single image and 3D human pose ambiguities, we estimate multiple 2D and 3D human pose candidates and select the best one which can be explained by a 2D human pose detector and a 3D shape model. We also introduce a model transformation which is incorporated into the 3D shape prior model, such that the proposed method can be applied to a novel test image. Experimental results show that the proposed method can provide good 3D reconstruction results when tested on a novel test image, despite inaccuracies of 2D part detections and 3D shape ambiguities.

Finally, we handle an action recognition problem from a video clip. Current studies show that high-level features obtained from estimated 2D human poses enable action recognition performance beyond current state-of-the-art methods using low- and mid-level features based on appearance and motion, despite inaccuracy of human pose estimation. Based on these findings, we propose an action recognition method using estimated 3D human pose information since the proposed PNDMM is able to reconstruct 3D shapes from 2D shapes. Experimental results show that 3D pose based descriptors are better than 2D pose based descriptors for action recognition, regardless of classification methods. Considering the fact that we use simple 3D pose descriptors based on a 3D shape model which is learned from 2D shapes, results reported in this dissertation are promising and obtaining accurate 3D information from 2D observations is still an important research issue for reliable computer vision systems.

Keywords: 3D Shape Recovery, Non-Rigid Structure from Motion, 3D Human Pose Estimation, Action Recognition

Student Number: 2010-20902

Contents

1 Introduction 1

1.1 Motivation 1

1.2 Research Issues 4

1.3 Organization of the Dissertation 6

2 Preliminary 9

2.1 Generalized Procrustes Analysis (GPA) 11

2.2 EM-GPA Algorithm 12

2.2.1 Objective function 12

2.2.2 E-step 15

2.2.3 M-step 16

2.3 Implementation Considerations for EM-GPA 18

2.3.1 Preprocessing stage 18

2.3.2 Small update rate for the covariance matrix 20

2.4 Experiments 21

2.4.1 Shape alignment with the missing information 23

2.4.2 3D shape modeling 24

2.4.3 2D+3D active appearance models 28

2.5 Chapter Summary and Discussion 32

3 Procrustean Normal Distribution Mixture Model 33

3.1 Non-Rigid Structure from Motion 35

3.2 Procrustean Normal Distribution (PND) 38

3.3 PND Mixture Model 41

3.4 Learning a PNDMM 43

3.4.1 E-step 44

3.4.2 M-step 46

3.5 Learning an *Adaptive* PNDMM 48

3.6 Experiments 50

3.6.1 Experimental setup 50

3.6.2 CMU Mocap database 53

3.6.3 UMPM dataset 69

3.6.4 Simple and short motions 74

3.6.5 Real sequence - qualitative representation 77

3.7 Chapter Summary 78

4 Recovering a 3D Human Pose from a Novel Image 83

4.1 Single View 3D Human Pose Estimation 85

4.2 Candidate Generation 87

4.2.1 Initial pose generation 87

4.2.2 Part recombination 88

4.3 3D Shape Prior Model 89

4.3.1 Procrustean mixture model learning 89

4.3.2 Procrustean mixture model fitting 91

4.4 Model Transformation 92

4.4.1 Model normalization 92

4.4.2 Model adaptation 95

4.5 Result Selection 96

4.6 Experiments 98

4.6.1 Implementation details 98

4.6.2 Evaluation of the joint 2D and 3D pose estimation 99

4.6.3 Evaluation of the 2D pose estimation 104

4.6.4 Evaluation of the 3D pose estimation 106

4.7 Chapter Summary 108

5 Application to Action Recognition 109

5.1 Appearance and Motion Based Descriptors 112

5.2 2D Pose Based Descriptors 113

5.3 Bag-of-Features with a Multiple Kernel Method 114

5.4 Classification - Kernel Group Sparse Representation 115

5.4.1 Group sparse representation for classification 116

5.4.2 Kernel group sparse (KGS) representation for classification . . . 118

5.5 Experiment on sub-JHMDB Dataset 120

5.5.1 Experimental setup 120

5.5.2 3D pose based descriptor 122

5.5.3 Experimental results 123

5.6 Chapter Summary 129

6 Conclusion and Future Work 131

Appendices 135

A Proof of Propositions in Chapter 2 137

A.1 Proof of Proposition 1 137

A.2 Proof of Proposition 3 138

A.3 Proof of Proposition 4 139

B Calculation of $p(\mathbf{X}_i|\mathbf{D}_i, \mathbf{\Phi}_i)$ in Chapter 3 141

B.1 Without the Dirac-delta term 141

B.2 With the Dirac-delta term 142

C Procrustean Mixture Model Learning and Fitting in Chapter 4 145

C.1 Procrustean Mixture Model Learning 145

C.2 Procrustean Mixture Model Fitting 147

List of Figures

1.1 Analysis of video data. 2

2.1 An example of an incorrectly reconstructed mean shape. 19

2.2 The value of the objective function of the EM-GPA. 21

2.3 An example of landmark points in a face. 24

2.4 Two examples of alignment of the EM-GPA. 25

2.5 The estimated mean shape and the ground truth shape obtained from the FRGC database. 26

2.6 The canonical correlation obtained from the FRGC database. 26

2.7 Examples of image fitting by the 2D AAM, 2D+3D AAM/real_shape, and 2D+3D AAM/virtual_shape. 31

3.1 A graphical illustration of the PNDMM. 34

3.2 A graphical representation of the PNDMM. 41

3.3 An example of the value of objective function of the PNDMM. 49

3.4 A comparison of initialization methods: continuous and interleaved. . . 56

3.5 The effect of an increasing camera rotation on the reconstruction stability. 57

3.6 An example of a temporal smoothing (CMU86_04). 62

3.7 Reconstruction results from the CMU Mocap dataset experiment. 63

3.8 An example of human motion clustering using an *adaptive* PNDMM (CMU86_05). 65

3.9 Reconstruction errors at three different noise levels. 68

3.10 Reconstruction errors at five different missing data ratios. 70

3.11 Reconstruction results from the UMPM dataset experiment. 72

3.12 Successful reconstruction results from the Penn Action dataset. 79

3.13 Successful reconstruction results from the Penn Action dataset (continued). 80

3.14 Successful reconstruction results from the Penn Action dataset (continued). 81

3.15 Failed reconstruction results from the Penn Action dataset. 82

4.1 An overview of single view 3D hum pose estimation using the PNDMM. 84

4.2 An example of a tree structure and the corresponding part different matrix **J**. 94

4.3 Examples of 3D pose estimation results from the Human Eva dataset. . 103

4.4 Examples of reconstruction results from the LSP dataset. 105

4.5 Normalized reconstruction error. 107

5.1 Overview. 111

5.2 An example of a score function for a binary classification. 120

5.3 Examples of the sub-JHMDB dataset. 121

5.4 A graphical illustration of a 3D human pose based descriptor. 123

5.5 Confusion matrix obtained from support vector machine. 127

5.6 Confusion matrix obtained from kernel group sparse representation. . . 128

C.1 A graphical representation of reconstruction process. 148

List of Tables

2.1 AAM fitting performance 30

2.2 The rate of successful convergence for various poses 30

3.1 Reconstruction errors of NRSfM methods on the CMU Mocap database. 59

3.2 Reconstruction errors of NRSfM methods on the UMPM dataset. 73

3.3 Reconstruction errors of NRSfM methods on a complex sequence synthesized using the UMPM dataset. 74

3.4 Reconstruction errors of NRSfM methods on simple and short sequences 76

4.1 Reconstruction errors (mm) according to multiple candidates (training set: CMU Mocap, test set: HumanEva). 101

4.2 Reconstruction errors (mm) on the HumanEva dataset (training set: CMU Mocap) 102

4.3 Reconstruction errors (mm) on the HumanEva dataset (training set: HumanEva). 102

4.4 2D Part Detection Performance on LSP Dataset 104

5.1 Evaluation of the proposed method on the sub-JHMDB dataset with estimated 2D and 3D human poses (w/o scale normalization) 125

5.2 Evaluation of the proposed method on the sub-JHMDB dataset with estimated 2D and 3D human poses (w/ scale normalization) 126

5.3 Evaluation of the proposed method on the sub-JHMDB dataset with ground truth 2D human poses and estimated 3D human poses 129

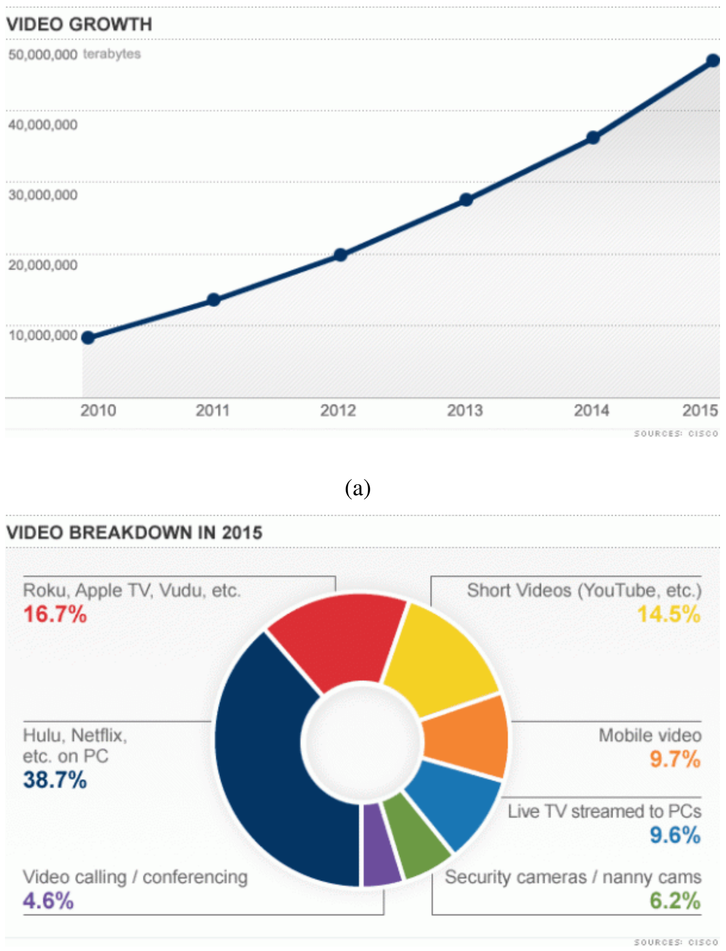
Chapter 1

Introduction

1.1 Motivation

The advent of digital cameras and smartphones, and the growth of social networking technologies, such as YouTube, Flickr, and Facebook, have accelerated making and sharing of a lot of images and videos. According to the research by Cisco Systems, Inc., the growth of videos in internet is expected as shown in Figure 1.1(a). The volume of videos would be 50,000,000 terabytes in the year 2015 and video data can be categorized as shown in Figure 1.1(b). Since the volume of video data is already over the limitation we can control, technologies for efficiently analyzing contents of data are required in these days.

When we analyze video data, understanding human actions is one of the important things due to its various and useful applications, such as intelligence surveillance, human-computer interaction, web-video search and retrieval, and to name a few. It is easy for a person to recognize actions performed by other persons, while current intelligent systems still struggle with robustly recognizing human actions in uncontrolled environments. In order to solve action recognition problems, most of the studies are



(a)

(b)

Figure 1.1: Analysis of video data. (a) Video growth in internet, (b) Categorization of video data. (Source: Cisco Systems, Inc., http://www.reelseo.com/rise-online-video-break-internet/)

based on general visual cues like texture, edge, and color in an image.

However, such methods have a limitation due to lack of high-level information inherited from weak visual cues. Since human motion can be interpreted as variations of a human pose in a physical view, moving information of joints of an articulated human pose becomes a high-level descriptor on human motion [1]:

- A pose is more invariant than appearance-based cues in the same action. For instance, appearance based features can differ between two persons wearing different clothes in color and texture, while poses are similar when they are performing the same action.
- The pose itself is able to simplify the learning process for action recognition since a pose is a kind of high-level descriptors on human motion.

Despite a lot of advantages in using human poses, posed-based action recognition has not received attention over past few decades due to the difficulty of human pose estimation on an image. However, current great progress in human pose estimation makes it possible to robustly estimate human poses in images. Yao *et al.*[2] showed that human pose estimation from multiple camera views are accurate enough for reliable action recognition. Although there is still a limitation for pose estimation in a monocular video, several studies point to that utilizing poses is important for better understanding human motion and improving the accuracy of action recognition systems [3, 4, 5]. Recently, Jhuang *et al.*[4] have shown that high-level features obtained from 2D human poses estimated by [6], as well as ground truth poses, enable action recognition performance beyond current state-of-the-art methods using low- and mid-level features based on appearance and motion.

Based on current findings about human analysis, we can easily think about that usage of 3D human poses might be better solutions than 2D human poses in action recognition systems. That is, the real world consists of 3D objects and a 2D human pose is the perspective of the 3D one. For that reason, cost effective depth sensors, *e.g.*, Microsoft Kinect, have been recently developed and several researchers have utilized them for 3D human pose estimation and action recognition. However, a depth camera is an additional sensor and depth based methods are not general technology which can be applied to unrestricted environments, such as RGB based systems. To better understand human motion in 2D images and videos, we need to obtain 3D information from 2D images, which remains as an unsolved problem.

1.2 Research Issues

In this dissertation, we will focus on estimating 3D information from 2D images and show an application on usage of a 3D human pose for video based human action recognition. Our research issues are can be summarized as follows.

- **Non-Rigid Structure from Motion:** Recovering a 3D shape and motion from a set of 2D observations is a central problem in computer vision, which can be applied to a number of interesting applications, such as scene understanding, motion capture and animation, medical imaging, and augmented reality, to name a few. Structure from motion (SfM) is a popular method to estimate the 3D shape and motion of a rigid object, which has been well studied. However, non-rigid structure from motion (NRSfM) has been remained as a challenging problem due to its increased degree of freedom. There have been significant efforts to solve NRSfM by introducing additional constraints and many have focused on restricting the degree of deformation with low-rank assumptions [7, 8, 9, 10]. While the assumption used in many methods helps to handle simple shape variations of a non-rigid object, *e.g.*, walking and drinking, it is rather restrictive to handle real world shape variations of complex non-rigid deformations.

- **Single View 3D Human Pose Estimation:** Estimating a 3D human pose from a single image has received a significant attention in computer vision due to its wide range of potentially useful applications, such as human-computer interaction, intelligent surveillance, and scene understanding, to name a few. In general, 3D human pose estimation based on 2D body part locations is done by first detecting body parts from the image and then recovering a 3D pose using a 3D shape model of a human body. However, currently available 2D part detectors cannot accurately localize key joints in all cases. In addition, recovering a 3D shape from its projection in a 2D image is inherently an ill-posed problem because different 3D shapes may generate similar 2D projections [11, 12]. While there has been many efforts to estimate a 3D human pose from 2D part locations with the prior information about a 3D human body, developing a sound mathematical model is still an open issue.

- **Action Recognition Using Virtual 3D Pose Based Descriptors:** Since Laptev [13] has introduced space-time interest points by extending the Harris detector, interest point based local descriptors have been successfully extended from images to videos, which have achieved state-of-the-art results for action recognition when combined with a bag-of-features representation. However, weak visual cue based approaches have a limitation for complex scenes, so 2D human pose based action recognition has been recently revisited. Moreover, Jhuang *et al.*[4] have shown that usage of 2D human poses can improve action recognition in complex scene when whole body is visible.

1.3 Organization of the Dissertation

Chapter 2 describes the study of simple shapes, such as a face, before studying complex shapes, such as a human body. According to Kendall's definition [14], the shape of an object is the geometrical information that remains after the effects of the Euclidean similarity transformations (rigid transforms) are filtered out. In many cases, this information can be found by aligning a set of shapes to a common reference using generalized Procrustes analysis (GPA) [15, 16]. However, a set of shapes has to be aligned with some missing information in many recent applications [17, 8, 9], which induces several problems. In particular, since 2D shapes can be consider 3D shapes with missing depth information, if we can apply GPA to the case with hidden (missing) variables, it will be very useful as shown in Chapter 2.

In addition, GPA with missing depth information can be considered to be equivalent to recovering 3D shapes and motion from a set of 2D shapes, *i.e.*, non-rigid structure from motion (NRSfM). However, a scale constraint in the GPA makes a nonlinear manifold, which leads difficulty on the NRSfM problem. To make GPA more tractable for NRSfM, Lee *et al.*[18] have proposed a new probability distribution, called the Procrustean normal distribution (PND), which captures the distribution of non-rigid variations of an object by excluding the effects of rigid motion.

In Chapter 3, we focus on reconstructing the 3D shape of a non-rigid object under complex shape variations by extending a PND [18] to a mixture of PNDs. We call the proposed method Procrustean normal distribution mixture model (PNDMM). Unlike existing methods which use a single model to solve an NRSfM problem, PNDMM decomposes complex shape variations into a collection of simpler shape variations, thereby model learning can be more tractable and accurate.

Chapter 4 extends the PNDMM to a single view 3D human pose estimation problem. Since it is a highly ambiguous problem caused by large degree of freedom of an articulated human body and self-occlusion in an image plane, introducing additional knowledge is required to restrict the size of the solution space. In order to solve the ill-posed problem, we learn a PNDMM proposed in Chapter 3 and adaptively fit it to a new 2D observation. We have also introduced model transformation which is incorporated into the 3D shape prior model, such that the proposed method can be applied to a novel test image.

Finally, Chapter 5 show the possibility of action recognition using estimated 3D human pose information. To generate 3D human pose based descriptors, we utilize the single view 3D human pose estimation method proposed in Chapter 4 and show that 3D pose based descriptors are better than 2D pose based descriptors for action recognition, regardless of classification methods. Considering the fact that we use simple 3D pose based descriptors based on a 3D shape model learned from 2D shapes, results in this dissertation are promising and obtaining accurate 3D information from 2D observations is a very important research issue for reliable computer vision systems.

Chapter 2

Preliminary

In computer vision, the study of shape using a set of landmark points is an important issue which appears in many application, such as image registration [20, 21, 22], gait recognition [23, 24], shape modeling [25, 26], motion analysis [27, 28, 23, 24], and stereo reconstruction [29]. According to Kendall's definition [14], the shape of an object is the geometrical information that remains after the effects of the Euclidean similarity transformations (rigid transforms) are filtered out. In many cases, this information can be found by aligning a set of shapes to a common reference using generalized Procrustes analysis (GPA) [15, 16]. GPA performs Euclidean similarity transforms on a set of shapes to minimize the sum of squared distances between all shapes and a reference shape.

In this chapter, we extend GPA to the case with hidden (missing) variables by using the expectation-maximization (EM) algorithm, which will be called EM-GPA hereafter. In expectation-step (E-step) of EM-GPA, the missing information is modeled as a Gaussian distribution, and in maximization step (M-step) of EM-GPA, the maximum like-

This chapter is based on the paper appeared in Computer Vision and Image Understanding: 'EM-GPA: Generalized Procrustes Analysis with Hidden Variables for 3D Shape Modeling [19]'.

lihood (ML) solution of the parameters is obtained by using the distribution of hidden variables computed from E-step. During M-step, some constraints that reflect the characteristics of GPA are enforced to resolve the ambiguity. These constraints align shapes with respect to (w.r.t.) the Euclidean measure, which makes the parameters related to the rigid transforms less affected by the parameters related to deformation. EM-GPA is not limited to the case of missing depth information, but it can be easily extended to more general cases.

In Section 2.4.2 and Section 2.4.3, we show that EM-GPA can find scales, rotations, the mean and covariance matrix of 3D shapes only with observed 2D facial shapes. The mean and covariance matrix of 3D shapes obtained by EM-GPA can be used to build a 3D shape model instead of using those trained by a real 3D landmark data, which usually requires extra efforts to produce.

Relation with other chapters Considering that this chapter estimates 3D information of shapes from a set of 2D shapes, the goal of EM-GPA is conceptually similar to recovering 3D shapes and motion from a set of 2D shapes (non-rigid structure from motion, NRSfM), and many studies have solved this by using factorization methods [30, 31, 7, 9]. We conjecture that it is better to put constraints on the rigid transforms so that they are not affected by the characteristics of the deformation space. Based on this conjecture, Lee *et al.*[18] recently proposed a new distribution representing non-rigid shape variations using the GPA concept, which shows state-of-the-art on NRSfM. In Chapter 3, we extend a PND into a PND mixture model and show that the PND mixture model can handle a complex and long shape variations, which results in improving the 3D reconstruction performances.

2.1 Generalized Procrustes Analysis (GPA)

GPA is one of the most popular algorithms to align shapes to a common reference. Given a set of shapes $\mathbf{X}_i \in \mathbb{R}^{m \times n_p}, i = 1, \ldots, n_s$, consisting of n_p landmarks $\mathbf{x}_j \in \mathbb{R}^m, j = 1, \ldots, n_p$, GPA superimposes the shapes to their mean shape $\overline{\mathbf{X}}$ by optimally translating, rotating and scaling [15, 16]. If shapes are identical, the shapes adjusted though GPA coincide exactly. When all the shapes $\mathbf{X}_i$'s are translated to have the origin $[0, \ldots, 0]^T \in \mathbb{R}^m$ as a common center, the problem can be formulated to minimize the shape differences from all shapes $\mathbf{X}_i$'s to the mean shape $\overline{\mathbf{X}}$ w.r.t. scales and rotations, *i.e.*,

$$\begin{aligned} &\arg\min_{\mathbf{R}_i, s_i} \quad \sum_i \left\| s_i \mathbf{R}_i \mathbf{X}_i - \overline{\mathbf{X}} \right\|_F^2 \\ &\text{subject to} \quad \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}, \quad \sum_i \left\| s_i \mathbf{X}_i \right\|_F^2 = 1, \end{aligned} \tag{2.1}$$

where $\mathbf{R}_i$ is an orthogonal matrix, s_i is a scale factor and $\|\cdot\|_F$ is the Frobenius norm [32], *i.e.*, $\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}^T\mathbf{A})$. It is noted that the objective function has the minimum value when all s_i's and $\overline{\mathbf{X}}$ are zero, which is not a desired solution. We need the second constraint in (2.1) to avoid this trivial solution.

The procedure of GPA can be summarized as follows [15]:

1. All the shapes are moved to a common center, the origin $[0, \ldots, 0]^T \in \mathbb{R}^m$.
2. Scale each $\mathbf{X}_i, i = 1, \ldots, n_s$, by ζ so that
$$\zeta \sum_{i=1}^{n_s} \mathrm{tr}(\mathbf{X}_i \mathbf{X}_i^T) = 1.$$
3. To initialize the mean shape, set $\overline{\mathbf{X}} = \mathbf{X}_1$. For $i = 2, 3, \ldots, n_s$, rotate $\mathbf{X}_i$ to fit $\overline{\mathbf{X}}$, and re-evaluate $\overline{\mathbf{X}}$ as the mean of $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{n_s})$. Evaluate initial residual sum-of-squares $X_r = n_s(1 - \mathrm{tr}(\overline{\mathbf{X}}\overline{\mathbf{X}}^T))$ and set $s_i = 1, i = 1, 2, \ldots, n_s$.

4. For $i = 1, 2, \ldots, n_s$, rotate the current shape $\mathbf{X}_i$ to fit $\overline{\mathbf{X}}$ giving $\mathbf{X}'_i = \mathbf{R}_i\mathbf{X}_i$. Here, $\mathbf{R}_i = \mathbf{V}_i\mathbf{U}_i^T$, where $\mathbf{U}_i$ and $\mathbf{V}_i$ are obtained by the singular value decomposition of $\mathbf{X}_i\overline{\mathbf{X}}^T = \mathbf{U}_i\mathbf{\Gamma}_i\mathbf{V}_i^T$. After setting $\mathbf{X}_i = \mathbf{X}'_i$, compute the mean shape $\overline{\mathbf{X}}$.

5. For $i = 1, 2, \ldots, n_s$, scale $\mathbf{X}'_i = s_i\mathbf{X}_i$. Here, the scale factor s_i is

$$s_i = \sqrt{\frac{\mathrm{tr}(\mathbf{X}_i\overline{\mathbf{X}}^T)}{n_s\mathrm{tr}(\mathbf{X}_i\mathbf{X}_i^T)\mathrm{tr}(\overline{\mathbf{X}}\overline{\mathbf{X}}^T)}}.$$

   After setting $\mathbf{X}_i = \mathbf{X}'_i$, compute $\overline{\mathbf{X}}$ and new residual sum-of-squares X'_r.

6. If $X_r - X'_r > tolerance$, set $X_r = X'_r$ and go to step 4, otherwise the iteration stops.

2.2 EM-GPA Algorithm

2.2.1 Objective function

Before formulating the problem addressed in this chapter, we define notations used in this chapter. We define the *vectorization operator* $\mathbf{vec}(\mathbf{A})$, which transforms a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ into a vector as

$$\mathbf{vec}(\mathbf{A}) = [\mathbf{a}_1^T \ \mathbf{a}_2^T \ \ldots \ \mathbf{a}_{n_2}^T]^T,$$

where $\mathbf{a}_i \in \mathbb{R}^{n_1}$ is the ith column of $\mathbf{A}$. The Frobenious norm can be expressed as $||\mathbf{A}||_F^2 = \mathbf{vec}(\mathbf{A})^T\mathbf{vec}(\mathbf{A})$.

Now, we explain the proposed algorithm that extends GPA to the case with hidden variables by using the EM algorithm. To make the explanation simple, we describe EM-GPA only for the case of missing depth information, but this algorithm can be easily extended to a more general case, where some of the landmarks are missing, as shown in Section 2.4. Given a set of shapes $\mathbf{X}_i, i = 1, 2, \ldots, n_s$, GPA finds scales s_i and rotations $\mathbf{R}_i$ that most closely map the shapes $\mathbf{X}_i$ to its mean $\overline{\mathbf{X}}$ [15]. To express shapes including hidden variables, we define a shape $\mathbf{X}_i$ in 3D space, which consists of n_p landmarks, as

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{D}_i \\ \mathbf{h}_i \end{bmatrix} \in \mathbb{R}^{3\times n_p},$$

where each column of $\mathbf{D}_i \in \mathbb{R}^{2\times n_p}$ and $\mathbf{h}_i \in \mathbb{R}^{1\times n_p}$, which correspond to the coordinates (x, y) and z, respectively. $\mathbf{D}_i$ and $\mathbf{h}_i$ are translated so that $\mathbf{D}_i\mathbf{1} = \mathbf{0} \in \mathbb{R}^2$ and $\mathbf{h}_i\mathbf{1} = 0 \in \mathbb{R}$, where $\mathbf{1} = [1, \ldots, 1]^T \in \mathbb{R}^{n_p}$. When gathering position information on the landmarks from 2D images, $\mathbf{D}_i$ is observable whereas $\mathbf{h}_i$ is not. We name $\mathbf{h}_i$ as a hidden vector. Since shape deformation is not dramatic in many cases, such as for human face, we assume an aligned shape to be a Gaussian distribution, which can be expressed as

$$\begin{aligned} &p(\mathbf{X}_i) \propto \\ &\exp\left(-\frac{1}{2}\mathbf{vec}\left(s_i\mathbf{R}_i\mathbf{X}_i - \overline{\mathbf{X}}\right)^T \mathbf{\Sigma}^{-1}\mathbf{vec}\left(s_i\mathbf{R}_i\mathbf{X}_i - \overline{\mathbf{X}}\right)\right), \end{aligned} \tag{2.2}$$

where s_i, $\mathbf{R}_i$, $\overline{\mathbf{X}}$, and $\mathbf{\Sigma}$ are a scale, rotation, mean shape, and covariance matrix, respectively. Note that all the shapes are aligned to have a common center at the origin, *i.e.*, $\mathbf{X}_i\mathbf{1} = \mathbf{0} \in \mathbb{R}^3$. In this case, the covariance matrix ($\mathbf{\Sigma}$) can not be full rank. To deal with the singularity of the covariance matrix, we reformulate (2.2) as follows. Let $\mathbf{P}_N = \frac{1}{\sqrt{n_p}}[\mathbf{I}_3, \mathbf{I}_3, , \ldots, \mathbf{I}_3]^T \in \mathbb{R}^{3n_p\times 3}$ be the basis matrix corresponding to the translation and let $\mathbf{P} \in \mathbb{R}^{3n_p\times 3(n_p-1)}$ be an orthogonal matrix that satisfies $\mathbf{P}^T\mathbf{P}_N = \mathbf{0}$, then the reduced covariance matrix can be represented as $\mathbf{\Sigma}_R = \mathbf{P}^T\mathbf{\Sigma}\mathbf{P} \in \mathbb{R}^{3(n_p-1)\times 3(n_p-1)}$. We define the parameter set as $\mathbf{\Phi} = \{s_i, \mathbf{R}_i, \overline{\mathbf{X}}, \mathbf{\Sigma} | i = 1, 2, \ldots, n_s\}$. Then, the distribution of shapes $(\mathbf{D}_i, \mathbf{h}_i)$ becomes

$$p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}) \propto \exp\left(-\frac{1}{2}\mathbf{v}_i^T\mathbf{P}\mathbf{\Sigma}_R^{-1}\mathbf{P}^T\mathbf{v}_i\right)\delta\left(\mathbf{P}_N^T\,\mathbf{v}_i\right), \tag{2.3}$$

where $\mathbf{v}_i = \mathbf{vec}(s_i\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T - \overline{\mathbf{X}}) \in \mathbb{R}^{3n_p}$ which is the shape difference from the mean shape and $\delta(\cdot)$ is a delta function [33].

We find parameters that maximize the log-likelihood of $p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi})$ by using the EM algorithm under the assumption that all the shapes $\mathbf{X}_i$ are independent. Since the EM algorithm iteratively run E-step and M-step. In E-step, we estimate the distribution of hidden vector $\mathbf{h}_i$ in the form of $p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}^{old})$ where the superscript old denotes the parameter set obtained from the previous M-step in the EM iteration procedure. In M-step, the maximum likelihood (ML) solution of the parameters is obtained by using the distribution of hidden variables computed from E-step and additional constraints as follows.

$$\begin{aligned}
\mathbf{\Phi}^* &= \arg\max_{\mathbf{\Phi}} \sum_i \int \ln p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}) p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}^{old}) d\mathbf{h}_i \\
\text{subject to} \quad & E_{\mathbf{h}_i}\left[\mathbf{R}_i \begin{bmatrix}\mathbf{D}_i \\ \mathbf{h}_i\end{bmatrix} \overline{\mathbf{X}}^T\right] \in \mathbf{S}_+^3, \\
& E_{\mathbf{h}_i}\left[\sum_i \left\| s_i \begin{bmatrix}\mathbf{D}_i \\ \mathbf{h}_i\end{bmatrix}\right\|_F^2\right] = 1,
\end{aligned} \tag{2.4}$$

where $\mathbf{S}_+^3$ is a set of three dimensional positive semi-definite matrices (PSDs). The constraints in (2.4), which will be called the *GPA constraints*, force s_i and $\mathbf{R}_i$ to be determined in a manner that is analogous to GPA in (2.1). Note that the second constraint is equivalent to the second constraint in (2.1). The first constraint in (2.4), which is a rotation constraint, can be shown to be equivalent to the solution of the following optimization problem:

$$\min_{\mathbf{R}_i} E_{\mathbf{h}_i}\left[\left\| s_i \mathbf{R}_i \begin{bmatrix}\mathbf{D}_i \\ \mathbf{h}_i\end{bmatrix} - \overline{\mathbf{X}}\right\|_F^2\right],$$

where $E_{\mathbf{h}_i}[\|s_i\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T - \overline{\mathbf{X}}\|_F^2] = E_{\mathbf{h}_i}[\|s_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\|_F^2] - 2E_{\mathbf{h}_i}[\mathrm{tr}(s_i\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\overline{\mathbf{X}}^T)] + E_{\mathbf{h}_i}[\|\overline{\mathbf{X}}\|_F^2]$. This is equivalent to maximizing $E_{\mathbf{h}_i}[\mathrm{tr}(\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\overline{\mathbf{X}}^T)]$, and it can be readily proved that the optimal rotation makes $E_{\mathbf{h}_i}[\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\overline{\mathbf{X}}^T]$ positive semi-definite [15]. In fact, the feasible rotation for the constant in (2.4) is unique when $[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\overline{\mathbf{X}}^T$

is full rank, which is true in most of the cases. These constraints make the expectations of 3D shapes aligned w.r.t. the Euclidean measure, which is the core of GPA.

2.2.2 E-step

Here, we explain how to estimate the distribution of $\mathbf{h}_i$ in E-step. From Bayes' theorem, the distribution of $\mathbf{h}_i$ can be written as follows:

$$p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}^{old}) = \frac{p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}^{old})}{\int p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}^{old})d\mathbf{h}_i}, \tag{2.5}$$

From now on, we will omit the superscript (old) if no confusion arises. Since we assumed that $\mathbf{X}_i$ is Gaussian, $\mathbf{h}_i$ is also Gaussian, which can be represented as

$$p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}) \propto \exp\left(-\frac{1}{2}(\mathbf{h}_i - \bar{\mathbf{h}}_i)\mathbf{C}_i^{-1}(\mathbf{h}_i - \bar{\mathbf{h}})^T\right), \tag{2.6}$$

where $\bar{\mathbf{h}}_i$ and $\mathbf{C}_i$ are the mean and covariance matrix of $\mathbf{h}_i$ conditioned on $\mathbf{D}_i$, respectively. As for shapes $\mathbf{X}_i$ in (2.2), the hidden vector $\mathbf{h}_i$ should have origin at the center, *i.e.*, $\mathbf{h}_i\mathbf{1} = 0$ and its covariance matrix $\mathbf{C}_i$ becomes singular. To address the singularity of the covariance matrix of $\mathbf{h}_i$, we reformulate (2.6) as follows. Let $\mathbf{P}_{N_h} = \frac{1}{\sqrt{n_p}}[1,\ 1,\ ,\ldots,\ 1]^T \in \mathbb{R}^{n_p}$ be the basis vector for the translation of $\mathbf{h}_i$ and let $\mathbf{P}_h \in \mathbb{R}^{n_p\times(n_p-1)}$ be an orthogonal matrix that satisfies $\mathbf{P}_h^T\mathbf{P}_{N_h} = \mathbf{0}$, then $\mathbf{C}_i$ can be expressed as $\mathbf{P}_h\mathbf{C}'_i\mathbf{P}_h^T$ and the distribution of $\mathbf{h}_i$ can be represented as

$$p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}) \propto \exp\left(-\frac{1}{2}\tilde{\mathbf{h}}_i\mathbf{P}_h\mathbf{C}_i'^{-1}\mathbf{P}_h^T\tilde{\mathbf{h}}_i^T\right)\delta\left(\mathbf{P}_{N_h}^T\tilde{\mathbf{h}}_i\right), \tag{2.7}$$

where $\tilde{\mathbf{h}}_i = \mathbf{h}_i - \bar{\mathbf{h}}_i$, which is the difference of $\mathbf{h}_i$ from its mean in the kth step. By comparing (2.3) and (2.7), we obtain the following proposition.

Proposition 1. *The mean* $\bar{\mathbf{h}}_i$ *and covariance matrix* $\mathbf{C}'_i$ *of* $\mathbf{h}_i$ *can be found from (2.3),* *(2.5), and (2.7) as*

$$\begin{aligned}
&\mathbf{C}'_i = \frac{1}{{s_i}^2}\left(\mathbf{\Psi}_i{}^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{\Psi}_i\right)^{-1}, \\
&\bar{\mathbf{h}}_i = s_i\mathbf{vec}\left(\overline{\mathbf{X}} - s_i[\mathbf{R}_{i1}\,\mathbf{R}_{i2}]\mathbf{D}_i\right)^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{\Psi}_i\mathbf{C}'_i\mathbf{P}_h^T, \qquad (2.8) \\
&\mathbf{R}_i = \left[\mathbf{R}_{i1}\,\mathbf{R}_{i2}\,\mathbf{R}_{i3}\right], \qquad \mathbf{\Psi}_i = \left(\mathbf{P}_h \otimes \mathbf{R}_{i3}\right),
\end{aligned}$$

where $\mathbf{R}_{ij}$ is the jth column of $\mathbf{R}_i$ and the symbol $\otimes$ denotes the Kronecker product [34].

Proof. See Appendix A. □

We will calculate ($\mathbf{R}_i$, s_i, $\overline{\mathbf{X}}$, $\mathbf{\Sigma}$) in M-step.

2.2.3 M-step

In this M-step, the ML solution of parameters $\mathbf{\Phi}$ is obtained by using the distribution of hidden variables computed from E-step. The objective function can be expressed as

$$\begin{aligned}
&J(\mathbf{\Phi}|\mathbf{\Phi}^{old}) = -n_s \ln|\mathbf{\Sigma}_R| \\
&-\sum_i s_i^2 \mathrm{tr}\left(\mathbf{\Psi}_i^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{\Psi}_i\mathbf{C}'_i\right) - \sum_i \bar{\mathbf{v}}_i^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\bar{\mathbf{v}}_i, \\
&\text{subject to} \qquad E_{\mathbf{h}_i}\left[\mathbf{R}_i \begin{bmatrix}\mathbf{D}_i \\ \mathbf{h}_i\end{bmatrix} \overline{\mathbf{X}}^T\right] \in \mathbf{S}_+^3, \qquad (2.9) \\
&\qquad\qquad E_{\mathbf{h}_i}\left[\sum_i s_i^2 \left\|\begin{bmatrix}\mathbf{D}_i \\ \mathbf{h}_i\end{bmatrix}\right\|_F^2\right] = 1,
\end{aligned}$$

where $\bar{\mathbf{v}}_i = \mathbf{vec}(s_i\mathbf{R}_i[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T]^T - \overline{\mathbf{X}})$, which is the expected value of $\mathbf{v}_i$ with respect to $\mathbf{h}_i$. Finally, we can obtain the solution by maximizing the objective function (2.9) with respect to parameters $\mathbf{\Phi}$ as follows.

Proposition 2. *If $\overline{\mathbf{X}}[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T]$ is not singular, the solution that maximizes $J(\mathbf{\Phi}|\mathbf{\Phi}^{old})$ in (2.9) with respect to $\mathbf{R}_i$ is*

$$\mathbf{R}_i = \mathbf{U}_i\mathbf{V}_i^T, \qquad (2.10)$$

where $\mathbf{U}_i$ and $\mathbf{V}_i$ are obtained from the singular value decomposition of the matrix $\overline{\mathbf{X}}[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T]$.

Proof. Note that $E_{\mathbf{h}_i}[\mathbf{R}_i[\mathbf{D}_i^T\ \mathbf{h}_i^T]^T\overline{\mathbf{X}}^T] = \mathbf{R}_i[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T]^T\overline{\mathbf{X}}^T = \mathbf{R}_i\mathbf{V}_i\mathbf{\Gamma}_i\mathbf{U}_i^T$. In order for this expression to be positive semi-definite, $\mathbf{R}_i\mathbf{V}_i$ should be equal to $\mathbf{U}_i$, if $\mathbf{\Gamma}_i$ is full rank. Hence $\mathbf{R}_i = \mathbf{U}_i\mathbf{V}_i^T$, which is the only feasible solution. □

In most cases, $\overline{\mathbf{X}}[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T]$ is not singular. If $\mathbf{\Gamma}_i$ has zero diagonal entries, $\mathbf{R}_i = \mathbf{U}_i\mathbf{M}\mathbf{V}_i^T$ is also feasible, where $\mathbf{M}$ is a diagonal matrix whose diagonal elements are 1 for the entries that corresponds to non-zero $\mathbf{\Gamma}_i$ elements, and ± 1 for the other diagonal entries. In this case, $\mathbf{M}$ is chosen so that the likelihood is maximized.

Proposition 3. *The solution that maximizes $J(\mathbf{\Phi}|\mathbf{\Phi}^{old})$ in (2.9) with respect to s_i is the eigenvector of the smallest eigenvalue obtained from the following generalized eigenvalues problem:*

$$\mathbf{G}\mathbf{s} = \lambda\mathbf{F}\mathbf{s} \quad \textit{subject to} \quad \mathbf{s}^T\mathbf{F}\mathbf{s} = 1, \tag{2.11}$$

where

$$\begin{aligned}
\mathbf{G} &= \begin{cases} \mathbf{G}_{ii} &= \mathrm{tr}(\mathbf{\Psi}_i^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{\Psi}_i\mathbf{C}_i') \\ & \quad + \left(1 - \frac{1}{n_s}\right)^T \mathbf{q}_i^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{q}_i \\ \mathbf{G}_{ij} &= -\frac{1}{n_s}\mathbf{q}_i^T\mathbf{P}\mathbf{\Sigma}_R{}^{-1}\mathbf{P}^T\mathbf{q}_j \quad \textit{for } i \neq j \end{cases} \\
\mathbf{q}_i &= \mathbf{vec}\left(\mathbf{R}_i\begin{bmatrix}\mathbf{D}_i \\ \bar{\mathbf{h}}_i\end{bmatrix}\right) \\
\mathbf{F} &= \begin{cases} \mathbf{F}_{ii} &= \left\|\begin{bmatrix}\mathbf{D}_i \\ \bar{\mathbf{h}}\end{bmatrix}\right\|_F^2 + \mathrm{tr}\left(\mathbf{C}_i'\right) \\ \mathbf{F}_{ij} &= 0 \quad \textit{for } i \neq j \end{cases}
\end{aligned} \tag{2.12}$$

Proof. See Appendix A. □

Also the solution that maximizes $J(\mathbf{\Phi}|\mathbf{\Phi}^{old})$ in (2.9) with respect to $\overline{\mathbf{X}}$ and $\mathbf{\Sigma}$ is found by differentiating the objective function in (2.9) w.r.t. $\overline{\mathbf{X}}$ and $\mathbf{\Sigma}$ and equating them to zero, which are

$$\overline{\mathbf{X}} = \frac{1}{n_s}\sum_i s_i \mathbf{R}_i \begin{bmatrix} \mathbf{D}_i \\ \bar{\mathbf{h}}_i \end{bmatrix}, \tag{2.13}$$

$$\mathbf{\Sigma}_R = \frac{1}{n_s}\sum_i \mathbf{P}^T \left(s_i^2 \mathbf{\Psi}_i \mathbf{C}' \mathbf{\Psi}_i^T + \bar{\mathbf{v}}_i \bar{\mathbf{v}}_i^T\right) \mathbf{P}.$$

These E-step and M-step constitute the parameter updating rule for $\mathbf{\Phi}$, which are executed iteratively until the parameters converge.

2.3 Implementation Considerations for EM-GPA

2.3.1 Preprocessing stage

Here, we investigate the effect of the covariance matrix during the process of parameter update. The covariance matrix represents the modes for shape variations that is separated from rigid transforms. However, if the rotations and scales are not correct, then the covariance matrix derived from them will also be incorrect, and the separation between rigid transformation and non-rigid deformation may not be successful. Here, we want to obtain the covariance matrix that only includes deformation information of the Procrustes aligned 3D shapes, removing the effect of rotation and scale. However, since the EM framework is an iterative procedure, we have to align the shapes by using the rotations and scales estimated from the current step and calculate the covariance matrix using the misaligned shapes. Therefore, in early iteration steps, the covariance matrix is usually corrupted by incorrect information on rotations and scales, which may reduce the need to update the rotations and scales properly in the next steps. That is, in the process

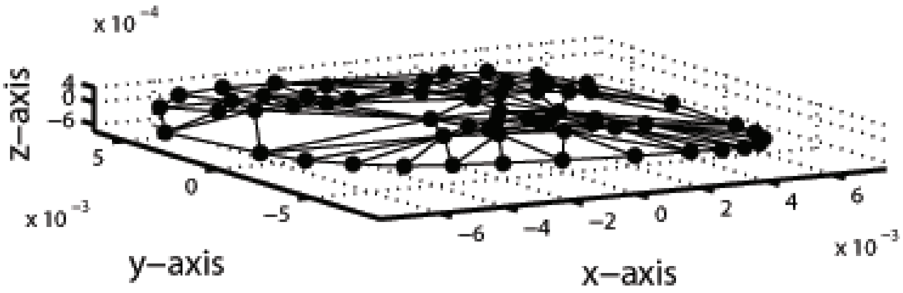


Figure 2.1: An example of an incorrectly reconstructed mean shape.

of maximizing the log-likelihood, the EM algorithm may adjust the covariance matrix too much and rotations and scales not properly. It makes EM-GPA fail to estimate the distribution of hidden vector correctly.

Figure 2.1 shows the estimated 3D mean shape by EM-GPA with randomly selected initial parameters. We can see that the estimated mean shape is almost two dimensional. Since this is due to the incorrect estimation of the covariance matrix, which is corrupted by the information on rotations and scales, we need to obtain relatively accurate rotations and scales before estimating the covariance matrix. To do this, we introduce another EM algorithm for preprocessing, where we set the covariance matrix to diagonal with the same variance, *i.e.*, $\mathbf{\Sigma}_R = \sigma^2 \mathbf{I}_{3(n_p-1)}$. Then (2.3) can be represented as

$$p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}) \propto \exp\left(-\frac{1}{2\sigma^2}\mathbf{v}_i^T\mathbf{v}_i\right)\delta\left(\mathbf{P}_N^T\,\mathbf{v}_i\right). \tag{2.14}$$

where $\mathbf{v}_i = \mathbf{vec}(s_i\mathbf{R}_i[\mathbf{D}_i^T\mathbf{h}_i^T]^T - \overline{\mathbf{X}}) \in \mathbb{R}^{3n_p}$ which is the shape difference from the mean shape, and $\delta(\cdot)$ is a delta function [33]. Although this formulation does not reflect the modes in shape variation, we can obtain approximate rotations and scales which minimize the Euclidean measure between all shapes and their mean shape (the Frobenius norm of the shape differences $\mathbf{v}_i$). Using this rotations and scales as initial parameters, we can perform EM-GPA more effectively. We call this procedure as the preprocessing stage and it is summarized as follows.

- E-step

$$\begin{aligned} \mathbf{C}'_i &= \frac{\sigma^2}{{s_i}^2}\mathbf{I}_{n_p-1}, \\ \bar{\mathbf{h}}_i &= \frac{1}{s_i}{\mathbf{R}_{i3}}^T\left(\overline{\mathbf{X}} - s_i[\mathbf{R}_{i1}\ \mathbf{R}_{i2}]\mathbf{D}_i\right). \end{aligned} \tag{2.15}$$

- M-step

$$\begin{aligned} \mathbf{R}_i &= \mathbf{U}_i\mathbf{V}_i^T, \qquad \overline{\mathbf{X}}\left[\mathbf{D}_i^T\ \bar{\mathbf{h}}_i^T\right] = \mathbf{U}_i\mathbf{\Gamma}_i\mathbf{V}_i^T, \\ s_i &= \sqrt{\frac{f_i^2}{\sum_{j=1}^{n_s}\frac{f_j^2}{g_j}g_i^2}}, \\ \overline{\mathbf{X}} &= \frac{1}{n_s}\sum_i s_i\mathbf{R}_i\begin{bmatrix}\mathbf{D}_i \\ \bar{\mathbf{h}}_i\end{bmatrix}, \\ \sigma &= \frac{1}{3n_s(n_p-1)}\sum_i\left(s_i^2\mathrm{tr}\left(\mathbf{C}'\right) + \bar{\mathbf{v}}_i^T\bar{\mathbf{v}}_i\right), \end{aligned} \tag{2.16}$$

where $f_j = \mathrm{tr}(\mathbf{R}_j[\mathbf{D}_j^T\ \bar{\mathbf{h}}_j^T]^T\overline{\mathbf{X}}^T)$ and $g_j = \|[\mathbf{D}_j^T\ \bar{\mathbf{h}}_j^T]^T\|_F^2 + \mathrm{tr}(\mathbf{C}'_j)$.

We omit the derivation of this procedure, because the solution procedure for (2.14) is a special case of the procedure for (2.3). The preprocessing stage iterates until $\|\overline{\mathbf{X}} - \overline{\mathbf{X}}^{old}\|_F$ becomes less than a prespecified $threshold$. After convergence, we use the estimated rotation $\mathbf{R}_i$, scale s_i, 3D mean shape $\overline{\mathbf{X}}$, and reduced covariance matrix $\sigma^2\mathbf{I}_{3(n_p-1)}$ as the initial parameters for EM-GPA.

2.3.2 Small update rate for the covariance matrix

For a similar reason as in Section 2.3.1, it is better to update the covariance matrix with a small update rate. As explain in Section 2.3.1, the rotations and scales must be updated ”*with a larger update rate*” than the covariance matrix. It can be achieved as follows.

Proposition 4. *Let*

$$\mathbf{Z} = \frac{1}{n_s}\sum_i \mathbf{P}^T\left(s_i^2\mathbf{\Psi}_i\mathbf{C}'\mathbf{\Psi}_i^T + \bar{\mathbf{v}}_i\bar{\mathbf{v}}_i^T\right)\mathbf{P}, \tag{2.17}$$

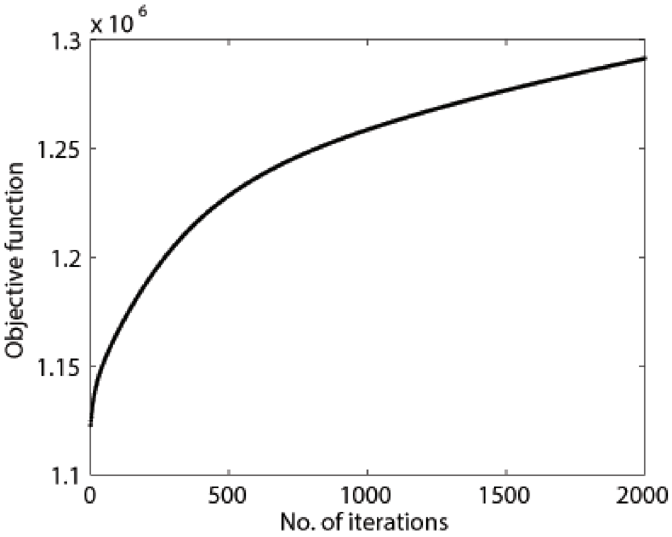


Figure 2.2: The value of the objective function of the EM-GPA.

then for any $0 \leq \alpha \leq 1$*,* $\boldsymbol{\Sigma}_R = \alpha\mathbf{Z} + (1-\alpha){\boldsymbol{\Sigma}_R}^{old}$ *increases the log-likelihood of EM-GPA.*

Proof. See Appendix A. □

In the experiments in Section 2.4, α was set to 0.01. Figure 2.2 shows an example of the objective function in (2.9) during the iterative process.

The EM-GPA algorithm is summarized in Algorithm 1.

2.4 Experiments

We perform three experiments to demonstrate the performance of EM-GPA. In the first experiment, we show that EM-GPA can align 2D shapes by taking the missing information into consideration. In the second experiment, we estimated the mean and covariance matrix of 3D shapes and compared them to the ground truth. In the third experiment, we constructed a 2D+3D AAM [35, 36] by using a 3D shape model obtained by EM-GPA

Algorithm 1 EM-GPA

```
Require: 2D shapes $\mathbf{D}_i \in \mathbb{R}^{2\times n_p}, i = 1, 2, \ldots, n_s$
Ensure: A set of parameters for EM-GPA $\mathbf{\Phi} = \{\mathbf{R}, \mathbf{s}, \overline{\mathbf{X}}, \mathbf{\Sigma}_R\}$
$\mathbf{D}_i = \mathbf{D}_i - \frac{1}{n_p}\mathbf{D}_i\mathbf{1}\mathbf{1}^T, i = 1, 2, \ldots, n_s$ (Translate to the origin).
Initialize a set of parameters $\mathbf{\Phi}_p = \{\mathbf{R}, \mathbf{s}, \overline{\mathbf{X}}, \sigma\}$ by using the method in Section 2.4.
repeat
    Calculate $\bar{\mathbf{h}}_i$ and $\mathbf{C}'_i$ using (2.15).
    Calculate $\mathbf{R}_i$, $s_i$, $\overline{\mathbf{X}}$, and $\sigma$ using (2.16).
until convergence
Initialize $\mathbf{\Phi}$ using $\mathbf{\Phi}_p$, where $\mathbf{\Sigma}_R$ = $\sigma^2\mathbf{I}_{3(n_p-1)}$.
repeat
    Calculate $\bar{\mathbf{h}}_i$ and $\mathbf{C}'_i$ using (2.8).
    Calculate $\mathbf{R}_i$, $s_i$, $\overline{\mathbf{X}}$, and $\mathbf{\Sigma}_R$ using (2.10), (2.11), (2.13), and (2.17).
until convergence
```

and compared it with a 2D AAM [37] and another 2D+3D AAM where the 3D shape model was constructed using real 3D shapes.

We initialize the parameters of EM-GPA as follows. To initialize $\mathbf{R}_i$, we first generated a 3×3 matrix $\mathbf{R}_{initial}$, whose elements were pseudorandom numbers drawn from the standard normal distribution. Then, by the QR decomposition of $\mathbf{R}_{initial}$, we obtained an orthogonal matrix $\mathbf{R}_i$. We initialized the scales s_i as $s_i = \frac{1}{\|n_s \mathbf{X}_i\|_F}$, where missing (hidden) variables $\mathbf{h}_i$'s in $\mathbf{X}_i$ were set to zero. $\overline{\mathbf{X}}$ was calculated using s_i and $\mathbf{R}_i$ as $\overline{\mathbf{X}} = \frac{1}{n_s}\sum_{i=1}^{n_s} s_i \mathbf{R}_i \mathbf{X}_i$. The preprocessing stage continues until $\|\overline{\mathbf{X}} - \overline{\mathbf{X}}\|_F$ becomes less than 10^{-5}, and then the optimization process in EM-GPA iterates for 100 times.

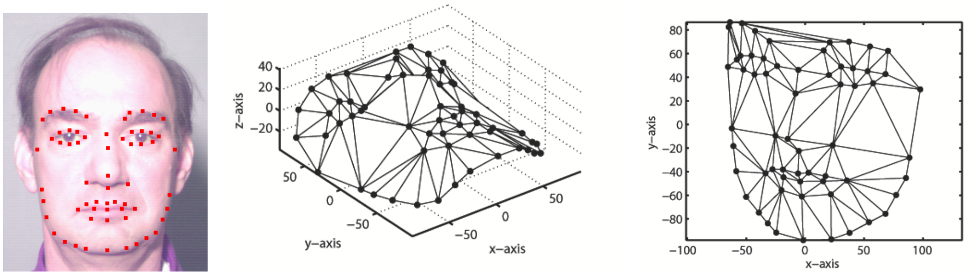
2.4.1 Shape alignment with the missing information

For the experiment, we generated arbitrary rotated 2D shapes as follows. From the FRGC database, we selected 400 subjects and manually located 62 landmarks in each subject to construct a set of 3D shapes $\mathbf{X}_i^*$ as shown in Figures 2.3(a) and Figure 2.3(b). Note that the centroid of each 3D shape is moved to the origin. Then, we generated a set of 2D shapes $\mathbf{D}_i$ by randomly rotating the 3D shapes in the range of $\left[0, \frac{\pi}{4}\right]$ and projecting them in the z direction as shown in Figure 2.3(c).

We investigate how well EM-GPA aligns 2D shapes by taking the missing information into consideration. We calculated the error distances from the aligned 2D shapes $\hat{\mathbf{D}}_{aligned,\ i}$ by EM-GPA to the ground truth $\mathbf{D}^*_{aligned,\ i}$, which was obtained by performing GPA to real 3D shapes $\mathbf{X}_i^*$ and then projecting them in the z direction, *i.e.*,

$$\mathbf{D}^*_{aligned,\ i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \times \mathbf{X}^*_{aligned,\ i}.$$

The reconstructed 3D shape is represented as $\hat{\mathbf{X}}_i = [\mathbf{D}_i^T \ \bar{\mathbf{h}}_i^T]^T$ and it can be aligned as $\hat{\mathbf{X}}_{aligned,\ i} = s_i \mathbf{R}_i \hat{\mathbf{X}}_i$. However, since there is a rotation ambiguity between the estimated 3D mean shape $\hat{\overline{\mathbf{X}}}$ and the ground truth 3D mean shape $\overline{\mathbf{X}}^*$, we calculated a rota-



(a) 62 landmarks (b) 3D Shape. (c) Generated 2D Shape.

Figure 2.3: An example of landmark points in a face. 62 landmarks, the corresponding 3D shape and generated 2D shape by projecting a randomly rotated 3D shape in z direction.

tion $\mathbf{R}_c$ between them by using Procrustes analysis (PA) to superimpose these 3D mean shapes. To obtain the aligned 2D shapes $\hat{\mathbf{D}}_{aligned,\ i}$, we rotated and projected $\hat{\mathbf{X}}_{aligned,\ i}$ in the z direction, *i.e.*,

$$\hat{\mathbf{D}}_{aligned,\ i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \times \mathbf{R}_c \hat{\mathbf{X}}_{aligned,\ i}.$$

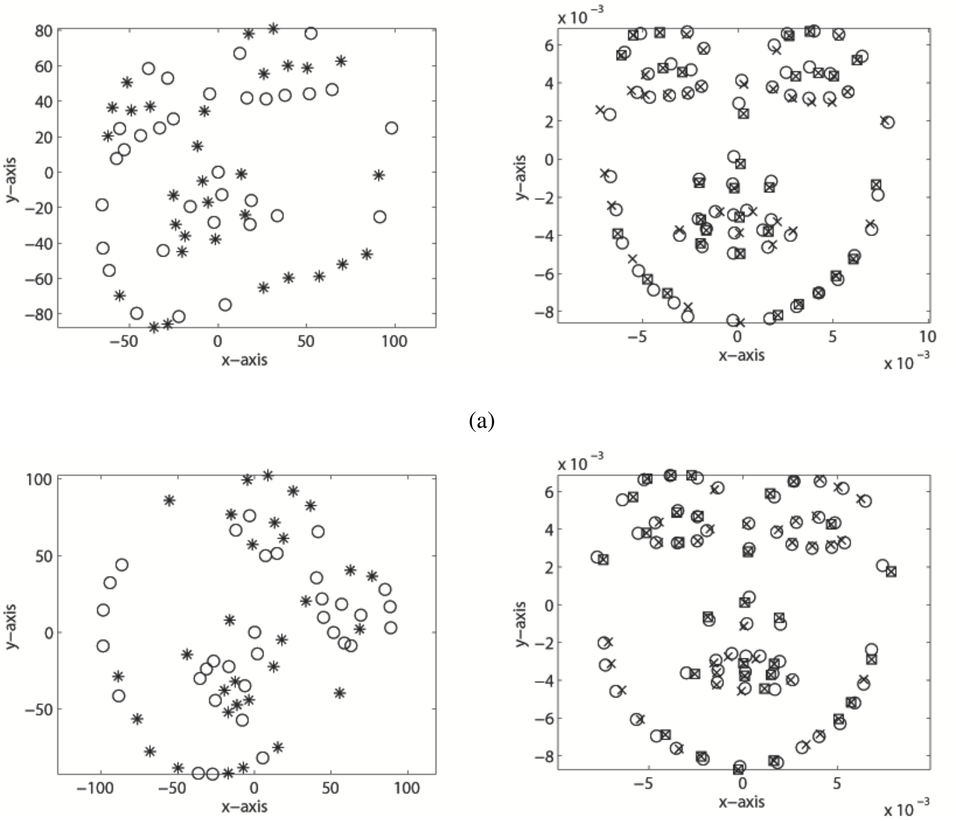
Then, to evaluate alignment performance, we used the alignment error calculated as

$$e_{aligned,\ i} = \frac{\left\| \hat{\mathbf{D}}_{aligned,\ i} - \mathbf{D}^*{}_{aligned,\ i} \right\|_F}{\left\| \mathbf{D}^*{}_{aligned,\ i} \right\|_F}.$$

As can be seen in Figure 2.4, the aligned 2D shape $\hat{\mathbf{D}}_{aligned,\ i}$ is very similar to the aligned ground truth 2D shape $\mathbf{D}^*_{aligned,\ i}$, and the alignment error $e_{aligned}$ is 0.0572 on average. It demonstrates that EM-GPA can successfully align arbitrary rotated 2D shapes by taking the missing information into consideration.

2.4.2 3D shape modeling

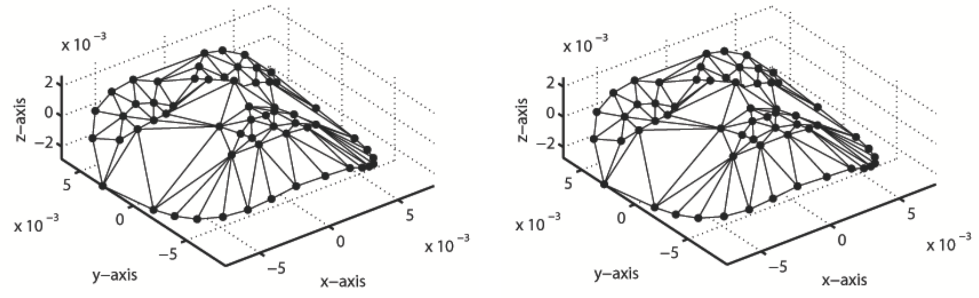
Here, we check the validity of the estimated mean and covariance matrix. To do this, we obtained the ground truth 3D mean shape $\overline{\mathbf{X}}^*$ and the covariance matrix $\mathbf{\Sigma}^*$ of



(a)

(b)

Figure 2.4: Two examples of alignment. The images in the left column are input shapes, where observed landmarks are marked by ‘o’ and missing landmarks are marked by ‘*’. The images on the right are the results aligned by EM-GPA, where the ground truth are marked by ‘o’ and the aligned landmarks are marked by ‘x’, and missing landmarks are marked by ‘□’.



(a) EM-GPA

(b) Ground truth.

Figure 2.5: The estimated mean shape and the ground truth shape obtained from the FRGC database.

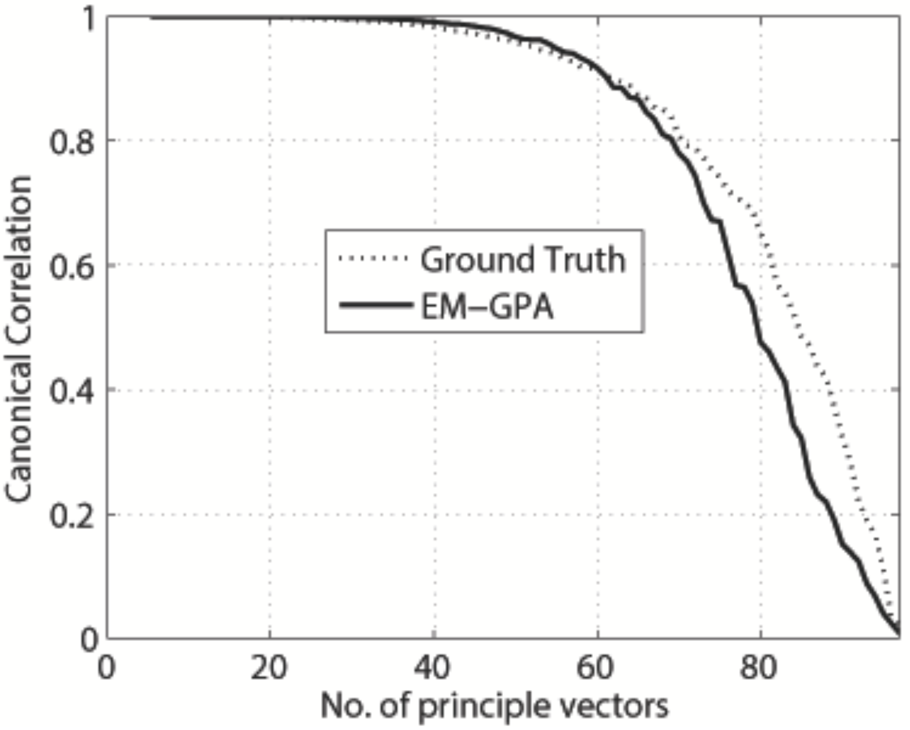


Figure 2.6: The canonical correlation obtained from the FRGC database.

$\mathbf{vec}(\mathbf{X}^*_{aligned,\ i})$ by performing GPA on 3D shapes $\mathbf{X}^*_i$'s. Since there was a rotation ambiguity between the estimated 3D mean shape $\hat{\bar{\mathbf{X}}}$ and the ground truth 3D mean shape $\overline{\mathbf{X}}^*$, we calculated a rotation $\mathbf{R}_c$ between them by using Procrustes analysis (PA) to superimpose these 3D mean shapes. The estimation error of $\hat{\bar{\mathbf{X}}}$ with respect to $\bar{\mathbf{X}}^*$ was calculated as

$$e = \frac{\left\| \mathbf{R}_c \hat{\bar{\mathbf{X}}} - \bar{\mathbf{X}}^* \right\|_F}{\left\| \bar{\mathbf{X}}^* \right\|_F}. \tag{2.18}$$

As can be seen in Figure 2.5, the estimated 3D mean shape $\hat{\bar{\mathbf{X}}}$ is very similar to the ground truth 3D mean shape $\overline{\mathbf{X}}^*$, and the estimation error e is 0.0052.

Also, we eliminated the rotation ambiguity between the estimated covariance matrix $\hat{\mathbf{\Sigma}}$ and the ground truth $\mathbf{\Sigma}^*$ by $\hat{\mathbf{\Sigma}}_c = (\mathbf{I}_{n_p} \otimes \mathbf{R}_c)\hat{\mathbf{\Sigma}}(\mathbf{I}_{n_p} \otimes \mathbf{R}_c)^T$. The performance of the estimated covariance matrix $\hat{\mathbf{\Sigma}}_c$ was evaluated in terms of the canonical correlations, which are cosines of principal angles between two linear subspaces $\mathcal{L}_1$ and $\mathcal{L}_2$ that are spanned by $\hat{\mathbf{\Sigma}}_c$ and $\mathbf{\Sigma}^*$. The canonical correlation are defined for $k = 1, \dots, n_e$, *i.e.*,

$$\begin{aligned} \cos\Theta_k &= \max_{\mathbf{u}\in\mathcal{L}_1} \max_{\mathbf{v}\in\mathcal{L}_2} \mathbf{u}_k^T \mathbf{v}_k \\ \text{subject to} \qquad \mathbf{u}_k^T \mathbf{u}_k &= \mathbf{v}_k^T \mathbf{v}_k = 1, \\ \mathbf{u}_i^T \mathbf{u}_k &= \mathbf{v}_i^T \mathbf{v}_k = 0,\ i = 1, \dots, k-1. \end{aligned} \tag{2.19}$$

Here the vectors in $\{\mathbf{u}_1, \dots, \mathbf{u}_{n_e}\}$ and $\{\mathbf{v}_1, \dots, \mathbf{v}_{n_e}\}$ are the principle vectors constituting $\mathcal{L}_1$ and $\mathcal{L}_2$ spaces, respectively. The canonical correlations show the proximity between the vectors that constitute the two linear subspaces. We used the singular value decomposition (SVD) to solve this problem [38]. We calculated canonical correlations between the two linear subspaces, which was consisted of the eigenvectors corresponding to the n_e largest eigenvalues in descending order. n_e was set to 114 to satisfy the relation $(\sum_{i=1}^{n_e} \lambda_i^2)/\sum_{i=1}^{3n_p} \lambda_i^2 > 0.99$, where λ_i is the eigenvalues of $\mathbf{\Sigma}$. About 80% of the canonical correlations between the eigenvectors of $\hat{\mathbf{\Sigma}}_c$ and $\mathbf{\Sigma}^*$ are bigger than 0.85.

To see how close $\hat{\mathbf{\Sigma}}_c$ is to $\mathbf{\Sigma}^*$, we also calculated another canonical correlations between the eigenvectors of two covariance matrices computed from the two sets of 200 real 3D shapes in the FRGC database. As can be seen in Figure 2.6, the canonical correlations computed from the covariance matrices of two real data sets show similar characteristics to the canonical correlation computed between those of EM-GPA and a real data set. From this result, we conclude that $\hat{\mathbf{\Sigma}}$ is a reasonable estimate of $\mathbf{\Sigma}^*$.

2.4.3 2D+3D active appearance models

Although AAM can be used to fit a 2D image of a 3D object, it is a difficult problem and becomes much more challenging when the 3D object undergoes a large rotation from the frontal position. This is because 2D AAM can generate model instances that are not possible in reality [35, 36]. Xiao *et al*. Xiao *et al.*[35, 36] proposed 2D+3D AAM, which constrains the model parameters of 2D AAM based on a 3D shape model. To show how the virtual 3D shape model constructed by EM-GPA can be applied in AAM fitting, we constructed a 2D+3D AAM based on the virtual 3D shape model, and compared it to a 2D AAM [37] and a 2D+3D AAM based on a real 3D shape model.

To construct a 2D shape model, we used the PF07 database [39], which includes 100 male and 100 female subjects captured in 5 different poses. The pose variation consists of front, left, right, up, and down, and the angle between the frontal pose and the other poses is 22.5 degrees. We selected 200 images, one for each subject, allocating 40 images in each pose, and located 62 landmarks in each image. We also constructed two 3D shape models (virtual/real) based on the FRGC database. Each 2D+3D AAM was built by using four 2D shape basis vectors, 122 appearance basis vectors and three 3D shape basis vectors, which accounted for 85%, 95% and 40% of their corresponding variations, respectively. In all the cases, each appearance image $A(\mathbf{X})$ [37] used to construct an appearance model had a resolution of 40×40 pixels for each of RGB, resulting in

$A(\mathbf{X}) \in \mathbb{R}^{4800}$. We tested AAM fitting for 100 test images from the PF07 database, which were not used in training the three AAMs, and the number of faces in each pose was 20. Following [40], we used the 2D mean shapes as the initial shape, whose position was displaced from the ground-truth position by $[-20, -10, 0, 10, 20]$ pixels in the x direction. Because the 2D mean shape was 40 pixels wide and a face in the test images was about 160 pixels wide on average, the 2D mean shape was scaled up by 4 times. The AAM fitting process stopped when the maximum shape displacement between two consecutive iterations was less than 0.5 pixel. We define the average error of landmarks from their ground-truth positions as

$$e_{landmark} = \frac{1}{n_p} \sum_{j=1}^{n_p} \sqrt{(x_j - x_j^*)^2 + (y_j - y_j^*)^2},$$

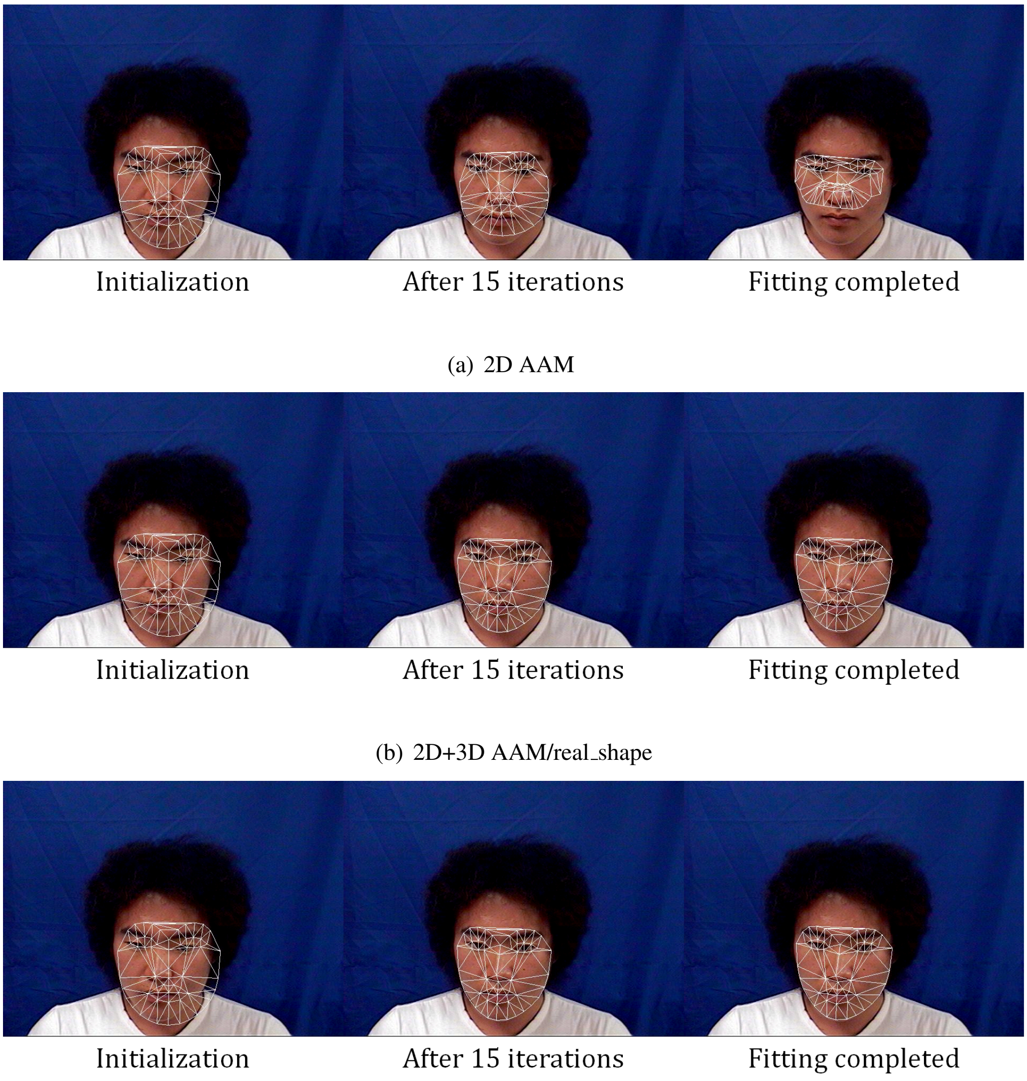
where (x_j, y_j) is the position of landmark j in the estimated shape, and (x_j^*, y_j^*) is the position of landmark j in the ground truth, which was obtained manually. Also, we computed the rate of successful convergence, which corresponds to the case of $e_{landmark}$ less than 6 pixels. Figure 2.7 shows an example where the model fitting was successful for 2D+3D AAM/real_shape and 2D+3D AAM/virtual_shape, but not for 2D AAM. Table 2.1 shows the fitting performance of three AAMs. We can see that both of 2D+3D AAMs are about 17% better than the 2D AAM in the convergence rate. In Table I, $e_{landmark}$ was computed by using only the data of successful convergence cases. Table 2.2 shows the rate of successful convergence of the three algorithms for pose variations. We can see that the two 3D models show better convergence rate compared to 2D AAM regardless of pose variation, and the convergence rate of 2D+3D/virtual_shape is almost the same as that using the 2D+3D AAM/real_shape. This demonstrates that EM-GPA can construct an accurate 3D shape model, which can be applied in the algorithms that need a 3D shape model.

Table 2.1: AAM fitting performance

Algorithm	Successful convergence (%)	$e_{landmark}$ (Final/Initial)	Average number of iterations
2D AAM [37]	60.6	4.7/17.9	25.1
2D+3D AAM [35, 36]/real_shape	78.0	4.5/18.2	30.6
2D+3D AAM/virtual_shape	77.8	4.6/18.1	32.0

Table 2.2: The rate of successful convergence for various poses

Algorithm	Front	Up	Down	Left	Right
2D AAM [37]	72%	62%	57%	61%	51%
2D+3D AAM [35, 36]/real_shape	93%	81%	67%	79%	70%
2D+3D AAM/virtual_shape	95%	79%	70%	84%	61%



(a) 2D AAM

(b) 2D+3D AAM/real_shape

(c) 2D+3D AAM/virtual_shape

Figure 2.7: Examples of image fitting by the 2D AAM [37], 2D+3D AAM [35, 36]/real_shape, and 2D+3D AAM/virtual_shape.

2.5 Chapter Summary and Discussion

In this chapter, we have proposed EM-GPA, which is a way of performing GPA when some variables are hidden. EM-GPA combines GPA and the EM algorithm to estimate scales, rotations, and a mean shape and covariance matrix of 3D shapes from multiple 2D shapes. EM-GPA can align rotated 2D shapes successfully by taking the missing information into consideration. The virtual 3D shape model created by EM-GPA can be successfully applied to AAM.

Since the E-step of EM-GPA is to calculate the posterior distribution of hidden variable $\mathbf{h}_i$ which normally corresponds to depth information for a given 2D shape. Hence, if we want to directly estimate the depth information of a specific 2D shape, rather than some parameters for constructing a 3D shape model, the posterior mean in E-step can be used for the estimated depth. It is equivalent to reconstructing 3D shapes form a set of 2D shapes, which are well known as non-rigid structure from motion (NRSfM) and we will focus on NRSfM in Chapter 3.

Chapter 3

Procrustean Normal Distribution Mixture Model

In this chapter, we are interested in reconstructing 3D shapes of a non-rigid object under complex shape variations. To address this problem, we propose a Procrustean normal distribution mixture model (PNDMM) under the assumption that complex shape variations can be decomposed into a collection of simpler and primitive shape variations. As can be seen in Figure 3.1, the PNDMM probabilistically models the generative process of 2D shapes from a mixture of 3D shapes and allows efficient 3D reconstruction.

In addition, we directly estimate the number of non-rigid shape mixture components using an *adaptive* PNDMM, which is based on the *maximum a posteriori* (MAP) principle with a prior on the number of mixture components derived from the minimum message length principle [42]. In order to make the proposed mixture model robust with respect to initialization, the component-wise expectation-maximization algorithm (CEM) [43] is applied.

This chapter is based on the paper appeared in International Journal of Computer Vision: 'Complex Non-Rigid 3D Shape Recovery Using a Procrustean Normal Distribution Mixture Model [41]'.

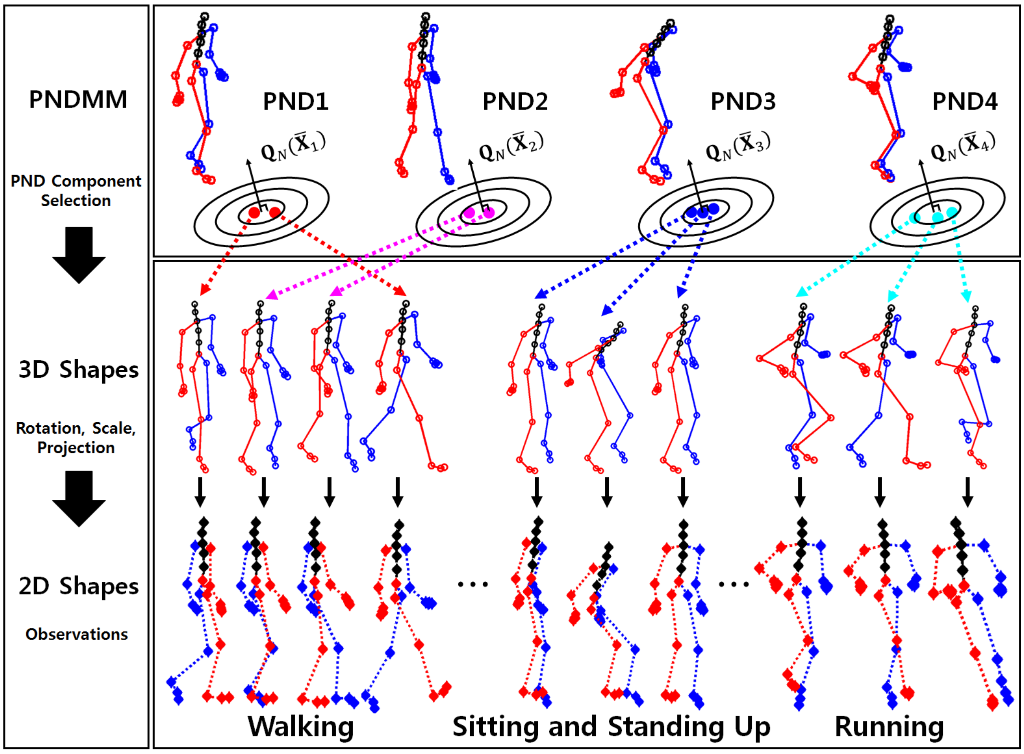


Figure 3.1: A graphical illustration of a PNDMM. 2D shapes can be considered as projections of scaled and rotated 3D shapes, in which 3D shapes are generated by corresponding Procrustean normal distribution (PND) components.

We have tested the proposed approach extensively on highly complex and long human motion sequences obtained from the CMU Mocap database[1], UMPM dataset [44], popular benchmark datasets [45] which consist of simple and short motion sequences, and the Penn Action dataset [46]. Experimental results show that the proposed method significantly outperforms existing methods. We also show that complex shape variations can be well modeled by a PNDMM and each component of the learned mixture model describes primitive non-rigid shape variations.

Relation with other chapters In this chapter, we assume that a set of shapes are given before estimating 3D shapes. However, if we can apply the PNDMM to a 2D shape obtained from an image, it is more useful in practical situations. For that reason, we will utilize the PNDMM to estimate a 3D human pose based on an image in Chapter 4.

3.1 Non-Rigid Structure from Motion

Recovering 3D shapes from a single image or multiple images is one of the fundamental problems in computer vision. Shape from stereo is to acquire information about the 3D structure and distances to objects from two or more images taken from different viewpoints [47]. Structure from motion (SfM) finds the three dimensional structure of an object by analyzing the image streams with the assumption that the object is rigid [30].

There have been efforts to extend the SfM approach to recover the shape of a non-rigid object. If an object deforms arbitrarily, it is impossible to reconstruct the shape from a set of 2D images. However, many non-rigid objects around us deform under a constrained space. Bregler *et al.*[31] have extended SfM to a deformable object by assuming that a 3D shape of a non-rigid object lie in a shape space, hence, a 3D shape can be described

[1] `http://mocap.cs.cmu.edu/subjects.php`

by a linear combination of a set of shape basis vectors. However, the bilinear formulation for non-rigid 3D shape recovery makes a solution ambiguous, which remains as a difficult problem in NRSfM. Hence, many have proposed to solve the NRSfM problem by introducing additional constraints [7, 8, 9, 45, 48, 10, 19, 18].

Xiao *et al.*[7] have shown that using only a rotation constraint is not sufficient for obtaining an unambiguous solution. Since the ambiguity of the solution comes from the shape basis is not unique, they proposed additional constraints, termed basis constraints, to derive a closed-form solution. Torresani *et al.*[8] proposed an approach, called EM-PPCA, using probabilistic principal components analysis with a Gaussian prior on each shape in the subspace. Paladini *et al.*[9] proposed a least-squares approach, called metric projections (MP), associated with a globally optimal projection step onto the manifold of metric constraints to recover 3D shapes and motion of deformable and articulated objects. A large number of existing approaches have focused on restricting the degree of deformation by fixing the number of shape basis vectors. However, it is difficult to know the optimal number of shape basis vectors and the choice on the number can greatly affect the reconstruction performance [31, 8, 9, 18].

A set of new approaches has been introduced to overcome the limitation of shape basis approaches. Akhter *et al.*[45] proposed a dual approach, in which 3D point trajectories are modeled compactly in the domain of the discrete cosine transform (DCT) basis, instead of estimating a set of shape basis vectors. Hence, there is a significant reduction in unknowns and it makes the estimation more stable. Gotardo *et al.*[48] modeled 3D shape deformation as a single point smoothly moving over time within a linear space spanned by 3D shape basis vectors and applied a DCT based approach to the smooth 3D shape trajectory. However, the number of DCT basis vectors must be known in advance, which is another difficulty. Although Zhu *et al.*[49], recently, proposed a method using convolutional sparse coding for NRSfM based on point trajectories, it requires learning an over-complete basis of 3D trajectories, prior to performing 3D reconstruction.

Meanwhile, Akhter *et al.*[50] have shown that the ambiguity of a solution, claimed by [7], is caused by overlooking of the rank three constraint on rotation matrices. By imposing the rank three constraint to the general solution given by [7], they have shown that the ambiguity in orthonormality constraints does not translate to an ambiguity in structure reconstruction. That is, the orthonormality constraints are sufficient for perfect structure reconstruction and the real problem in NRSM is the complexity of the underlying non-linear optimization. Based on the proof of the uniqueness of the solution, Dai *et al.*[10] proposed an algorithm called a simple prior-free method. Under the assumption the measurement is already truncated to a specific rank, *i.e.*, the number of shape basis vectors K has been estimated, they have shown outstanding performance against existing non-rigid factorization methods without any prior knowledge on basis vectors. However, Dai *et al.*[10] have assumed that the optimal number of shape basis vectors is known in advance. But it is difficult to estimate the correct number of shape basis vectors in practice. Once an incorrect number of shape basis vectors is used, rotations will be incorrectly estimated and, consequentially, an NRSfM algorithm fails to find a good solution. Zhu *et al.*[51] have shown that complex shape variations involving a sequence of primitive actions is hard to model in a low-dimensional linear space and represented a complex motion as a union of linear subspaces using low-rank representation [52]). In [51]), they assumed existing schemes on NRSfM can sufficiently align 2D projections in a 3D space but did not provide a particular method for estimating rotation matrices. However, obtaining right rotation matrices is the main difficulty in NRSfM.

Recently, Cho *et al.*[19] have proposed EM-GPA for finding a solution to NRSfM without any rank constraints using generalized Procrustes analysis (GPA) and expectation-maximization (EM). To make GPA more tractable for NRSfM, Lee *et al.*[18] have proposed a new probability distribution, called the Procrustean normal distribution (PND), which captures the distribution of non-rigid variations of an object by excluding the effects of rigid motion. Moreover, Lee *et al.*[18] have shown that NRSfM can be efficiently solved by learning a PND from 2D point tracks using the EM algorithm. The PND has been extended to a first-order stationary Markov process, which is called as Procrustean Markov process [53]), and it has achieved the state-of-the-art performance on a number of popular benchmark datasets.

Although many successful approaches have been introduced, there is still a limitation. The low-rank assumption is too restrictive to handle real world shape variations of complex non-rigid deformations. Moreover, while the PND proposed by [18] has shown outstanding results on non-rigid shape variations for short sequences, it is still difficult to capture complex shape variations using a PND. For these reasons, existing methods are not suitable for estimating the 3D shape of a non-rigid object undergoing complex shape variations.

There are methods designed to handle a complex NRSfM problem [54, 55, 56, 57]. However, these methods are restricted to reconstruct 3D shapes of a surface-like-object, or using prior information about the object. Unlike the previous work which focus on local rigidity in the spatial direction, we decompose variations of a non-rigid object into primitive shape variations and focus on reconstructing 3D shapes with given data for articulated objects like a human body.

3.2 Procrustean Normal Distribution (PND)

Finding the correct set of rotations is the most important issue in NRSfM. Cho *et al.*[19] and Lee *et al.*[18] have proposed a novel method for modeling rotations by incorporating GPA, which finds relative motions between similar shapes by aligning them under the common reference using rigid transformation, *i.e.*, scale, rotation, and translation. This principle determines rigid motions by minimizing non-rigid variations, which can improve the accuracy of NRSfM.

Let $\mathbf{X}_i \in \mathbb{R}^{3\times n_p}$, $s_i \in \mathbb{R}$, $\mathbf{R}_i \in \mathbb{R}^{3\times 3}$, and $\mathbf{t}_i \in \mathbb{R}^3$ be the 3D shape, scale, rotation, and translation, respectively, for the ith sample (or frame), $1 \leq i \leq n_s$, where n_p and n_s are the number of landmarks in a frame and the number of frames, respectively. Then, the GPA problem can be written as

$$\begin{aligned} &\min_{s_i,\mathbf{R}_i,\mathbf{t}_i,\overline{\mathbf{X}}} \sum \|s_i\mathbf{R}_i\mathbf{X}_i + \mathbf{t}_i\mathbf{1}^T - \overline{\mathbf{X}}\|_F^2 \\ &\text{subject to} \qquad \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}, \quad \|s_i\mathbf{X}_i\|_F = 1, \end{aligned} \tag{3.1}$$

where $\mathbf{1}$ is a vector of ones and $\overline{\mathbf{X}}$ is the mean shape. $\|\cdot\|_F$ denotes the Frobenius norm, *i.e.*, $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T\mathbf{A}) = \|\mathbf{vec}(\mathbf{A})\|_2^2$ with a vectorization operator $\mathbf{vec}(\cdot)$. Here, the scale constraint makes the aligned shapes lie on a Procrustes shape space [58]. However, the Procrustes shape space is a nonlinear manifold, which makes it difficult to solve the NRSfM problem using constraints in (3.1). Although Pizarro *et al.*[59] proposed a new GPA method by a global optimization and it can handle missing information, it is still hard to be extended to the NRSfM problem.

Lee *et al.*[18] addressed this nonlinearity issue by introducing a new scale constraint so that each shape variation from the mean shape is orthogonal to the mean shape, *i.e.*, $\mathbf{vec}(s_i\mathbf{R}_i\mathbf{X}_i - \overline{\mathbf{X}})^T\mathbf{vec}(\overline{\mathbf{X}}) = 0$. Here, if we impose an additional constraint that the norm of the mean shape is one, *i.e.*, $\|\overline{\mathbf{X}}\|_F = 1$, then the constraint can be rewritten as $s_i\text{tr}(\mathbf{R}_i\mathbf{X}_i\overline{\mathbf{X}}^T) = 1$. Note that this is a linear constraint with respect to s_i and it scales the aligned shape such that its projection onto the mean shape is one, *i.e.*, $s_i = \frac{1}{\mathbf{vec}(\mathbf{R}_i\mathbf{X}_i)^T\mathbf{vec}(\overline{\mathbf{X}})}$.

Based on this new constraint, Lee *et al.*[18] made another important observation. The necessary condition for the optimality of the GPA problem (3.1) can be obtained as

$$\begin{aligned} \mathbf{R}_i^T\mathbf{R}_i = \mathbf{I}, &\qquad \|\overline{\mathbf{X}}\|_F^2 = 1, \\ s_i\text{tr}(\mathbf{R}_i\mathbf{X}_i\overline{\mathbf{X}}^T) = 1, &\qquad \mathbf{R}_i\mathbf{X}_i\overline{\mathbf{X}}^T \in \mathbf{S}_+^3, \end{aligned} \tag{3.2}$$

where $\mathbf{S}_+^3$ is a set of three dimensional positive semi-definite matrices (PSDs), which is convex. The last two constraints in (3.2) can be considered as the convex constraints of aligned shapes [18]. Since a shape alignment method can be used to extract only non-rigid variations from a set of shapes, these constraints can concisely describe the convex set of non-rigid shape variations.

Lee *et al.*[18] also proposed a new probability distribution, called the Procrustean normal distribution (PND). It defines the distribution of non-rigid variations of shapes, by separating rigid variations from non-rigid variations of shapes based on the above constraints, as follows:

$$p(\mathbf{Y}) \propto \frac{1}{|\mathbf{\Sigma}_R|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{v}^T\mathbf{Q}\mathbf{\Sigma}_R^{-1}\mathbf{Q}^T\mathbf{v}\right)\delta(\mathbf{Q}_N^T\mathbf{v}). \tag{3.3}$$

where $\mathbf{v} = \mathbf{vec}(\mathbf{Y} - \overline{\mathbf{Y}})$, $\overline{\mathbf{Y}} = \overline{\mathbf{X}}$, and $\mathbf{Y}$ is an aligned shape expressed as $\mathbf{Y} = s\mathbf{R}\mathbf{X}$ using s and $\mathbf{R}$ satisfying (3.2). Let $\mathbf{\Sigma} \in \mathbb{R}^{n_Y \times n_Y}$ be the covariance matrix of $\mathbf{vec}(\mathbf{Y}) \in \mathbb{R}^{n_Y}$, where $n_Y = 3n_p$. Then the reduced non-singular covariance matrix is represented by $\mathbf{\Sigma}_R = \mathbf{Q}^T\mathbf{\Sigma}\mathbf{Q} \in \mathbb{R}^{n_R \times n_R}$, which includes only non-rigid shape variations, where $\mathbf{Q} \in \mathbb{R}^{n_Y \times n_R}$ is a column orthogonal matrix to remove rigid shape variations[2] with $n_R = n_Y - 7$. Here, 7 is the degree of freedom of rigid shape variations which is explained in (3.4). The Dirac-delta term in (3.3) is introduced so that $p(\mathbf{Y})$ can be expressed with the degenerate $\mathbf{\Sigma}$ and makes the PND has zero probability whenever $\mathbf{Y}$ has a component in the subspace of rigid variations. $\mathbf{Q}$ and $\mathbf{Q}_N$ satisfy $\mathbf{Q}^T\mathbf{Q}_N = \mathbf{0}$[3]. $\mathbf{Q}_N$ is an orthogonal matrix for rigid shape variations, which is derived from (3.2), and can be expressed as [18]:

$$\mathbf{Q}_N = \begin{bmatrix}\mathbf{vec}(\overline{\mathbf{Y}}) & \mathbf{Q}_L & (\mathbf{1} \otimes \mathbf{I})/\sqrt{n_p}\end{bmatrix} \in \mathbb{R}^{n_Y \times 7}, \tag{3.4}$$

where 7 is the number of basis vectors for rigid variations in a three dimensional space, *i.e.*, the number of basis vectors for $\mathbf{vec}(\overline{\mathbf{Y}})$ (scale), $\mathbf{Q}_L$ (rotation), and $(\mathbf{1} \times \mathbf{I})/\sqrt{n_p}$

[2] $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ but $\mathbf{Q}\mathbf{Q}^T \neq \mathbf{I}$.

[3] In this chapter, we use $\mathbf{0}$ to denote both matrices and vectors of zeros.

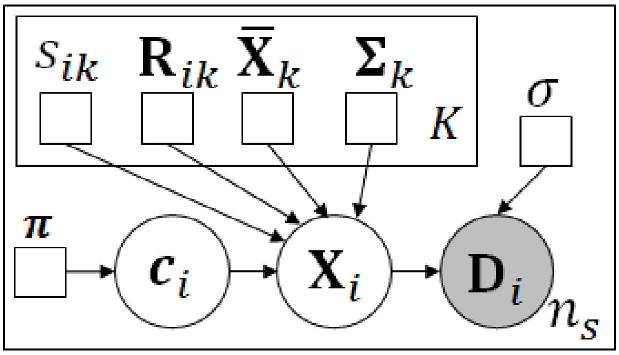


Figure 3.2: A graphical representation of the PNDMM.

(translation) are one, three, and three, respectively. $\mathbf{Q}_L$ is an orthogonalized version of $\mathbf{L}(\overline{\mathbf{Y}}) = \left[[\overline{\mathbf{y}}_1]_\times, \cdots, [\overline{\mathbf{y}}_{n_p}]_\times\right]^T$, where $\overline{\mathbf{y}}_i$ is the ith column vector of $\overline{\mathbf{Y}}$ and $\otimes$ is the Kronecker product and $[\overline{\mathbf{y}}]_\times \in \mathbb{R}^{3\times3}$ is a skew-symmetric matrix [18].

Since the PND does not include any rigid shape variations, it is possible to find relative non-rigid variations between sample shapes by learning a PND. This has the same effect as solving a GPA problem. Moreover, we can apply existing statistical estimation techniques to the PND. A PND random matrix $\mathbf{Y}$ is denoted as $\mathbf{Y} \sim \mathcal{N}_P(\overline{\mathbf{Y}}, \mathbf{\Sigma})$. If a distribution satisfies the properties of the PND but its mean is not unit-norm, then it is called a scaled PND and denoted by $\mathcal{N}_P^s$.

3.3 PND Mixture Model

A PND mixture model (PNDMM) is represented as a generative probabilistic model of an observed 2D shape from a non-rigid object as shown in Figure 3.2. The main idea is that we observe 2D shapes with missing depth information about 3D shapes, in which the 3D shapes are represented by a mixture of PNDs and each PND is characterized by (3.3).

Let $\mathbf{D}_i \in \mathbb{R}^{3\times n_p}$ be a matrix representing input landmarks of the ith sample, observed by an orthographic camera. The first two rows of $\mathbf{D}_i$ are filled with observed 2D landmark positions and the third row is filled with zeros, since the depth is unknown. Let $\mathbf{X}_i$ be the 3D shape of the ith sample. Also, let $\mathbf{c}_i$ be a K-dimensional binary random variable having 1-of-K representation in which a particular element of $\mathbf{c}_i$ is equal to 1 and all other elements are equal to 0, *i.e.*, $c_{ik} \in \{0, 1\}$ and $\sum_k c_{ik} = 1$. Its role is to indicate which component has generated the ith 3D shape $\mathbf{X}_i$, thus, K is the number of components. The joint distribution of a PNDMM can be expressed as:

$$\begin{aligned} p(\mathbf{D}, \mathbf{X}, \mathbf{c}|\pi) &= \prod_i \prod_k p(\mathbf{D}, \mathbf{X}, c_{ik} = 1|\pi)^{c_{ik}} \\ &= \prod_i \prod_k \{p(\mathbf{D}_i|\mathbf{X}_i) p(\mathbf{X}_i|c_{ik} = 1) p(c_{ik} = 1|\pi)\}^{c_{ik}}. \end{aligned} \tag{3.5}$$

The generative process for each 2D shape $\mathbf{D}_i$ can be represented as follows.

1. The distribution of $\mathbf{c}_i$ is specified in terms of the mixing coefficients, such that

$$p(c_{ik} = 1|\pi) = \pi_k, \qquad k = 1, \ldots, K, \tag{3.6}$$

   where $\pi = \{\pi_1, \ldots, \pi_K\}$ and $\pi_k \geq 0$ is a mixing probability and $\sum_{k=1}^{K} \pi_k = 1$. Therefore, $\mathbf{c}_i$ is chosen from the following distribution:

$$p(\mathbf{c}_i|\pi) = \prod_k {\pi_k}^{c_{ik}}. \tag{3.7}$$

2. Since we assume that the 3D shape has a PND, the aligned shape $\mathbf{Y}_{ik} = s_{ik}\mathbf{R}_{ik}\mathbf{X}_i$ has the corresponding PND, where s_{ik} is a scale and $\mathbf{R}_{ik}$ is a rotation matrix for the ith sample and the kth PND component obtained from the modified GPA constraints in (3.2), *i.e.*, $\mathbf{Y}_{ik} \sim \mathcal{N}_P(\overline{\mathbf{X}}_k, \mathbf{\Sigma}_k)$. Therefore, according to Proposition 2 in [18], $\mathbf{X}_i|c_{ik} = 1$ is chosen from a scaled PND as

$$\mathbf{X}_i|c_{ik} = 1 \quad \sim \quad \mathcal{N}_P^s \left(s_{ik}^{-1}\mathbf{R}_{ik}^T\overline{\mathbf{X}}_k, s_{ik}^{-2}\mathbf{\Sigma}'_k\right), \tag{3.8}$$

   where $\mathbf{\Sigma}'_k = (\mathbf{I} \otimes \mathbf{R}_{ik}^T)\mathbf{\Sigma}_k(\mathbf{I} \otimes \mathbf{R}_{ik})$.

3. Let $\mathbf{B}_i \in \mathbb{R}^{3\times n_p}$ be a mask matrix which indicates whether the corresponding elements are observed (one) or missing (zero). In case of NRSfM, the last row of $\mathbf{B}_i$ is filled with zeros because the z coordinates are unknown. If there are additional missing observations, the corresponding elements in $\mathbf{B}_i$ are also filled with zeros. We define a projection matrix $\mathbf{F}_i$ to handle missing observations as follows [18]: $\mathbf{F}_i = \widehat{\mathbf{B}}_i - \widehat{\mathbf{B}}_i(\mathbf{1}\otimes\mathbf{I})\mathrm{diag}(\mathbf{a}_i)(\mathbf{1}^T\otimes\mathbf{I})\widehat{\mathbf{B}}_i$ and $\mathbf{F}_i^2 = \mathbf{F}_i$, where $\mathbf{a}_i$ is a three dimensional vector whose jth element is $1/\sum_l b_{ijl}$, b_{ijl} is the (j,l)th element of $\mathbf{B}_i$, $\widehat{\mathbf{B}}_i = \mathrm{diag}(\mathbf{vec}(\mathbf{B}_i))$, and $\mathrm{diag}(\cdot)$ denotes a diagonal matrix with elements of a vector on the main diagonal.

   Let the input landmark $\mathbf{D}_i$ be initialized, such that $\mathbf{D}_i\mathbf{1} = \mathbf{0}$, as follows:

$$d_{ijl} \leftarrow \begin{cases} d_{ijl} - \frac{\sum_m b_{ijm} d_{ijm}}{\sum_m b_{ijm}} & \text{if } b_{ijl} = 1 \\ 0 & \text{otherwise} \end{cases}, \tag{3.9}$$

   where d_{ijl} is the (j,l)th element of $\mathbf{D}_i$. Then the 2D observation $\mathbf{D}_i$ is obtained from the 3D shape $\mathbf{X}_i$ with a Gaussian noise as

$$\mathbf{vec}(\mathbf{D}_i) = \mathbf{F}_i\mathbf{vec}(\mathbf{X}_i) + \mathbf{u}_i, \tag{3.10}$$

   where $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

In the following sections, we learn a PNDMM from input data using the EM algorithm and component-wise EM algorithm, resulting two algorithms: PNDMM and *adaptive* PNDMM.

3.4 Learning a PNDMM

The goal of a general EM algorithm is to maximize the log-likelihood function $p(\mathbf{D}|\mathbf{\Phi})$, given a joint distribution $p(\mathbf{D}, \mathbf{X}, \mathbf{c}|\mathbf{\Phi})$ over observed variables $\mathbf{D}$, hidden variables

$\mathbf{X}$, membership variables $\mathbf{c}$, and parameters $\mathbf{\Phi}$. Since $\mathbf{X}$ is a continuous random variable and $\mathbf{c}$ is a discrete random variable, if we define the parameter set as $\mathbf{\Phi}_{ik} = \{\sigma, s_{ik}, \mathbf{R}_{ik}, \overline{\mathbf{X}}_k, \mathbf{\Sigma}_k, \pi_k\}$, the cost function for the EM algorithm can be represented as follows:

$$J(\mathbf{\Phi}|\mathbf{\Phi}^{old}) = \sum_k \sum_i w_{ik} \int \ln(p(\mathbf{D}_i, \mathbf{X}_i, c_{ik} = 1|\mathbf{\Phi}_{ik})) \\ \times p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}^{old}_{ik}) d\mathbf{X}_i. \tag{3.11}$$

Here, we use the chain rule as $p(\mathbf{X}_i, c_{ik} = 1|\mathbf{D}_i, \mathbf{\Phi}^{old}_{ik}) = p(c_{ik} = 1|\mathbf{D}_i, \mathbf{\Phi}^{old}_{ik}) p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}^{old}_{ik})$ and denote $p(c_{ik} = 1|\mathbf{D}_i, \mathbf{\Phi}^{old}_{ik})$ by w_{ik}, since $p(c_{ik} = 1|\mathbf{D}_i, \mathbf{\Phi}^{old}_{ik})$ plays the role as a weight for the component indicated by $\mathbf{c}_i$. The superscript old denotes the parameter set obtained from the previous M-step in the EM iteration procedure.

3.4.1 E-step

In the E-step, we estimate w_{ik} and the distribution of $\mathbf{X}_i$ given the current estimates of parameter $\mathbf{\Phi}^{old}_{ik}$ and observation $\mathbf{D}_i$. From now on, we will omit the superscript (old) if no confusion arises.

Calculation of w_{ik}

Using Bayes' rule, w_{ik}, the posterior distribution of c_{ik}, can be written as

$$w_{ik} = \frac{\pi_k p(\mathbf{D}_i|c_{ik} = 1, \mathbf{\Phi}_{ik})}{\sum_l \pi_l p(\mathbf{D}_i|c_{il} = 1, \mathbf{\Phi}_{il})}, \tag{3.12}$$

where π_k is a priori probability as shown in (3.6) and $p(\mathbf{D}_i|c_{ik} = 1, \mathbf{\Phi}_{ik})$ is a marginal distribution over $\mathbf{X}_i$, which can be written as

$$p(\mathbf{D}_i|c_{ik} = 1, \mathbf{\Phi}_{ik}) \\ = \int p(\mathbf{D}_i|\mathbf{X}_i, \sigma) p(\mathbf{X}_i|c_{ik} = 1, \mathbf{\Phi}_{ik}) d\mathbf{X}_i. \tag{3.13}$$

From (3.8), (3.10), and (3.13), the conditional distribution of $\mathbf{D}_i$ given $c_{ik} = 1$ can be calculated as

$$\mathbf{D}_i|c_{ik} = 1 \sim \mathcal{N}\left(\mathbf{F}_i\mathbf{vec}\left(\frac{1}{s_{ik}}\mathbf{R}_{ik}^T\overline{\mathbf{X}}_k\right), \widetilde{\mathbf{\Sigma}}_{ik}\right), \tag{3.14}$$

where $\widetilde{\mathbf{\Sigma}}_{ik} = \frac{1}{s_{ik}^2}\mathbf{F}_i(\mathbf{I}\otimes\mathbf{R}_{ik}^T)\mathbf{Q}_k\mathbf{\Sigma}_{R_k}\mathbf{Q}_k^T(\mathbf{I}\otimes\mathbf{R}_{ik})\mathbf{F}_i^T + \sigma^2\mathbf{I}$. Therefore, we can compute (3.12) using (3.14).

Calculation of $p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}_{ik})$

Using Bayes' rule, the posterior distribution of $\mathbf{X}_i$ is represented as

$$p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}_{ik}) \propto p(\mathbf{D}_i|\mathbf{X}_i, \sigma)p(\mathbf{X}_i|c_{ik} = 1, \mathbf{\Phi}_{ik}) \tag{3.15}$$

and after some algebra using facts, such as $\mathbf{vec}(\overline{\mathbf{X}}_k)^T\mathbf{Q}_k = 0$, $\mathbf{vec}(\mathbf{D}_i) = \mathbf{F}_i\mathbf{vec}(\mathbf{D}_i)$, and $\mathbf{F}_i^2 = \mathbf{F}_i$, we obtain

$$p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}_{ik}) \propto \exp\left(-\frac{1}{2}\zeta + \frac{1}{\sigma^2}\xi\right)\delta', \tag{3.16}$$

where

$$\begin{aligned}
\zeta &= \mathbf{vec}(\mathbf{X}_i)^T\mathbf{H}_{ik}\mathbf{vec}(\mathbf{X}_i)\\
\xi &= \mathbf{vec}(\mathbf{D}_i)^T\mathbf{vec}(\mathbf{X}_i)\\
\mathbf{H}_{ik} &= s_{ik}^2(\mathbf{I}\otimes\mathbf{R}_{ik}^T)\mathbf{\Sigma}_k^+(\mathbf{I}\otimes\mathbf{R}_{ik}) + \frac{1}{\sigma^2}\mathbf{F}_i\\
\delta' &= \delta\Big(\mathbf{Q}_{N_k}^T\mathbf{vec}(\mathbf{R}_{ik}\mathbf{X}_i) - \left[1/s_{ik}\ \mathbf{0}^T\right]^T\Big),
\end{aligned}$$

in which $\mathbf{\Sigma}_k^+$ is the psedudo-inverse of $\mathbf{\Sigma}_k$, *i.e.*, $\mathbf{\Sigma}_k^+ = \mathbf{Q}_k\mathbf{\Sigma}_{R_k}^{-1}\mathbf{Q}_k^T$, and δ' comes from the Dirac-delta term in (3.3). However, since the Dirac-delta term is too restrictive to allow a meaningful update in the EM procedure in practice, we ignore the Dirac-delta term as done in [18]. Since (3.15) is the posterior distribution for an individual PND when the kth PND is selected for the ith sample, $p(\mathbf{X}_i|c_{ik} = 1, \mathbf{D}_i, \mathbf{\Phi}_{ik})$ can be represented by the Gaussian distribution as follows (see Chapter B):

$$\begin{aligned} & p(\mathbf{X}_i|c_{ik}=1,\mathbf{D}_i,\mathbf{\Phi}_{ik}) \\ & = p(\mathbf{vec}(\mathbf{X}_i)|c_{ik}=1,\mathbf{D}_i,\mathbf{\Phi}_{ik}) \sim \mathcal{N}\left(\mathbf{m}_{ik},\mathbf{\Omega}_{ik}\right), \\ & \text{where} \\ & \mathbf{m}_{ik} = \frac{1}{\sigma^2}\mathbf{\Omega}_{ik}\mathbf{vec}(\mathbf{D}_i) \text{ and } \mathbf{\Omega}_{ik} = \mathbf{H}_{ik}^{+}. \end{aligned} \tag{3.17}$$

3.4.2 M-step

In the M-step, the maximum likelihood solution of parameter $\mathbf{\Phi}_{ik}$ is obtained using w_{ik} and the posterior distribution of $\mathbf{X}_i$ computed from the E-step. The objective function (3.11) can be rewritten as

$$\begin{aligned} & J(\mathbf{\Phi}|\mathbf{\Phi}^{old}) = -\sum_k\sum_i w_{ik}\Bigg(n_i^{\mathbf{B}}\ln\sigma \\ & + \frac{1}{2\sigma^2}\|\mathbf{vec}(\mathbf{D}_i) - \mathbf{F}_i\mathbf{m}_{ik}\|_2^2 + \frac{1}{2\sigma^2}\mathrm{tr}(\mathbf{F}_i\mathbf{\Omega}_{ik}) - n_R\ln s_{ik} \\ & + \frac{1}{2}\ln|\mathbf{\Sigma}_{R_k}| + \frac{1}{2}\mathbf{h}_{ik}^T\mathbf{\Sigma}_{R_k}^{-1}\mathbf{h}_{ik} \\ & + \frac{s_{ik}^2}{2}\mathrm{tr}\left((\mathbf{I}\otimes\mathbf{R}_{ik}^T)\mathbf{Q}_k\mathbf{\Sigma}_{R_k}^{-1}\mathbf{Q}_k^T(\mathbf{I}\otimes\mathbf{R}_{ik})\mathbf{\Omega}_{ik}\right) - \ln\pi_k\Bigg), \end{aligned} \tag{3.18}$$

where $n_i^{\mathbf{B}} = \sum_j\left(\sum_l b_{ijl}\right) - \mathrm{sign}\left(\sum_l a_{il}\right)$ and $\mathbf{h}_{ik} = \mathbf{Q}_k^T(s_{ik}(\mathbf{I}\otimes\mathbf{R}_{ik})\mathbf{m}_{ik} - \mathbf{vec}(\overline{\mathbf{X}}_k))$. Then, the optimization problem for the M-step can be formulated as

$$\begin{aligned} & \max_{\mathbf{\Phi}} J(\mathbf{\Phi}|\mathbf{\Phi}^{old}) \\ & \text{subject to} \quad \sum_k\pi_k = 1, \quad \mathbf{R}_{ik}^T\mathbf{R}_{ik} = \mathbf{I}, \quad \left\|\overline{\mathbf{X}}_k\right\|_F^2 = 1, \\ & s_{ik}\mathrm{tr}(\mathbf{R}_{ik}\mathbf{M}_{ik}\overline{\mathbf{X}}_k^T) = 1, \quad \mathbf{R}_{ik}\mathbf{M}_{ik}\overline{\mathbf{X}}_k^T \in \mathbf{S}_+^3, \end{aligned} \tag{3.19}$$

where $\mathbf{M}_{ik}$ is the expectation of $\mathbf{X}_i$ with respect to its posterior distribution, *i.e.*, $\mathbf{vec}(\mathbf{M}_{ik}) = \mathbf{m}_{ik}$. The last four constraints in (3.19) are the same as the constraints in (3.2) except that $\mathbf{X}_i$ is replaced with its expectation $\mathbf{M}_{ik}$. Since this problem is highly complicated, we alternatively update each parameter with the other parameters fixed. From now on, we explain how to update each parameter.

When updating $\overline{\mathbf{X}}_k$, there is a difficulty coming from the dependency of $\overline{\mathbf{X}}_k$ on $\mathbf{Q}_k$ as well as the constraints in (3.19). To solve it, we regard $\mathbf{Q}_k$ as an independent parameter and ignore the constraints in the update of $\overline{\mathbf{X}}_k$. Then, by differentiating the cost function with respect to $\overline{\mathbf{X}}_k$ and equating it to zero, and normalizing the solution, we obtain the following update equation:

$$\overline{\mathbf{X}}_k = \sum_i w_{ik} s_{ik} \mathbf{R}_{ik} \mathbf{M}_{ik} \Big/ \Big\| \sum_i w_{ik} s_{ik} \mathbf{R}_{ik} \mathbf{M}_{ik} \Big\|_F. \tag{3.20}$$

The scale and rotation are relatively easy to update, since the feasible s_{ik} and $\mathbf{R}_{ik}$ are unique according to the constraints in (3.19), if the samples are non-degenerate and the other parameters are fixed. The corresponding update equations are

$$\begin{aligned} &\mathbf{M}_{ik} \overline{\mathbf{X}}_k^T = \mathbf{U}_{ik} \mathbf{\Lambda}_{ik} \mathbf{V}_{ik}^T, \qquad \mathbf{R}_{ik} = \mathbf{V}_{ik} \mathbf{U}_{ik}^T, \\ &s_{ik} = 1/\mathrm{tr}(\mathbf{R}_{ik} \mathbf{M}_{ik} \overline{\mathbf{X}}_k^T) = 1/\mathrm{tr}(\mathbf{\Lambda}_{ik}), \end{aligned} \tag{3.21}$$

where $\mathbf{U}_{ik} \mathbf{\Lambda}_{ik} \mathbf{V}_{ik}^T$ is the singular value decomposition of $\mathbf{M}_{ik} \overline{\mathbf{X}}_k^T$.

$\mathbf{Q}_{N_k}$ and $\mathbf{Q}_k$ can be updated by (3.4) using the new $\overline{\mathbf{X}}_k$, as described in Section 3.2. $\mathbf{\Sigma}_{R_k}$ can be obtained by solving the first-order necessary condition of (3.19), *i.e.*,

$$\mathbf{\Sigma}_{R_k} = \sum_i w_{ik} \widetilde{\mathbf{\Omega}}_{ik} \Big/ \sum_i w_{ik}, \tag{3.22}$$

where $\widetilde{\mathbf{\Omega}}_{ik} = \mathbf{h}_{ik} \mathbf{h}_{ik}^T + s_{ik}^2 \mathbf{Q}_k^T (\mathbf{I} \otimes \mathbf{R}_{ik}) \mathbf{\Omega}_{ik} (\mathbf{I} \otimes \mathbf{R}_{ik}^T) \mathbf{Q}_k$. Notice that, due to $\mathbf{Q}_k$ in (3.22), $\mathbf{\Sigma}_{R_k}$ does not include rigid variations.

Accordingly, the covariance matrix is calculated as $\mathbf{\Sigma}_k = \mathbf{Q}_k \mathbf{\Sigma}_{R_k} \mathbf{Q}_k^T$. σ^2 can be derived in a similar way:

$$\sigma^2 = \frac{\sum_k \sum_i w_{ik} (\|\mathbf{vec}(\mathbf{D}_i) - \mathbf{F}_i \mathbf{m}_{ik}\|_2^2 + \mathrm{tr}(\mathbf{F}_i \mathbf{\Omega}_{ik}))}{\sum_i \sum_k n_i^{\mathbf{B}} w_{ik}}. \tag{3.23}$$

Finally, we maximize the cost function in (3.19) with respect to π_k. Since the sum of mixing probabilities has to be one, this can be achieved using the following problem:

$$\max_{\pi_k} \sum_k \sum_i w_{ik} \ln \pi_k, \quad \text{subject to} \quad \sum_k \pi_k = 1, \tag{3.24}$$

and the solution can be easily obtained as

$$\pi_k = \sum_i w_{ik} \Big/ \sum_k \sum_i w_{ik}. \tag{3.25}$$

We have empirically found that a single iteration of alternating parameter updates in an M-step is enough for getting a good solution. The E-step and M-step constitute the parameter updating rule for $\mathbf{\Phi}$, which are executed iteratively until convergence. While the covergence property of the proposed EM algorithm cannot be formalized easily, we have empirically found that each step of the EM algorithm improves the objective function (3.18) as shown in Figure 3.3. The complexity of the PNDMM per EM iteration is $\mathcal{O}((n_p)^3 K n_s)$, where n_p, K, and n_s are the number of landmark points, components, and samples, respectively, because the dominant operation of the proposed algorithm is the inverse of a matrix.

3.5 Learning an *Adaptive* PNDMM

In Section 3.4, we have solved NRSfM by learning a PND mixture model from observed data $\mathbf{D}_i$ using the standard EM algorithm. Since we have implicitly assumed that the correct number of mixture components is known in advance, it has limited applications. In this section, we address this limitation by extending the *adaptive* learning method proposed in [42] to the PNDMM, which can find the number of components. In addition, the method is less sensitive to initialization.

The PNDMM proposed in Section 3.4 estimates the model parameter using *maximum likelihood* (ML). However, the ML of $p(\mathbf{D}, \mathbf{X}, \mathbf{c}|\mathbf{\Phi})$ is a nondecreasing function of K

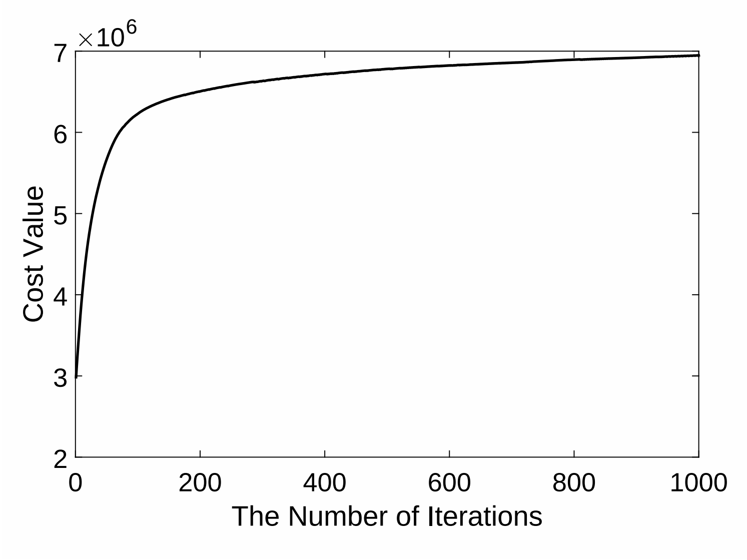


Figure 3.3: An example of the value of objective function of the PNDMM. An example showing that the value of objective function (3.18) increases with additional EM steps (CMU Mocap sequence CMU86_04).

[42], hence, it cannot be used to estimate the number of components. Therefore, we take the *maximum a posteriori* (MAP) approach as follows:

$$\mathbf{\Phi}_{MAP} = \arg\max_{\mathbf{\Phi}}\{\ln p(\mathbf{D}, \mathbf{X}, \mathbf{c}|\mathbf{\Phi}) + \ln p(\mathbf{\Phi})\}, \tag{3.26}$$

where $p(\mathbf{\Phi})$ is a Dirichlet-type prior proposed by Figueiredo and Jain [42], in which the prior for π is derived from the minimum message length (MML) principle as:

$$p(\pi_1, \ldots, \pi_K) \propto \exp\Big(-\frac{n_c}{2}\sum_k \ln \pi_k\Big). \tag{3.27}$$

Since this prior is only applied to π, it does not change the likelihood term. Hence, the E-step and M-step are the same as in Section 3.4, except for the update of π. The update for π in the M-step is now changed to:

$$\begin{aligned} &\max_{\pi_k} \sum_i \sum_k w_{ik} \ln \pi_k - \frac{n_c}{2}\sum_k \ln \pi_k, \\ &\text{subject to} \quad \sum_k \pi_k = 1. \end{aligned} \tag{3.28}$$

Unlike Section 3.4, there is an additional term, $\frac{n_c}{2}\sum \ln \pi_k$, due to (3.27). Owing to this additional term, the solution (3.25) is changed as

$$\pi_k = \max\Big(0, \sum_i w_{ik} - \frac{n_c}{2}\Big)\Big/\sum_k \max\Big(0, \sum_i w_{ik} - \frac{n_c}{2}\Big), \tag{3.29}$$

where w_{ik} are given by (3.12) in the E-step. Note that any component corresponding to $\pi_k = 0$ does not contribute to the log-likelihood, hence, the problem becomes

$$\mathbf{\Phi}_{ik} = \arg\max_{\mathbf{\Phi}_{ik}} J(\mathbf{\Phi}|\mathbf{\Phi}^{old}) \quad \text{for } \{k|\pi_k > 0\}, \tag{3.30}$$

subject to the same constraints in (3.19).

An important feature of the M-step defined in (3.29) is that it performs component annihilation: When starting with K which is larger than the optimal number of mixture components, a component which is fully supported by samples will survive, otherwise, it will be removed. Thereby, the proposed algorithm becomes more robust with respect to initialization [42]. However, there is an issue with the initial value of K. If K is too large, no component will have $\sum_i w_{ik} > \frac{n_c}{2}$ and π will be undetermined. We avoid this problem by using the component-wise EM algorithm [42], such that each component is updated sequentially, rather than simultaneously, *i.e.*, update π_1 and $\mathbf{\Phi}_1$, recompute all weights w_{ik}, update π_2 and $\mathbf{\Phi}_2$, recompute all weights w_{ik}, and so on. When one component dies ($\pi_k = 0$), an immediate redistribution of its probability mass to the other components can be made and it increases the chance of survival for other components. This allows initialization with an arbitrarily large K.

3.6 Experiments

3.6.1 Experimental setup

Since the proposed PNDMM is an extension of the PND, we first explain the parameter initialization method for the PND in Section 3.6.1 and extend it for PNDMM and *adaptive* PNDMM in Section 3.6.1.

PND initialization

To get initial rotation matrices $\mathbf{R}_i$, we use non-rigid factorization [31, 7, 9, 50, 48, 10] which decomposes a measurement matrix $\widetilde{\mathbf{D}} \in \mathbb{R}^{2n_s \times n_p}$ into a product of two matrices $\widetilde{\mathbf{R}} \in \mathbb{R}^{2n_s \times n_r}$ and $\widetilde{\mathbf{S}} \in \mathbb{R}^{n_r \times n_p}$ using singular value decomposition (SVD), where $\widetilde{\mathbf{D}} = \left[\widehat{\mathbf{D}}_1^T \dots \widehat{\mathbf{D}}_{n_s}^T\right]^T$ and $\widehat{\mathbf{D}}_i$ contains the first two rows of $\mathbf{D}_i$. The rank n_r for SVD is determined to keep 99.999 % of the total energy. Based on $\widetilde{\mathbf{R}}$, we get initial rotations with the orthonomality constraint as follows [7]:

$$\begin{aligned} &\min_{\mathbf{G}} \sum_i \|\mathbf{A}_i \mathbf{vec}(\mathbf{G}\mathbf{G}^T)\|_2^2, \\ &\text{where} \quad \mathbf{A}_i = \begin{bmatrix} \widetilde{\mathbf{R}}_{2i-1} \otimes \widetilde{\mathbf{R}}_{2i-1} - \widetilde{\mathbf{R}}_{2i} \otimes \widetilde{\mathbf{R}}_{2i} \\ \widetilde{\mathbf{R}}_{2i-1} \otimes \widetilde{\mathbf{R}}_{2i} + \widetilde{\mathbf{R}}_{2i} \otimes \widetilde{\mathbf{R}}_{2i-1} \end{bmatrix}, \end{aligned} \tag{3.31}$$

$\mathbf{G} \in \mathbb{R}^{n_r \times 3}$, and $\widetilde{\mathbf{R}}_i$ is the ith row of $\widetilde{\mathbf{R}}$. Instead of directly solving (3.31), we replace $\mathbf{G}\mathbf{G}^T$ with a positive semidefinite matrix $\mathbf{N} \in \mathbb{R}^{n_r \times n_r}$ and add a constraint $\mathrm{tr}(\mathbf{N}) = 1$ to avoid a trivial solution. After solving this convex semidefinite programming (SDP) problem, we solve (3.31) again with an explicit rank three constraint, because the true rank of $\mathbf{N} = \mathbf{G}\mathbf{G}^T$ is three [50, 10]. The initial rotation matrices $\mathbf{R}_i \in \mathbb{R}^{3\times 3}$ can be obtained from $\widetilde{\mathbf{R}}$ and $\mathbf{G}$.[4]

Given rotation matrices $\mathbf{R}_i$, initial shape matrices $\mathbf{S}_i$ are obtained so that the trace norm of the sample shape covariance matrix is minimized:

$$\begin{aligned} &\min_{\mathbf{z}_i} \mathrm{tr} \left(\frac{1}{n_s} \sum_i \mathbf{vec}(\mathbf{S}_i)\mathbf{vec}(\mathbf{S}_i)^T - \mathbf{vec}(\overline{\mathbf{S}})\mathbf{vec}(\overline{\mathbf{S}})^T \right), \\ &\text{subject to} \quad \mathbf{S}_i = \mathbf{R}_i \begin{bmatrix} \mathbf{D}_i \\ \mathbf{z}_i \end{bmatrix}, \quad \overline{\mathbf{S}} = \frac{1}{n_s} \sum_i \mathbf{S}_i. \end{aligned} \tag{3.32}$$

[4] Let $\mathbf{R}_i^f$ be a rotation matrix obtained from the factorization method. Then $\mathbf{R}_i$ in (3.32) is a transpose of $\mathbf{R}^f$.

The solution of the above problem can be found in a closed-form. After obtaining the initial 3D shapes $\widehat{\mathbf{X}}_i = \mathbf{R}_i^T \mathbf{S}_i$, we recalculate the scale s_i and rotation $\mathbf{R}_i$ base on (3.21) so that it is aligned in the GPA manner. The initial 3D mean shape $\overline{\mathbf{X}}$, null-space $\mathbf{Q}$, and the reduced covariance $\mathbf{\Sigma}_R$ are then calculated accordingly from the aligned initial shapes. When the observation matrix $\widetilde{\mathbf{D}}$ has missing elements, we perform a simple matrix completion to $\widetilde{\mathbf{D}}$ using the method from [60] and get initial parameters for a PND. The standard deviation σ of the observation noise is initialized as 10^{-4}. The EM iteration for a PND is performed until $\|\overline{\mathbf{X}} - \overline{\mathbf{X}}^{old}\|_F^2 < 10^{-7}$ and the maximum number of iterations was limited by 50, since this threshold gave better results for long and complex sequences than the threshold used in [18].

Initialization for PNDMM and *adaptive* PNDMM

Since a PNDMM consists of K PND components, we need K sub-frame sets ($\mathcal{S}_k, k = 1, \ldots, K$) for initialization. The choice of a sub-frame set selection method is discussed in Section 3.6.2. We independently initialized K PND components using K sub-frame sets, respectively, and the initial 3D mean shape $\overline{\mathbf{X}}_k$, null-space matrix $\mathbf{Q}_k$, and the reduced covariance $\mathbf{\Sigma}_{R_k}$ are computed, where the subscript k indicates that $\overline{\mathbf{X}}_k$, $\mathbf{Q}_k$, and $\mathbf{\Sigma}_{R_k}$ are computed using the corresponding kth sub-frame set. When combining independent K PND components to a PNDMM, s_{ik} and $\mathbf{R}_{ik}$ for samples $\mathbf{D}_i$ with $i \notin \mathcal{S}_k$ are initialized as follows. The rotation $\mathbf{R}_{ik}$ is initialized as $\mathbf{I}$ and the scale parameter is initialized so that the norm of $\mathbf{D}_i$ is 1, *i.e.*, $s_{ik} = \frac{1}{\|\mathbf{D}_i\|_F}$.

For an *adaptive* PNDMM, we set the prior parameter n_c to $2n_R$, meaning that the minimum value of $\sum_i w_{ik}$ needed to support component k grows linearly with the dimension of the non-rigid space, n_R. Since an *adpative* PNDMM gradually removes PND components which are not supported by samples, each PND component is required to have reasonable parameters to compete with other PND components. We run three iterations of the EM algorithm for a PNDMM and use estimated parameters to initialized an *adpative* PNDMM.

The EM or CEM iteration procedure is then performed until $\max(\|\overline{\mathbf{X}}_1 - \overline{\mathbf{X}}_1^{old}\|_F^2, \|\overline{\mathbf{X}}_2 - \overline{\mathbf{X}}_2^{old}\|_F^2, \ldots, \|\overline{\mathbf{X}}_K - \overline{\mathbf{X}}_K^{old}\|_F^2) < 10^{-6}$, and the maximum number of iterations was limited by 50. After finishing EM or CEM iterations, the final $\mathbf{M}_{ik}$ corresponding to a PND component with the maximum weight w_{ik} is used as the reconstructed 3D shape. If a shape model rather than a reconstructed shape is needed, then $\overline{\mathbf{X}}_k$ and $\mathbf{\Sigma}_k$ can be used instead [19]. Similarly, s_{ik} and $\mathbf{R}_{ik}$ can be used to represent rigid motion.

The performance was evaluated in terms of the average normalized reconstruction error:

$$e = \frac{1}{n_s} \sum_i \|\widehat{\mathbf{X}}_i - \mathbf{X}_i^*\|_F / \|\mathbf{X}_i^*\|_F, \tag{3.33}$$

where $\mathbf{X}_i^*$ and $\widehat{\mathbf{X}}_i$ are the ith ground truth and the reconstructed shape, respectively. Since the reconstructed shape has the reflection ambiguity, we also measured the error for the inverted shape and the smallest error is reported.

3.6.2 CMU Mocap database

Since the proposed algorithm is designed for long and compound human motion sequences including several simple motions, we validated it on highly non-rigid and long human motion sequences obtained from the CMU Mocap database. This database provides 41 landmark positions corresponding to human motions, however, the raw landmark positions are highly unstable, so we converted the raw landmark positions to the biovision hierarchical data (BVH) format[5], which is an easy-to-use motion capture format in computer graphics. We obtained six human motion sequences with 28 landmarks from the CMU Mocap database and they are denoted as CMU86_04, CMU86_05, CMU86_07, CMU86_08, CMU86_10, and CMU86_14. The numbered label is according

[5] http://vipbase.net/amc2bvh/

to the CMU Mocap database and the lengths of sequences vary from $6,055$ frames to $10,078$ frames. The provided description for each sequence is as follows (the number in parentheses is the length of the sequence):

- CMU86_04 (10,078 frames): walking, stretching, punching, chopping, and drinking
- CMU86_05 (8,340 frames): walking, jumping, jumping jacks, jumping on one foot, punching, and chopping
- CMU86_07 (8,702 frames): walking, swinging arms, stretching, jumping on one leg, and jumping
- CMU86_08 (9,206 frames): walking, squats, stretching, kicking, and punching
- CMU86_10 (7,583 frames): walk around, sit, stand up, and running
- CMU86_14 (6,055 frames): bouncing basketball, shooting basketball, dribble basketball, and two handed dribble.

Since sequences in the CMU Mocap dataset are sampled at 120 frames per second (fps) and they have many redundant frames, we have reduced the frame rate to 40 fps.

Initial sub-frame selection

To split n_s frames to K sets of sub-frames with each set containing about n_s/K frames, we tested two different methods on CMU Mocap sequences without any synthetic camera rotations. The first method assigns n_s frames in a continuous manner. For example, when $n_s = 10$ and $K = 3$, $\mathcal{S} = \{1, 2 \ldots, 10\}$ is split into $\mathcal{S}_1 = \{1, 2, 3\}$, $\mathcal{S}_2 = \{4, 5, 6\}$, and $\mathcal{S}_3 = \{7, 8, 9, 10\}$. The second method alternately assigns n_s frames in an interleaved manner. For the same example, $\mathcal{S} = \{1, 2 \ldots, 10\}$ is split into $\mathcal{S}_1 = \{1, 4, 7, 10\}$,

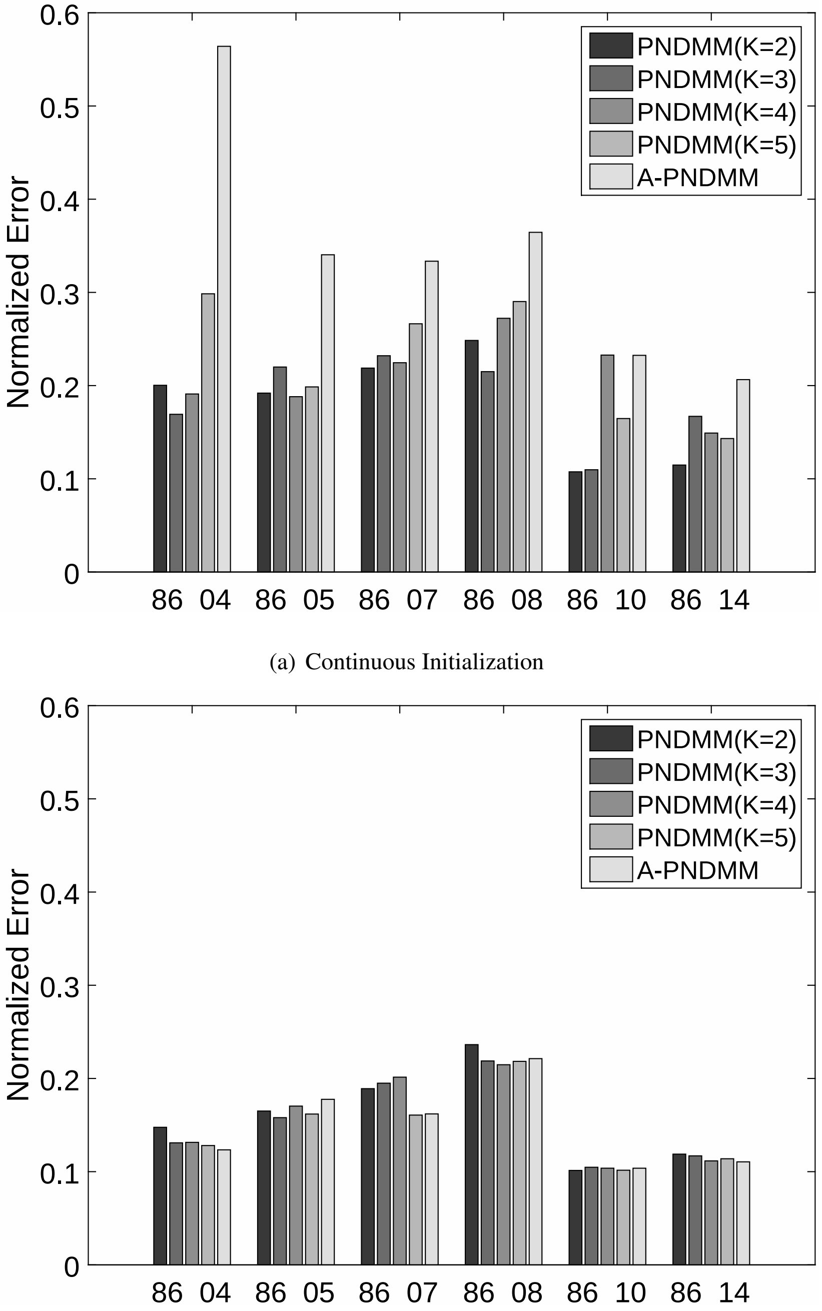
$\mathcal{S}_2 = \{2, 5, 8\}$, and $\mathcal{S}_3 = \{3, 6, 9\}$. Figure 3.4 shows that the interleaved initialization method is better than the continuous initialization method. The results can be interpreted as follows. The interleaved initialization method makes each PND component use diverse samples from the whole sequence and it can give a positive effect on longer sequences. Based on this result, we have used interleaved sub-frame sets to obtain initial parameters for PNDMM and *adaptive* PNDMM.

Effects of synthetic rotations

We also verified the reconstruction stability by adding synthetic camera rotations with 0, 0.1, 0.3, 0.5, 1, and 2 degrees per frame around the y-axis at 40 fps, and compared the proposed algorithms to PND. Figure 3.5 shows that the proposed PNDMM and *adaptive* PNDMM perform better than the PND, regardless of additional synthetic camera rotations.

For comparison with other state-of-the-art schemes, we have selected a synthetic camera rotation with 0.3 degrees per frame, which is slow and realistic camera motion [61]. The compared methods are EM-PPCA [8], MP [9], CSF2 [48], SPM [10], PND [18], and PMP [53]. In the case of CSF2 and SPM, we ran the algorithms with different numbers of shape basis vectors taken from $\{2, 3, \ldots, 9\}$[6] and reported the best result as in [48]. For CMU86_04, CMU86_05, CMU86_07, CMU86_08, CMU86_10, and CMU86_14, the best numbers of shape basis vectors for CSF2 are six, nine, six, nine, two, and eight, and the best numbers of shape basis vectors for calculating rotations in SPM are nine, eight, nine, nine, eight, and eight, respectively. Notice that this is the weakness of methods, such as CSF2 [48] and SPM [10], since in practice the ground-truth shape is not available and the best case cannot be evaluated.

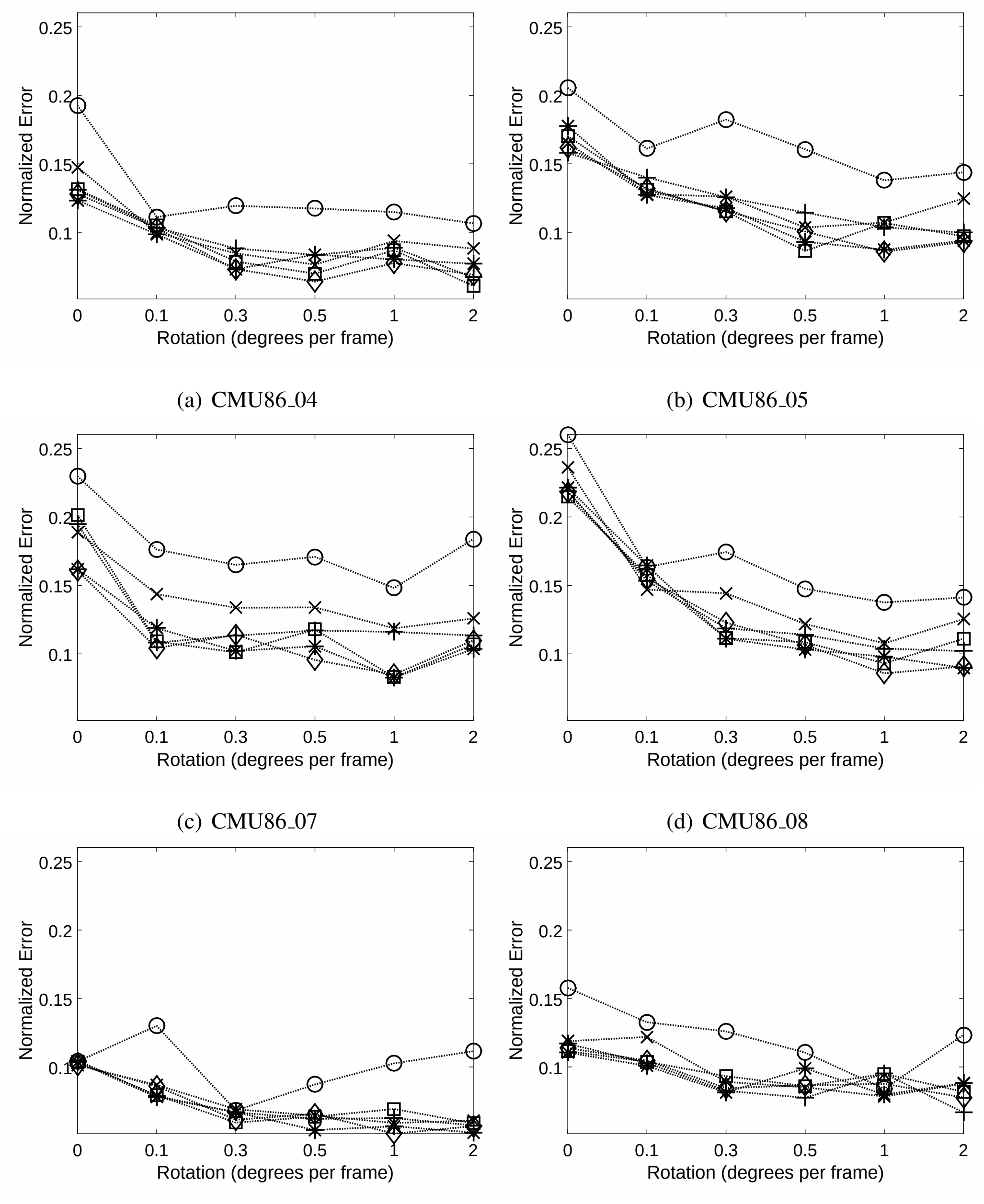
[6] The maximum number of shape basis vectors is limited up to $\lfloor \frac{28}{3} \rfloor$ when the number of landmarks is 28 [48].



(a) Continuous Initialization

(b) Interleaved Initialization

Figure 3.4: A comparison of initialization methods: continuous and interleaved. x-axis represents tested sequences, where we omit the prefix ‘CMU’. The legend A-PNDMM indicates *adaptive* PNDMM.



(a) CMU86_04

(b) CMU86_05

(c) CMU86_07

(d) CMU86_08

(e) CMU86_10

(f) CMU86_14

Figure 3.5: The effect of an increasing camera rotation on the reconstruction stability. Synthetic camera motion is added to the reduced sequences with 40 fps. The markers "○", "x", "+", "□", "◇", and "*" correspond to the PND, PNDMM with K=2, 3, 4, 5, and *adaptive* PNDMM, respectively.

Table 3.1 shows reconstruction errors of various algorithms, where K is the selected number of PND mixtures and the top three results are denoted by superscript numbers. For complex and long human motion sequences, a PNDMM significantly outperforms the other methods. Unlike a PND, a PNDMM includes a weight parameter for assigning a sample to the corresponding component, probabilistically. Owing to the weight parameter, a PNDMM is able to efficiently represent complex shape variations and achieves the state-of-the-art results on complex and long sequences.

Table 3.1: Reconstruction errors of NRSfM methods on the CMU Mocap database. (Note that (av. err.) is the average reconstruction error and (rel. err.) is the relative average reconstruction error with respect to the PND.)

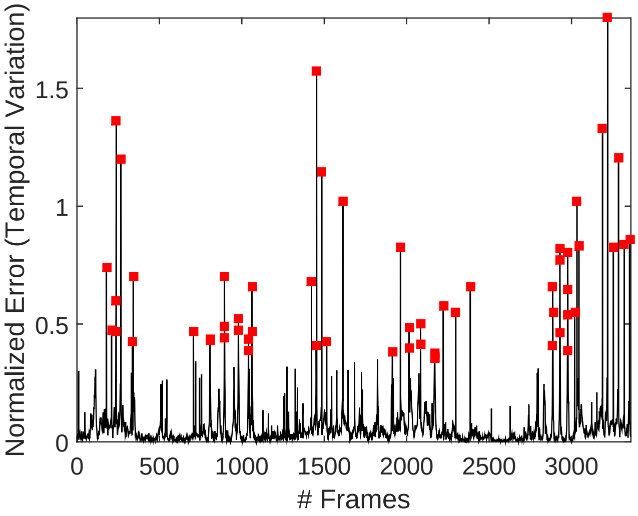
Method \ Data		CMU86_04	CMU86_05	CMU86_07	CMU86_08	CMU86_10	CMU86_14	av. err.	rel. err.
EM-PPCA		0.9953	0.7271	0.8825	1.0432	0.2560	0.6615	0.7609	5.4658
MP		0.5587	0.5776	0.5229	0.6134	0.2920	0.4628	0.5046	3.6243
CSF2		0.1842	0.2007	0.2215	0.2113	0.2912	0.1508	0.2100	1.5081
SPM		0.1421	0.1691	0.1805	0.1796	0.1390	0.1368	0.1579	1.1339
PND		0.1193	0.1823	0.1651	0.1744	0.0682	0.1260	0.1392	1.0000
PMP		0.2214	0.3045	0.2106	0.2431	0.1008	0.1843	0.2108	1.5140
PNDMM	$K = 2$	0.0846	0.1260	0.1337	0.1443	0.0691	0.0891	0.1078	0.7742
	$K = 3$	0.0882	0.1256	$0.1134^{(3)}$	$0.1187^{(3)}$	$0.0668^{(3)}$	$0.0826^{(2)}$	0.0992	0.7126
	$K = 4$	$0.0786^{(3)}$	$\mathbf{0.1152}^{(1)}$	$\mathbf{0.1011}^{(1)}$	$0.1114^{(2)}$	$\mathbf{0.0594}^{(1)}$	0.0933	0.0932	$0.6692^{(2)}$
	$K = 5$	$\mathbf{0.0730}^{(1)}$	$0.1153^{(2)}$	0.1140	0.1228	$0.0614^{(2)}$	$0.0846^{(3)}$	0.0952	$0.6838^{(3)}$
Adaptive PNDMM		$0.0731^{(2)}$	$0.1175^{(3)}$	$0.1021^{(2)}$	$\mathbf{0.1110}^{(1)}$	0.0669	$\mathbf{0.0814}^{(1)}$	0.0920	$\mathbf{0.6609}^{(1)}$
$\widehat{K}$		4	6	5	6	2	3		

* Top three results are denoted by superscript numbers.

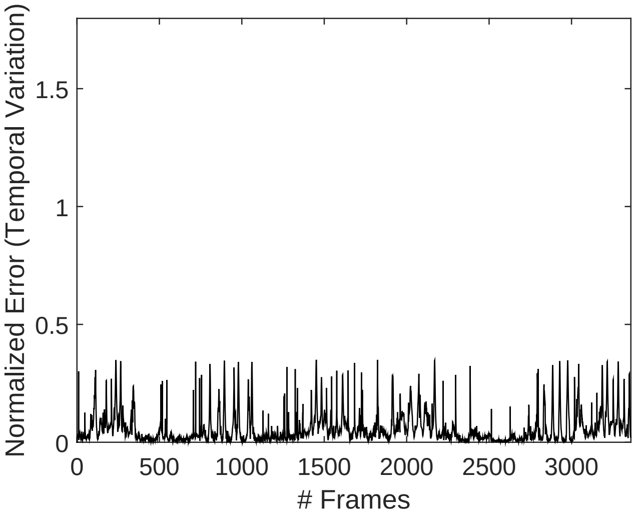
We also tested an *adaptive* PNDMM which is initialized with ten mixture components. Table 3.1 shows the results, where $\widehat{K}$ is the estimated number of components by an *adaptive* PNDMM. The *adaptive* PNDMM does not give the best results except for CMU86_08 and CMU86_14, but rather the PNDMM with the optimal number of K gives the best results. However, the optimal K for a PNDMM is explicitly set in the reconstruction process and it is hardly the case for real-world applications. Nevertheless, an *adaptive* PNDMM performs the second or third best in some cases and performs comparable for other cases. Notice that the *adaptive* PNDMM achieves the smallest relative average reconstruction error. Considering the fact that all results except for a PND and an *adaptive* PNDMM in Table 3.1 were obtained by using the optimal number of shape basis vectors tuned for the best performance, the gain for an *adaptive* PNDMM is significantly meaningful. It shows that the proposed *adaptive* PNDMM algorithm is even useful when the optimal number of shape basis vectors is unknown, unlike other algorithms.

One might ask whether using a temporal smoothing between reconstruction results could improve the performance. Unfortunately, NRSfM algorithms based on an orthographic camera model give a sign ambiguity in depth direction. Hence, we applied a temporal smoothing to reconstruction results of *adaptive* PNDMM under the assumption that the sign ambiguity is already solved. In addition, if we apply a temporal smoothing to all consecutive reconstruction results, high reconstruction errors in some frames could be propagated, thereby reconstruction results may be spoiled. To prevent the risk, we selected frames which have high variations between two consecutive frames, *i.e.*, $e_v = \|\widehat{\mathbf{X}}_{i-1} - \widehat{\mathbf{X}}_i\|_F / \|\widehat{\mathbf{X}}_{i-1}\|_F > threshold$, and applied a temporal smoothing around those selected frames as follows. Let i be the frame number of a selected frame. Then a temporal smoothing was applied from $(i-j)$th to $(i+j)$th frames. We used a simple temporal smoothing which replaces values of a selected frame with average values of previous and next frames, *i.e.*, $\widehat{\mathbf{X}}_i = (\widehat{\mathbf{X}}_{i-1} + \widehat{\mathbf{X}}_{i+1})/2$. Figure 3.6 shows effects of the temporal smoothing. The procedure of finding high variational frames and applying a temporal smoothing is repeated until the variation e_v of all frames was below 1e-6. We experimentally set $threshiold$ and the window size j to 0.3 and 7. For CMU86_04, CMU86_05, CMU86_07, CMU86_08, CMU86_10, and CMU86_14, the improved results of *adaptive* PNDMM are 0.0725, 0.1138, 0.0993, 0.1105, 0.0657, and 0.0793.

For visualization, we provide examples of reconstruction results of different algorithms in Table 3.1. As shown in Figure 3.7, two proposed algorithms show better fits between the reconstructed points and the corresponding ground truth than other algorithms.



(a)



(b)

Figure 3.6: An example of a temporal smoothing (CMU86_04). (a) and (b) show temporal variations before and after applying a temporal smoothing, respectively. In both (a) and (b), a black solid line indicates the temporal variations and red squares corresponds to frames which have high temporal variations.

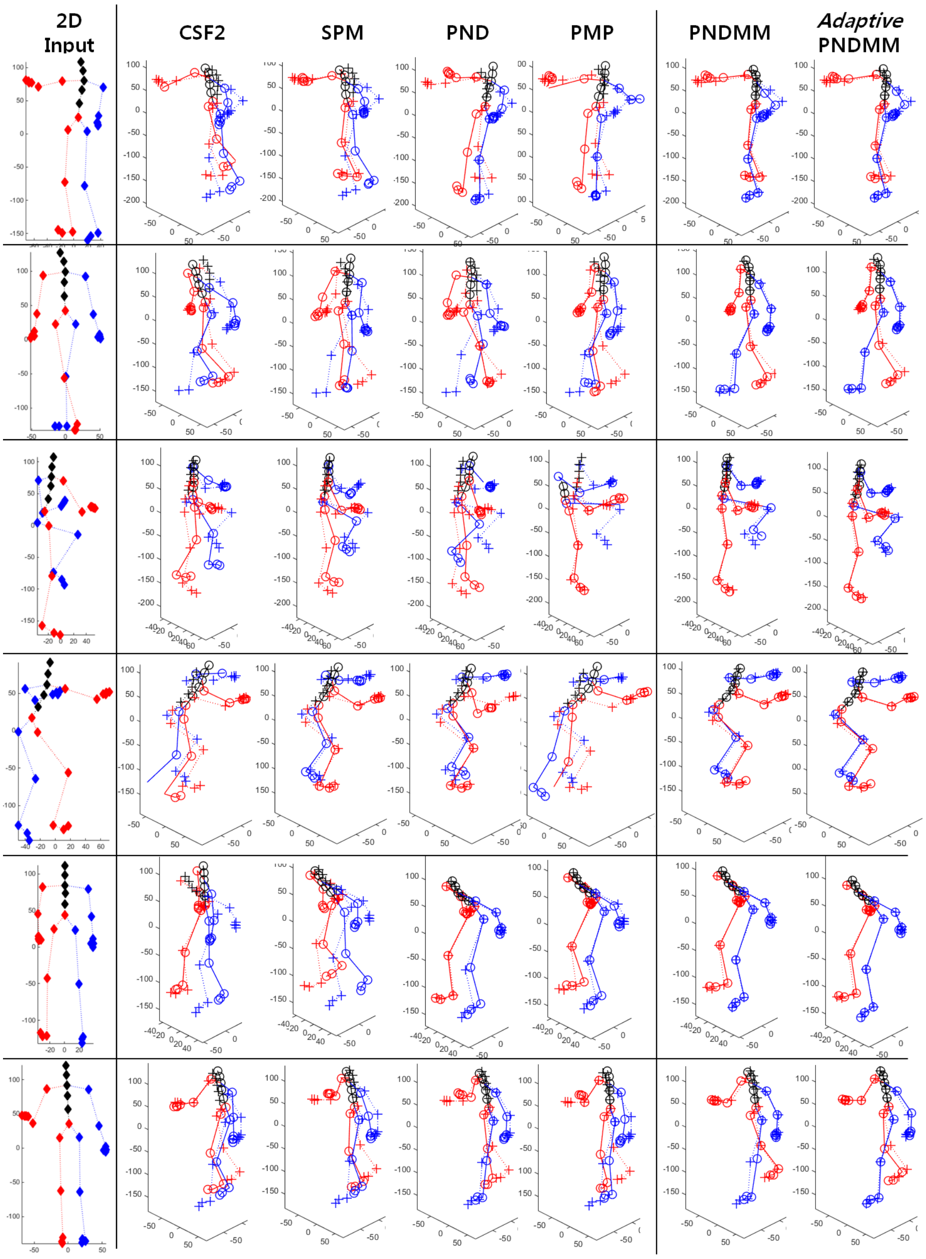


Figure 3.7: Reconstruction results from the CMU Mocap dataset experiment. “+” and “○” indicate the ground truth and the reconstruction result, respectively. From top to bottom, results are from CMU86_04, CMU86_05, CMU86_07, CMU86_08, CMU86_10, and CMU86_14. For a PNDMM, results with the best K are shown.

Figure 3.8 shows the selected component index for the whole sequence and examples of 2D input sequences. As can be seen in Figure 3.8, each component captures simple shape variations. For example, component five and six capture jumping jacks motions and component one and three capture the punching motion. It demonstrates that a PNDMM can decompose complex shape variations to simpler ones. As a result, we can handle them more efficiently. Also, due to these characteristics, a PNDMM may be used for human motion clustering in an unsupervised manner by assigning $\mathbf{X}_i$ according to a PND component with the maximum weight. Moreover, it may help getting a good understanding of the temporal structure of complex human motions.

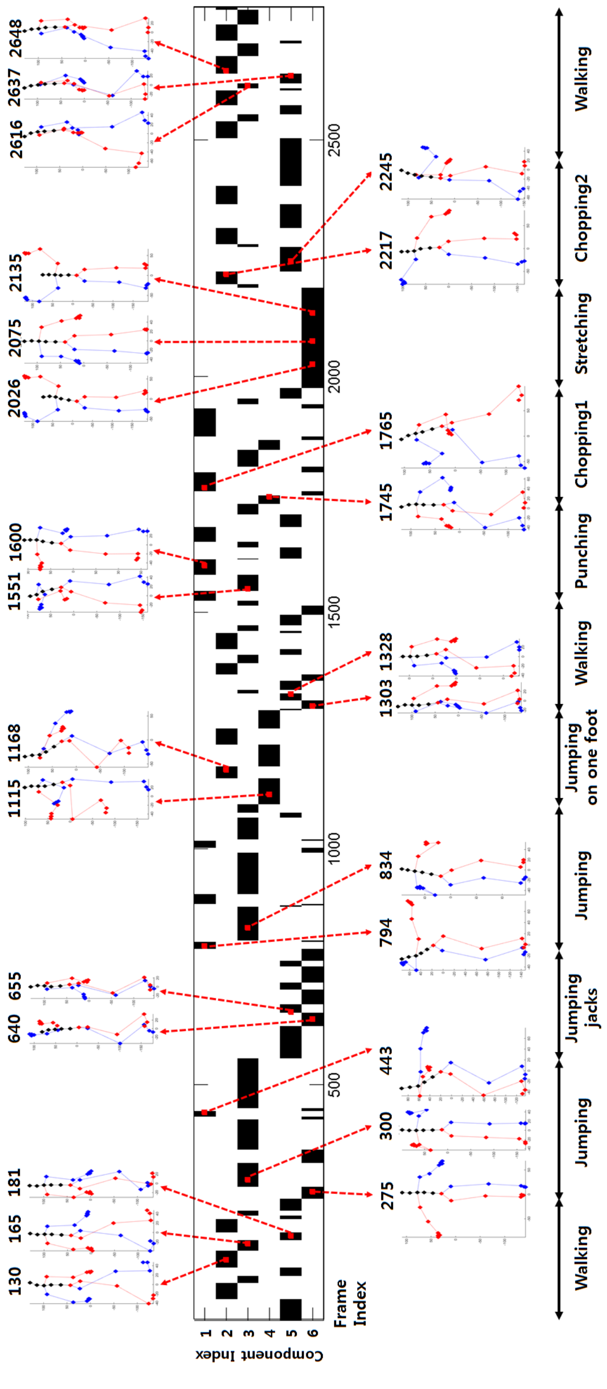


Figure 3.8: An example of human motion clustering using an *adaptive* PNDMM (CMU86_05). As the time progresses in x-axis, we show the most probable cluster index for each frame.

In our implementation using MATLAB on a PC with Intel i7-2600 CPU, the computation time (the number of EM iterations) of PNDMM ($K = 5$) and *adaptive* PNDMM on CMU86_04, which is the longest sequence, were 721.4 *sec.* (23) and 849.2 *sec.* (22), respectively. We can conclude that the small number of EM iterations are enough to obtain good 3D reconstruction results by PNDMM and *adaptive* PNDMM. Since the PND[7] has only a PND component and it does not calculate the posterior distribution of c_{ik}, it is faster than the proposed methods. The computation time (the number of EM iterations) of PND on CMU86_04 is 86.4 *sec.* (50). We also checked the computation time of CSF2[8] and SPM[9] on CMU86_04, which were 3964.3 *sec.* and 14070.6 *sec.*, respectively.

Effects of measurement noises and missing landmarks

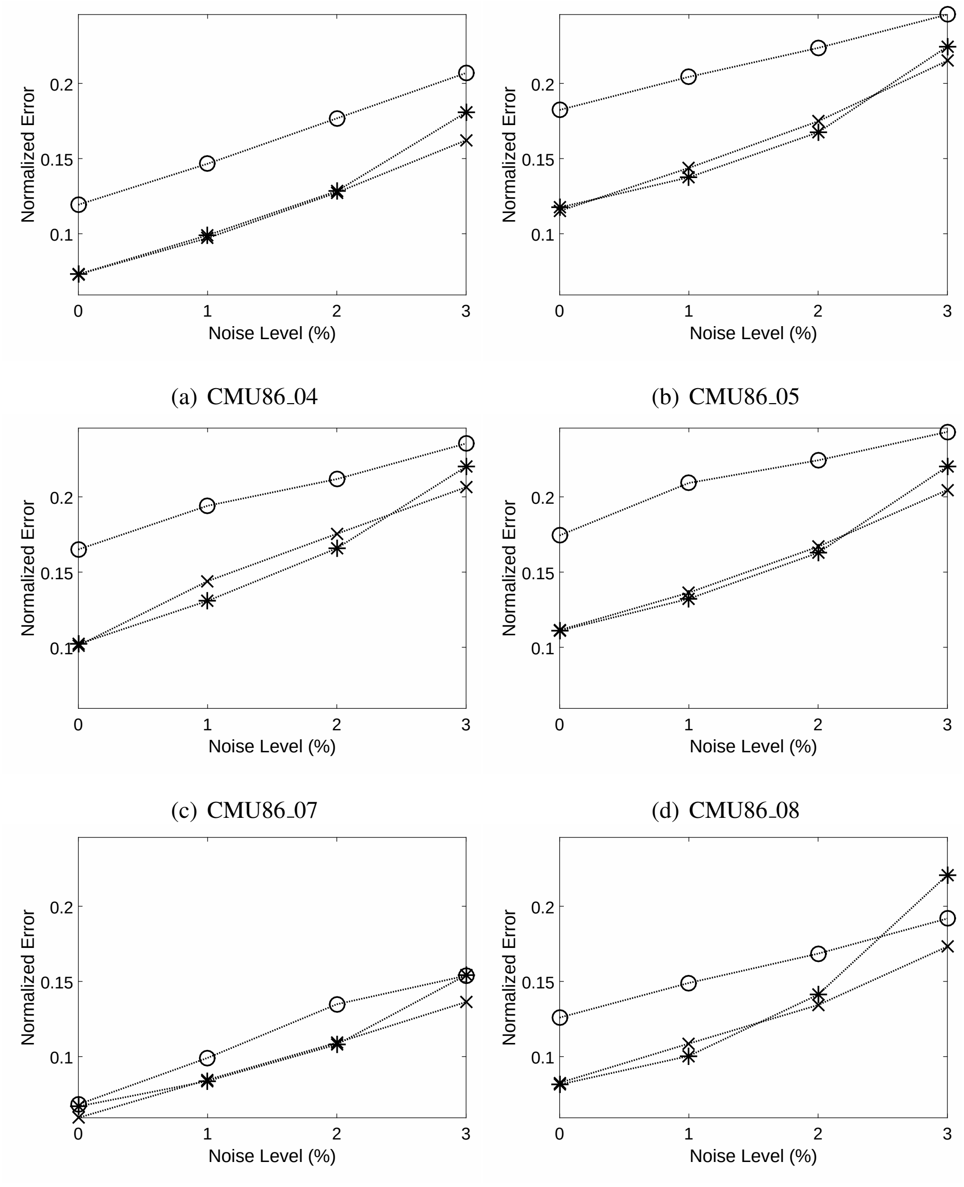
We have analyzed the proposed algorithms under different levels of measurement noises and missing measurements. To generate noisy data, observation $\mathbf{D}_i$ is corrupted by a zero-mean Gaussian noise with standard deviation $\sigma_{noise} = d_{max}\beta/100$, where β is the noise strength and $d_{max} = \max_{i,j,l}\{|d_{ijl}|\}$, where d_{ijl} is the (j, l)th element of $\mathbf{D}_i$. We have run experiments for a noise level of up to 3%. To simulate cases with missing data, each landmark was randomly set to be missing with probability $\gamma = 0$, 0.1, 0.2, 0.3, 0.4, and 0.5. With the missing probability over 0.5, the proposed method sometimes fails to reconstruct 3D shape due to the numerical instability when calculating the covariance matrix in (3.14), hence, we have reported the experiment results up to 0.5 missing probability. Each case is independently run ten times and the average values are reported. We have compared the proposed algorithms to the PND method. As shown in Figure 3.9 and Figure 3.10, the proposed methods significantly outperform the PND under various conditions. In addition, methods based on a PNDMM are not sensitive to

[7] `http://hosting01.snu.ac.kr/~cutybug/pnd/`

[8] `http://www2.ece.ohio-state.edu/~gotardop/`

[9] `http://users.cecs.anu.edu.au/~yuchao`

missing data.



(a) CMU86_04

(b) CMU86_05

(c) CMU86_07

(d) CMU86_08

(e) CMU86_10

(f) CMU86_14

Figure 3.9: Reconstruction errors at three different noise levels. The results were obtained from sequences with a synthetic camera rotation of 0.3 degrees per frame. The number of PND mixture components for a PNDMM were set as the best K based on Table 3.1. The markers "∘", "x", and "*" correspond to PND, PNDMM, and *adaptive* PNDMM.

3.6.3 UMPM dataset

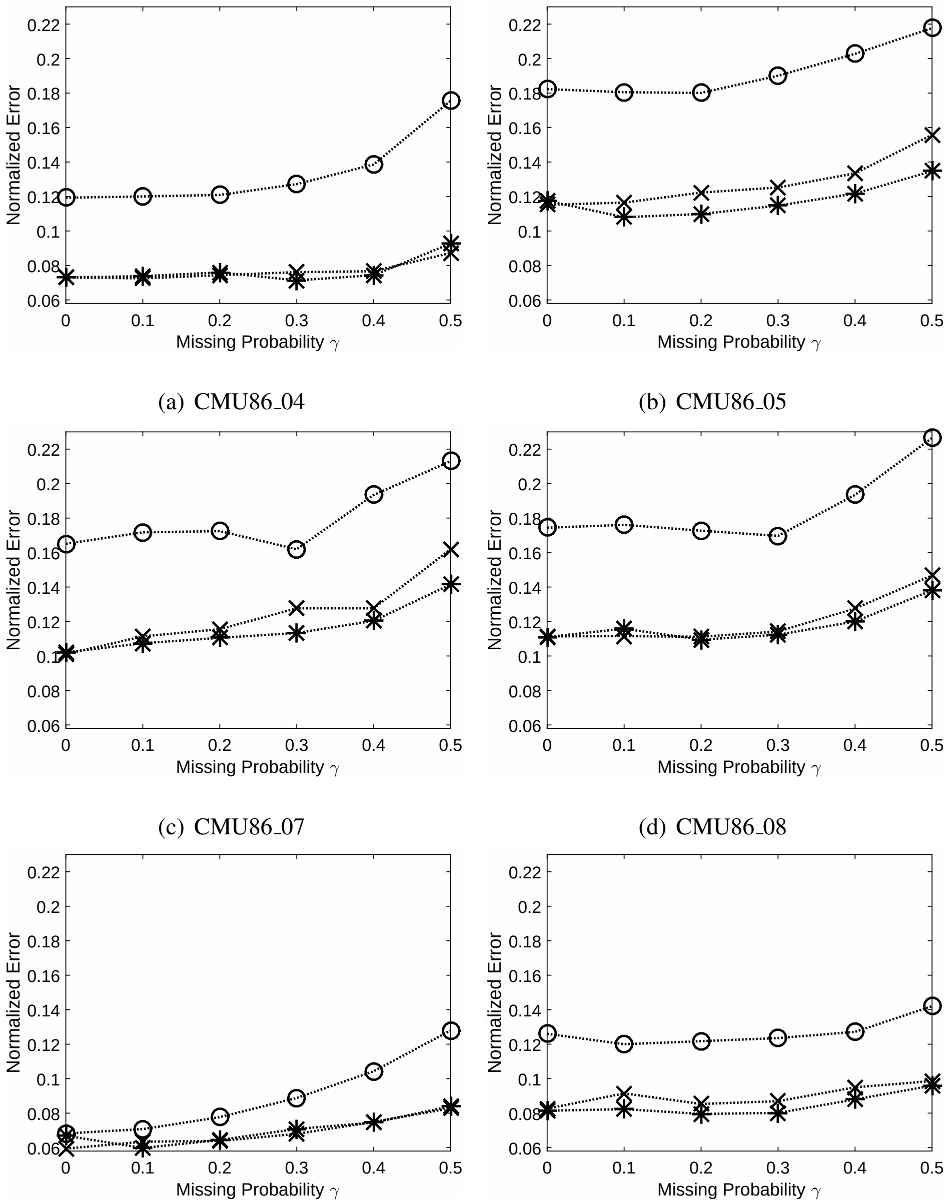
We have also validated it on the Utrecht Multi-Person Motion (UMPM) dataset[10]. The UMPM benchmark is a collection of video recordings together with a ground truth based on motion capture data [44]. To describe the bone joints, namely head, neck, shoulders, elbows, wrists, pelvis, tighs, knees, and ankles, the UMPM dataset gives two sets of 15 virtual joint positions derived from the 37 marker positions. One set gives the average joints and the other set gives joints computed by means of kinematic constraints. Since the average virtual joints include natural noises like moving markers caused by motion of clothes and it is more practical, we used the 15 average virtual marker positions. By orthographically projecting from motion capture data onto a front camera view obtained by extrinsic parameters provided from the UMPM dataset, we obtained 2D observations. Since the motion capture data has 100 fps and it is also redundant, we reduced the frame rate to 50 fps. The UMPM dataset gives marker positions of multiple persons in a sequence. However, multiple persons perform similar activities, so we used only a set of marker positions corresponding to the first person. We used six sequences and they are denoted as $p3\_ball\_12$, $p3\_chair\_16$, $p3\_triangle\_11$, $p4\_circle\_12$, $p4\_free\_11$, and $p4\_table\_11$ according to the naming convention of the UMPM dataset.[11]

We have compared the proposed methods with six algorithms [8, 9, 48, 10, 18, 53]. Note that no synthetic camera motion is introduced in this experiment. For CSF2 and SPM, we ran the algorithms with different numbers of shape basis vectors taken from $\{2, 3, \ldots, 5\}$[12] and reported the best result as in [48]. For $p3\_ball\_12$, $p3\_chair\_16$, $p3\_triangle\_11$, $p4\_circle\_12$, $p4\_free\_11$, and $p4\_table\_11$, the best numbers of shape basis vectors for CSF2 are three, four, three, five, five, and five, and the best numbers of

[10] `http://www.projects.science.uu.nl/umpm/`

[11] Naming convention: $p\langle n\rangle\_\langle a\rangle\_\langle k\rangle$, where n is the number of persons, a is the action type, and k is the take number.

[12] The maximum number of shape basis vectors is limited up to $\lfloor \frac{15}{3} \rfloor$ for 15 landmarks [48].



(a) CMU86_04

(b) CMU86_05

(c) CMU86_07

(d) CMU86_08

(e) CMU86_10

(f) CMU86_14

Figure 3.10: Reconstruction errors at three different missing data ratios. The results were obtained from sequences with a synthetic camera rotation of 0.3 degrees per frame. The number of PND mixture components for a PNDMM were set as the best K based on Table 3.1. The markers "$\circ$", "x", and "*" correspond to PND, PNDMM, and *adaptive* PNDMM.

shape basis vectors for calculating rotations in SPM are four, three, five, three, four, and four, respectively.

Table 3.2 shows reconstruction errors of various algorithms[13], where K is the selected number of PND mixtures and the top three results are denoted by superscript numbers. When an algorithm does not converge, its result is denoted by "-". Figure 3.11 shows examples of reconstruction results on the UMPM dataset. Although the proposed algorithms give better performance than other methods, the performance gaps between the PNDMM and the PND are smaller than the case with the CMU dataset.

[13] In this table, we denote six sequences as only action types.

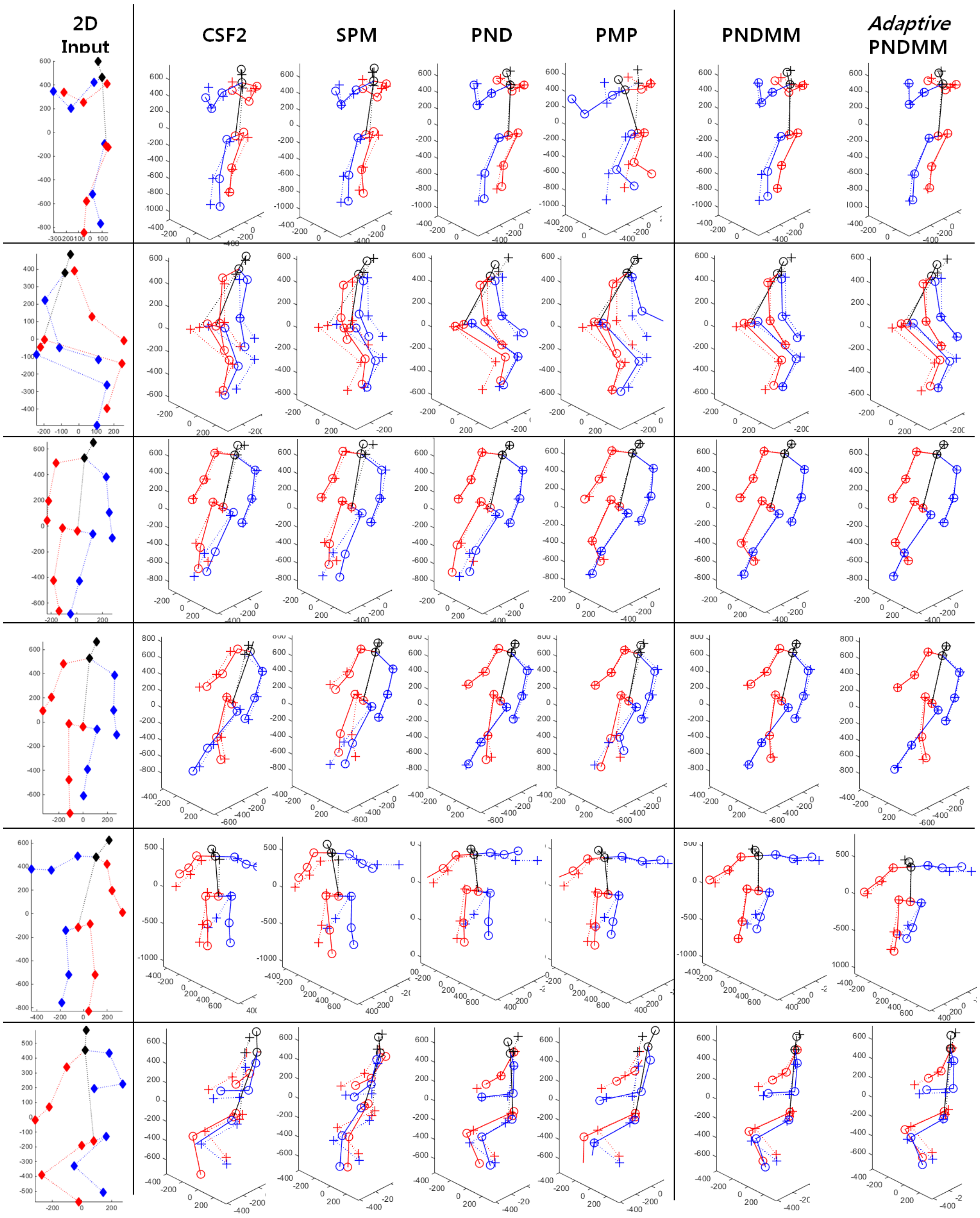


Figure 3.11: Reconstruction results from the UMPM dataset experiment. “+” and “○” indicate the ground truth and the reconstruction result, respectively. From top to bottom, results are from $p3\_ball\_12$, $p3\_chair\_16$, $p3\_triangle\_11$, $p4\_circle\_12$, $p4\_free\_11$, and $p4\_table\_11$. For a PNDMM, results with the best K are shown.

Table 3.2: Reconstruction errors of NRSfM methods on the UMPM dataset. (Note that (av. err.) is the average reconstruction error and (rel. err.) is the relative average reconstruction error with respect to the PND.)

Method \ Data		$ball$	$chair$	$triangle$	$circle$	$free$	$table$	av. err.	rel. err.
EM-PPCA		0.3207	0.2739	0.1862	0.1389	0.2605	0.3088	0.2482	1.8082
MP		0.4321	0.3666	-	0.2821	0.3514	0.3881	0.3641	2.6526
CSF2		0.2655	0.2515	0.1866	0.2748	0.2332	0.2450	0.2428	1.7688
SPM		0.2785	0.2446	0.1562	0.1437	0.2549	0.2475	0.2209	1.6095
PND		0.1574	0.1586	0.1050	0.0909	0.1527	$\mathbf{0.1588}^{(1)}$	0.1372	1.0000
PMP		0.3006	0.1752	0.1238	0.1057	0.2112	0.1908	0.1846	1.3447
PNDMM	$K = 2$	$\mathbf{0.1476}^{(1)}$	$\mathbf{0.1441}^{(1)}$	0.1000	0.0872	0.1552	0.1560	0.1317	$0.9597^{(2)}$
	$K = 3$	$0.1492^{(2)}$	$0.1482^{(3)}$	$0.0991^{(3)}$	$0.0850^{(2)}$	$0.1476^{(2)}$	$0.1589^{(3)}$	0.1313	$\mathbf{0.9569}^{(1)}$
	$K = 4$	$0.1532^{(3)}$	$0.1462^{(2)}$	0.1022	0.0911	0.1504	$\mathbf{0.1588}^{(1)}$	0.1336	$0.9738^{(3)}$
	$K = 5$	0.1648	0.1487	$\mathbf{0.0967}^{(1)}$	$0.0861^{(3)}$	$0.1489^{(3)}$	$0.1589^{(3)}$	0.1340	0.9765
Adpative PNDMM		0.1666	0.1514	$0.0981^{(2)}$	$\mathbf{0.0849}^{(1)}$	$\mathbf{0.1458}^{(1)}$	0.1594	0.1344	0.9790
$\widehat{K}$		7	7	4	6	5	7		

* Top three results are denoted by superscript numbers.

Table 3.3: Reconstruction errors of NRSfM methods on a complex sequence synthesized using the UMPM dataset. (Note that (rel. err.) is the relative average reconstruction error with respect to the PND.)

Data \ Method	PND	PNDMM				*Adaptive*	
		$K = 2$	$K = 3$	$K = 5$	$K = 10$	PNDMM	$\widehat{K}$
error	0.1557	0.1393	0.1067	0.1061	0.1193	0.1217	10
rel. err.	1.0000	0.8945	0.6851	0.6816	0.7664	0.7814	-

Actually, the UMPM dataset contains many natural rotations, whereas the complexity of human behavior is not high. To test the proposed methods on a more complex sequence, we synthesized a long sequence by concatenating all sequences at 50 fps. We removed the natural rotations by performing the Procrustes alignment for all frames in a concatenated sequence to the first frame and added a slow camera rotation of 0.3 degrees per frame. As shown in Table 3.3, the proposed methods using a PND mixture model show significant improvements over the PND.

3.6.4 Simple and short motions

We have also applied the proposed methods on simple and short motion sequences using human motion datasets provided in [45]. For simple and short human motion sequences, a PND shows better performance than a PNDMM as shown in Table 3.4. Since motion sequences provided in [45] are simple, a PND is sufficient to represent the shape distribution. Moreover, it seems that favorable results for a PND are attributed due to the nature of the small number of frames provided in [45]. In other words, the short length of a sequence makes it difficult to support more than one non-rigid shape component in a PNDMM. This model selection problem is a common issue with mixture models.

Nevertheless, the results of a PNDMM is comparable to the other methods and the reconstruction error of an *adpative* PNDMM on the dance sequence is better than that of a PND.

Table 3.4: Reconstruction errors of NRSfM methods on simple and short sequences [45]. (Note that (av. err.) is the average reconstruction error and (rel. err.) is the relative average reconstruction error with respect to the PND.)

Method \ Data		*walking*	*pickup*	*stretch*	*yoga*	*drink*	*dance*	av. err.	rev. err.
EM-PPCA		0.1485	0.5149	0.5392	0.6100	0.1292	0.2325	0.3624	10.2465
MP		0.4231	0.3465	0.5915	0.5924	0.2650	0.4062	0.4375	12.3690
CSF2		0.0708	0.0607	0.0219	0.0226	0.0123	0.1339	0.0537	1.5185
SPM		0.0861	0.0356	0.0288	0.0224	0.0216	0.1445	0.0565	1.5975
PND		$0.0410^{(2)}$	$0.0171^{(2)}$	$0.0162^{(2)}$	$0.0141^{(2)}$	$0.0031^{(2)}$	$0.1207^{(3)}$	0.0354	$1.0000^{(3)}$
PMP		$\mathbf{0.0353}^{(1)}$	$\mathbf{0.0141}^{(1)}$	$\mathbf{0.0133}^{(1)}$	$\mathbf{0.0136}^{(1)}$	$\mathbf{0.0018}^{(1)}$	$\mathbf{0.1179}^{(1)}$	0.0327	$\mathbf{0.9232}^{(1)}$
PNDMM	$K = 2$	0.0681	0.0273	0.0170	0.0163	0.0035	0.1242	0.0428	1.2088
	$K = 3$	0.0751	0.0277	0.0201	0.0201	0.0035	0.1249	0.0452	1.2790
	$K = 4$	0.0735	0.0298	0.0209	0.0191	0.0041	0.1260	0.0456	1.2888
	$K = 5$	0.0746	0.0289	0.0207	0.0195	0.0037	0.1265	0.0457	1.2919
Adaptive PNDMM		$0.0410^{(2)}$	$0.0171^{(2)}$	$0.0162^{(2)}$	$0.0143^{(3)}$	$0.0031^{(2)}$	$0.1201^{(2)}$	0.0353	$0.9978^{(2)}$
$\widehat{K}$		1	1	1	1	1	1		

* Top three results are denoted by superscript numbers.

3.6.5 Real sequence - qualitative representation

We have tested the proposed algorithm to the images with a large degree of freedom and strong self occlusion. We used the Penn Action dataset [46][14], which contains 15 different action types obtained from various online video repositories, such as YouTube[15]. This dataset includes manually annotated 2D marker positions, which consist of 13 joint positions in each video frame and their corresponding visibilities. We randomly selected two sequences for each action class with two different view points [16] and generated a long sequence by concatenating them. That is, the concatenated sequence has 15 different types of actions performed by different persons and each action has randomly selected two different view points performed by different persons.

Since SPM [10] does give not any method to handle missing variables, we cannot apply it to this experiment. As the ground truth of 3D shapes is not available, the optimal number of shape basis vectors cannot be determined for CSF2 [48], hence, we tried to use the CSF2 method [48] by adjusting the number of shape basis vectors. However, CSF2 [48] completely failed to give reconstruction results *i.e.*, the reconstruction results were quite poor and diverge. We only show reconstruction results obtained from three methods, PND, PNDMM ($K = 4$), and *adaptive* PNDMM in Figure 3.12, Figure 3.13, and Figure 3.14[17], where images (from the left to right) correspond to the 2D input image, 3D reconstruction results of PND, PNDMM, and *adaptive* PNDMM, respectively. We cane see that an *adaptive* PNDMM gives better results than that by PND and PNDMM,

[14] http://dreamdragon.github.io/PennAction/

[15] http://www.youtube.com

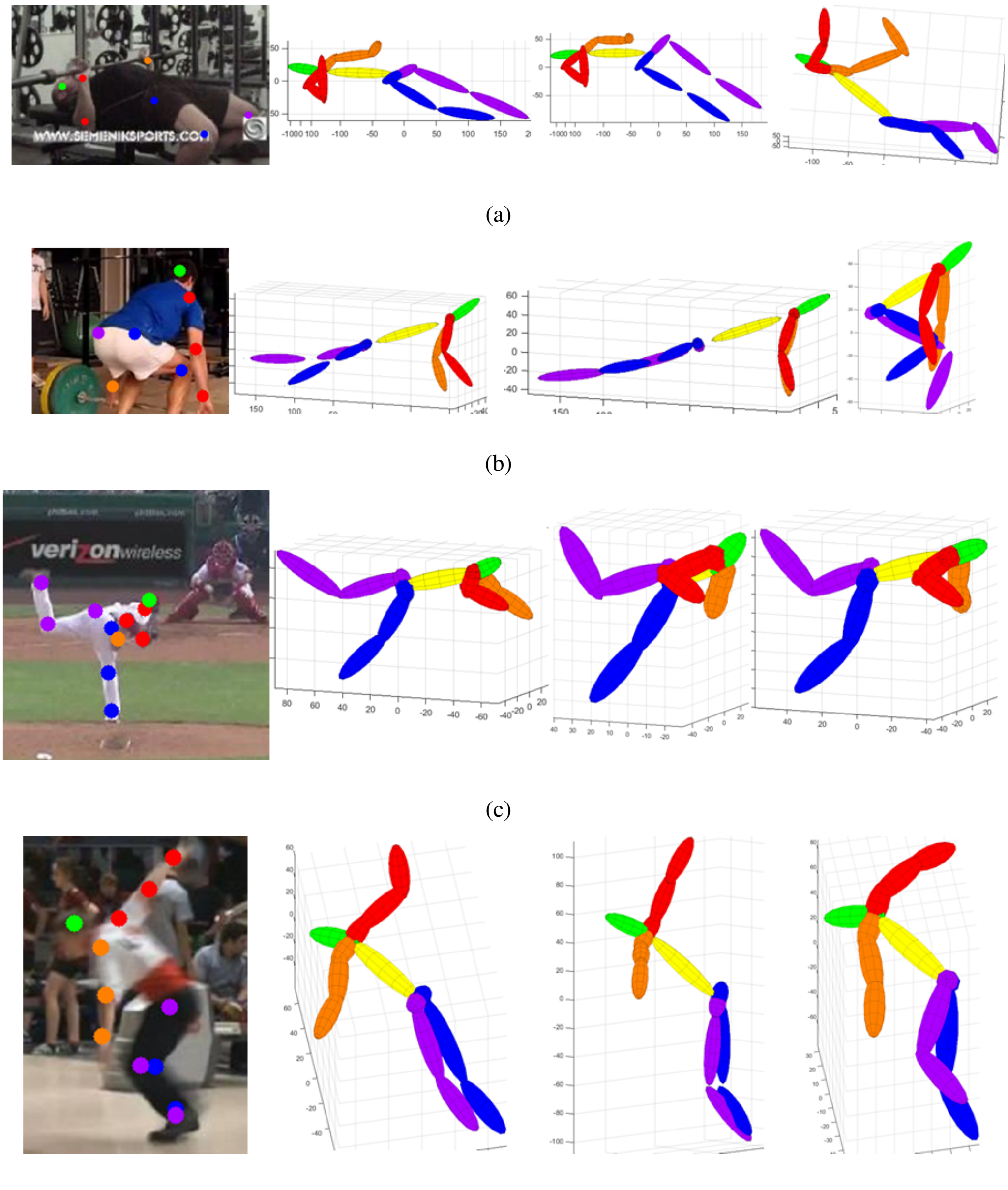
[16] The view point of each video sequence is assigned to one of four coarse camera view points, *i.e.*, front, back, left, and right.

[17] we select a more plausible result between the reconstructed 3D shape and its depth inverted version to remove the sign ambiguity. Also, we made two virtual 3D landmarks of a torso, *i.e.*, upper body and lower body, for visualization using reconstructed results, since the Penn Action dataset does not give torso landmark positions.

since the number of components of PNDMM is not set by the optimal number and a PND uses only one PND component. Failed reconstruction cases are shown in Figure 3.15 and many cases are due to many missing 2D landmark positions.

3.7 Chapter Summary

In this chapter, we have proposed a Procrustean normal distribution mixture model (PNDMM), which is a generative probabilistic mixture model to solve an NRSfM problem for complex and long human motion sequences. Unlike existing methods which use a single model to solve an NRSfM problem, we have used the fact that complex shape variations can be decomposed to a collection of simpler shape variations. The decomposition converts a complex problem into a set of simpler ones, thereby the model learning can be more tractable and accurate. We have solved NRSfM by learning a PNDMM from 2D observations of a non-rigid object using the EM algorithm and component-wise EM algorithm. Experimental results show that the PNDMM and *adaptive* PNDMM significantly outperform existing methods under various conditions using various datasets.



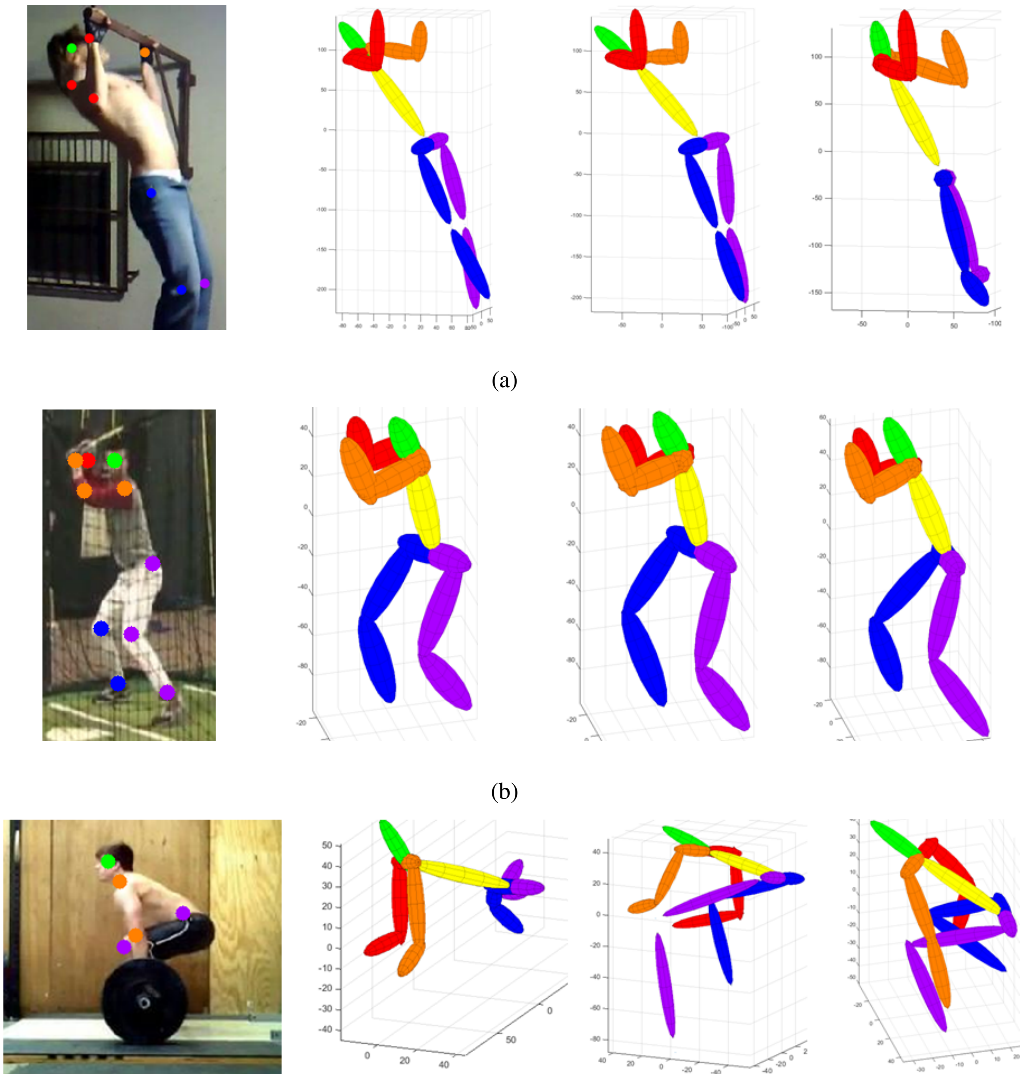
(a)

(b)

(c)

(d)

Figure 3.12: Successful reconstruction results from the Penn Action dataset. Images (from the left to right) correspond to the 2D input image, 3D reconstruction results of PND, PNDMM, and *adaptive* PNDMM, respectively. Markers "o" in a 2D input image correspond to the 2D observations and marker colors correspond to body parts according to the reconstruction results.

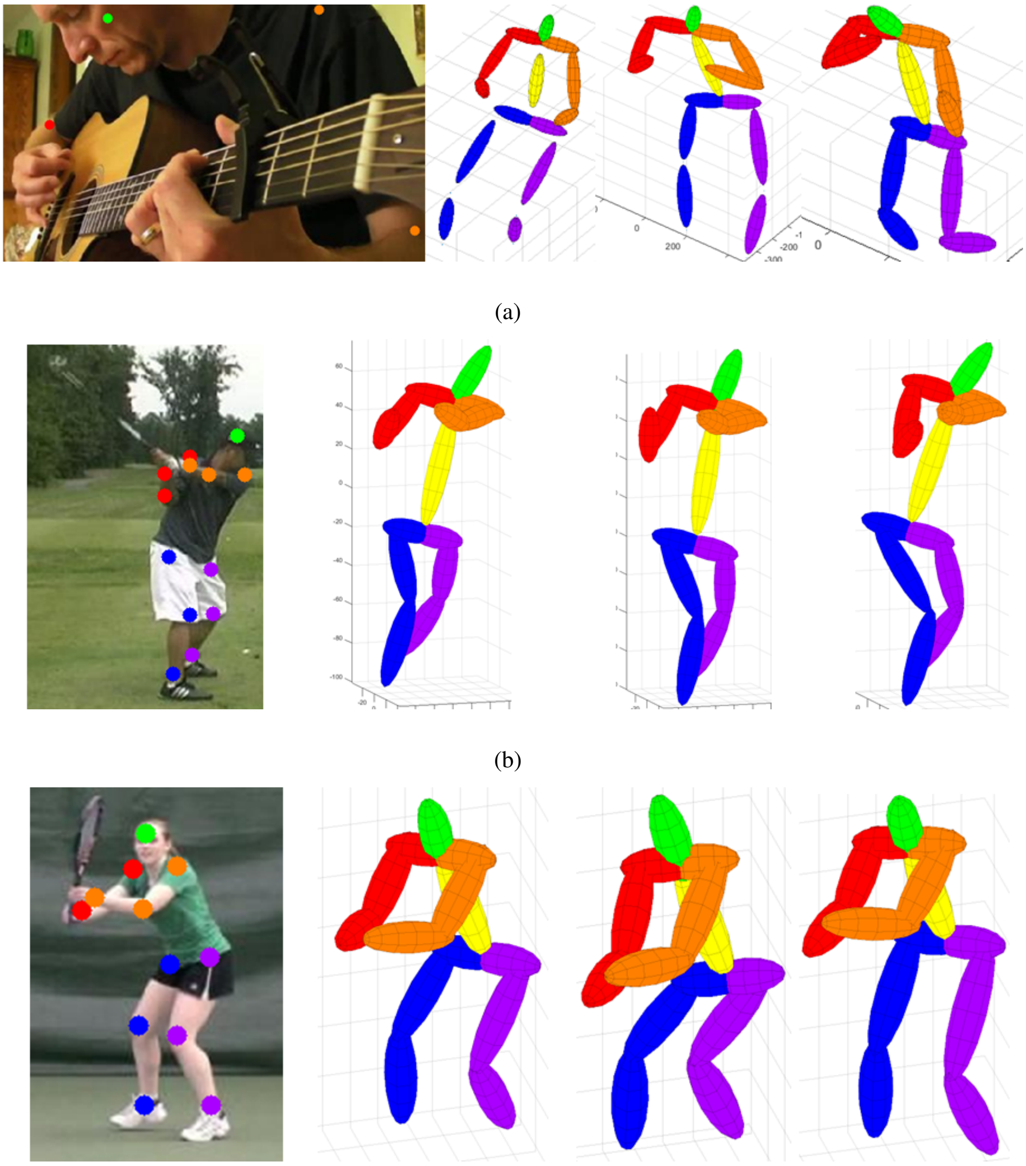


(a)

(b)

(c)

Figure 3.13: Successful reconstruction results from the Penn Action dataset (continued). Images (from the left to right) correspond to the 2D input image, 3D reconstruction results of PND, PNDMM, and *adaptive* PNDMM, respectively. Markers "o" in a 2D input image correspond to the 2D observations and marker colors correspond to body parts according to the reconstruction results.

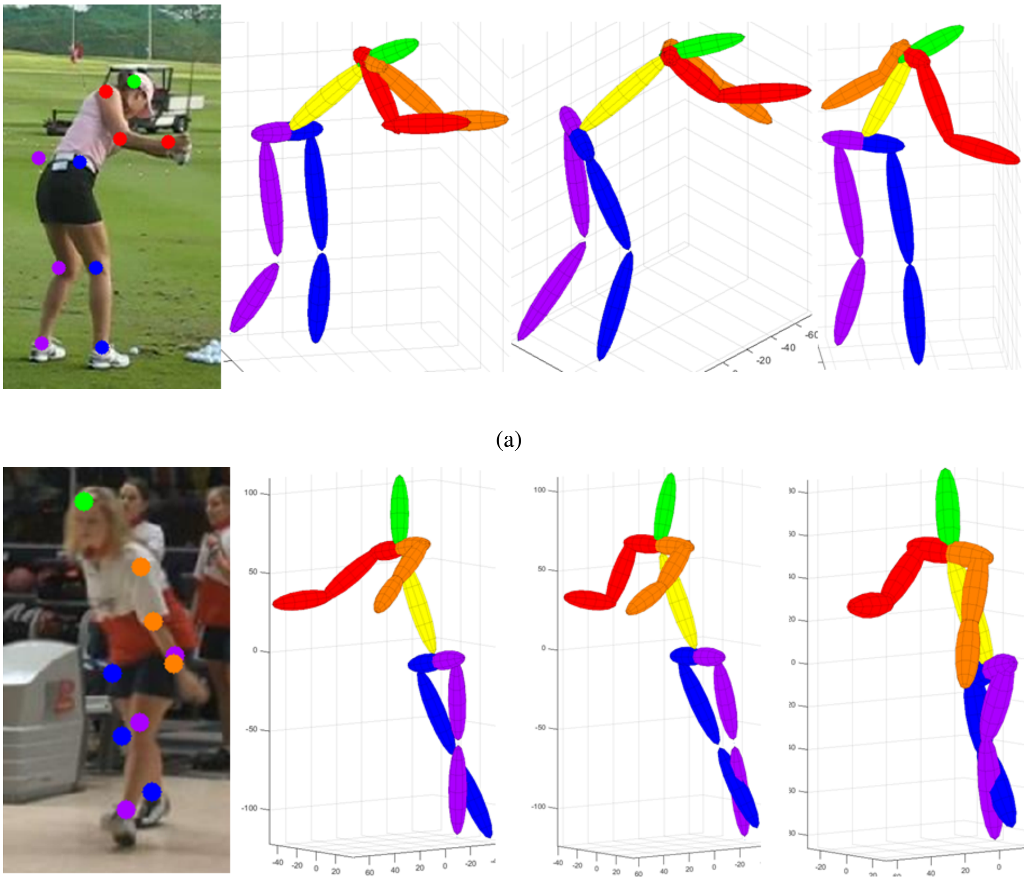


(a)

(b)

(c)

Figure 3.14: Successful reconstruction results from the Penn Action dataset (continued). Images (from the left to right) correspond to the 2D input image, 3D reconstruction results of PND, PNDMM, and *adaptive* PNDMM, respectively. Markers “o” in a 2D input image correspond to the 2D observations and marker colors correspond to body parts according to the reconstruction results.



(a)

(b)

Figure 3.15: Failed reconstruction results from the Penn Action dataset. Images (from the left to right) correspond to the 2D input image, 3D reconstruction results of PND, PNDMM, and *adaptive* PNDMM, respectively. Markers “o” in a 2D input image correspond to the 2D observations and marker colors correspond to body parts according to the reconstruction results.

Chapter 4

Recovering a 3D Human Pose from a Novel Image

In this chapter, we handle an extension of the PNDMM for single view 3D human pose estimation. While there are various approaches for monocular 3D human pose estimation [62, 63, 64, 65, 66, 11, 67, 68, 12, 69, 70, 71], our work focuses on recovering a 3D human pose using 2D part locations obtained from an image, since the use of the 3D shape model with 2D part locations makes the proposed method more robust against changes in viewpoint.

The overall structure of the proposed method is shown in Figure 4.1. In order to handle inaccuracies of 2D part detections, we generate a diverse set of 2D part candidates by decomposing and recombining multiple 2D part detections, and then select the one which explains the 2D part model and 3D shape model the best. To overcome the complexity of human shapes and noisy observations, we apply the Procrustean normal distribution (PND) [18] as a probability model for non-rigid shape variations. Since our goal is to estimate a 3D human pose from a single image, we learn the prior information about 3D configurations in a form of a mixture of PNDs or a Procrustean normal distribution

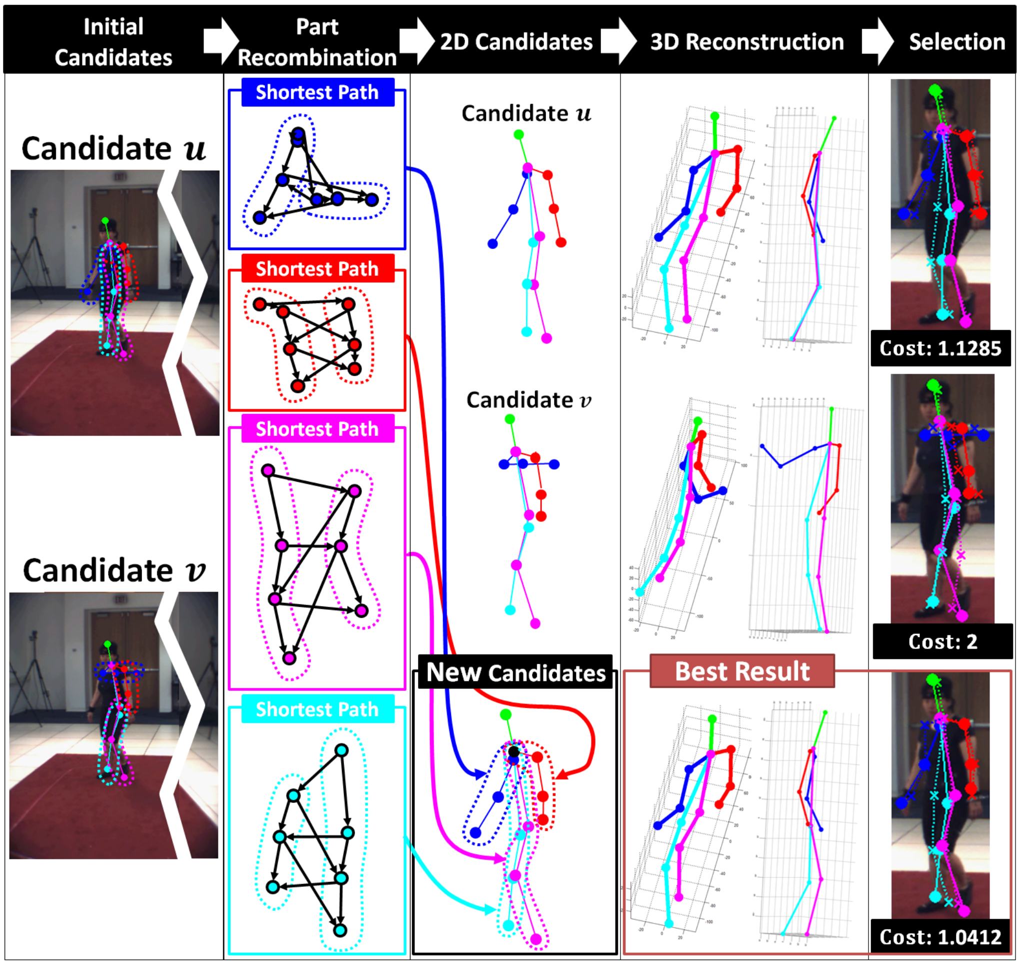


Figure 4.1: An overview of the proposed method.

mixture model (PNDMM) and fit the mixture model to 2D observations. The PNDMM plays a role of making a set of specific pose subspaces in unsupervised manner. By restricting 3D pose estimation on a subspace, performances of 3D pose estimation can be improved.

When a learned 3D shape model is applied to a novel image, a problem can arise since the test image may contain a human subject which has limb lengths significantly different from subjects in the training set. When working with a single image, we cannot recover limb lengths of the new subject. Hence, we propose a model transformation method which consists of model normalization and model adaptation. In the model normalization step, we normalize limb lengths of mixture components using their mean limb lengths. The model adaptation step adjusts the normalized model using the initial 3D human pose estimated by the proposed method.

From an extensive set of experiments, we show that the proposed method performs favorably against the state-of-the-art methods by overcoming inaccuracies of 2D part detections and 3D shape ambiguities. In addition, when the proposed method is applied to a novel test set, which is different from the training set, the proposed method performs the best, showing its generalization power.

Relation with other chapters Since a 3D human pose has more information than a 2D human pose, intuitively, action recognition using 3D human poses would be more robust in complex scenes. In Chapter 5, we will show an application on action recognition using estimated 3D human poses.

4.1 Single View 3D Human Pose Estimation

The problem of estimating a 3D human pose from a single image can be considered as a structured output regression problem based on 2D features, such as silhouettes

[62, 63, 64, 65, 66]. Agarwal *et al.*[62] have estimated 3D human poses from 2D silhouettes by using a Relevance Vector Machine (RVM) regressor, which is a sparse Bayesian nonlinear regression. Sigal *et al.*[63] have proposed a parameterized triangulated mesh model for 3D human pose estimation, in which parameters are initialized using a conditional mixture of kernel regressor based on silhouettes. Bo *et al.*[65] have proposed a Twin Gaussian process (TGP), which minimizes the Kullback-Leiber divergence between two Gaussian processes of input and output data. While silhouette-based regression methods have shown excellent performance, obtaining a body silhouette from a single image is difficult in practice. Furthermore, these methods are inherently limited by the amount and quality of the training data, since they require a large number of training samples to represent the appearance variability of different people and viewpoints unlike part detection based methods.

Recently, methods based on a 2D part detection algorithm are proposed to fit detected 2D joint locations to a 3D shape model [11, 12, 69]. While the accuracy of 2D body part detection can greatly affect the performance of 3D pose estimation, currently available part detection algorithms frequently report incorrect body parts. In order to overcome inaccuracies in a part detection algorithm, Simo-Serra *et al.*[11] used a stochastic sampling strategy which propagated 2D observation noises to the 3D shape space. Radwan *et al.*[12] generated 2D part locations in synthetic views by regressing a set of 2D part locations from the input view to multiple oriented views. Then the pose was estimated using multi-view geometry. Wang *et al.*[69] minimized the l_1-norm error between the projection of an estimated 3D pose and corresponding 2D detections, in which the 3D pose was represented as a linear combination of a sparse set of basis vectors of human poses.

Our work addresses this issue by utilizing a diverse set of 2D pose candidates and a sound 3D shape prior model.

4.2 Candidate Generation

4.2.1 Initial pose generation

Yang and Ramanan [6] have proposed a 2D human pose estimation method by representing human body parts as a mixture of pictorial structures. Let $G = (V, E)$ be a relational tree, where V is a set of body parts and E is a set of edges connecting body parts. Then the score of a specific pose configuration is represented as follows [6]:

$$S(I, z) = \sum_{j \in V} \Phi_j(I, z_j) + \sum_{(i,j) \in E} \Psi_{ji}(z_j, z_i), \tag{4.1}$$

$$\begin{aligned} \text{where} \quad & z_j = (l_j, t_j) \\ & \Phi_j(I, z_j) = w_j^{t_j} \cdot \phi(I, l_j) + b_j^{t_j} \\ & \Psi_{ji}(z_j, z_i) = w_{ji}^{t_j, t_i} \cdot \psi(l_j - l_i) + b_{ji}^{t_j, t_i}. \end{aligned}$$

For an edge (i, j), i denotes a parent node and j denotes a child node. Here, l_j is the location of part j and t_j is the configuration type for part j, *e.g.*, different hand appearances due to its orientation. The first sum represents the sum of local appearance scores computed by pre-trained template $w_j^{t_j}$ and HOG [72] descriptor $\phi(I, l_j)$ extracted at location l_j in image I. The second sum encodes shape deformations by $w_{ji}^{t_j, t_i}$, which is often interpreted as a spring between adjacent parts, and $\psi(l_j - l_i)$, which is the relative location of part j with respect to part i. $b_j^{t_j}$ and $b_{ji}^{t_j, t_i}$ are trained offsets. We can efficiently find z^* which maximizes $S(I, z)$ using dynamic programming [6] by sequentially optimizing from leaf nodes to the root node. The method has been extended to generate N-best candidates [73], which can be used to find multiple detections anchored at the same root.

Since 3D pose estimation using a single image can be highly ambiguous, it is important to use accurate 2D part detection results. To avoid reconstructing 3D poses based on incorrect part detection results, we utilize 2D pose candidates with high scores. After performing part specific non-maximum suppression, we select n_c 2D pose candidates with the highest pose detection scores. Additional pose candidates are generated from n_c candidates using the part recombination step described below.

4.2.2 Part recombination

For each pose candidate selected from the N-best extension [73], we decompose a pose into four segments: a left arm, right arm, left leg, and right leg (see Figure 4.1). All generated segments share the common neck and head and the positions of the neck and head are obtained from the part detection with the highest score. For each segment, a new segment is generated using corresponding segments from all candidates by solving a shortest path problem (see Figure 4.1 under Part Recombination).

For each segment, if there is an edge $e_{u_i,u_j} = (u_i, u_j)$, we introduce new directed edges e_{u_i,v_j} for all candidates u and v, where u_i and v_j are the ith and jth joints from the uth and vth candidates, respectively (see Figure 4.1). For segment s, let E_s be a set of all edges introduced for the segment. Let $\mathcal{P}_s$ be a set of all possible paths in E_s. Then we solve the following shortest path problem:

$$\min_{path \in \mathcal{P}_s} \sum_{(u_i,v_j) \in path} f(e_{u_i,v_j}), \tag{4.2}$$

where f consists of a part cost and neighborhood cost.

Part cost: We define the part cost for part i using the detection score of appearance, *i.e.*, $S_i = \Phi_i(I, z_i)$ in (4.1). We look for a path in E_s with high part detection scores, hence, we define an edge cost as follows:

$$f_p(e_{u_i,v_j}) = -S_i^u - S_j^v, \tag{4.3}$$

where S_i^u is the score for part i of candidate u.

Neighborhood cost: The neighborhood cost introduces constraints on limb lengths and it is defined as

$$f_n(e_{u_i,v_j}) = \left| dist(l_i^u, l_j^v) - \bar{l}_{ij}^{train} \right|, \tag{4.4}$$

where $dist(a, b)$ is the Euclidean distance between a and b, l_i^u is the location of part i of candidate u, and $\bar{l}_{ij}^{train}$ is a reference length of the limb obtained from the 2D part training data.

Since ranges of the part cost and neighborhood cost are not the same, we normalize their values between 0 and 1 using costs from the candidate poses. Let f_p' and f_n' be the normalized part cost and neighborhood cost, respectively. The shortest path problem (4.2) is solved with

$$f(e_{u_i,v_j}) = f_p'(e_{u_i,v_j}) + f_n'(e_{u_i,v_j}). \tag{4.5}$$

When all segments are generated by solving (4.2), we combine them to make an additional candidate pose with the common neck position.

Candidate generation: Since the discussed part recombination step gives a single candidate pose, we generate a diverse set of candidate poses by performing the part recombination step repeatedly with a different set of initial candidates. The first recombined candidate is found using all n_c initial candidates and the second recombined candidate is found using n_c-1 initial candidates by removing the candidate with the highest detection score. We repeat the process until at least two initial candidates are remained. In total, $n_c - 1$ recombined candidates are generated and will be considered for 3D reconstruction along with n_c initial candidates.

4.3 3D Shape Prior Model

4.3.1 Procrustean mixture model learning

To model the deformation of 3D shapes, we use the Procrustean normal distribution (PND) [18], which makes 3D shapes closely aligned in a linear subspace. The PND can be extended to a mixture of PNDs and the resulting Procrustean normal distribution mixture model (PNDMM) can be expressed as:

$$p(\mathbf{X}) = \sum_{k=1}^{K} \pi_k p(\mathbf{X}|c_k = 1), \tag{4.6}$$

where $\mathbf{X} \in \mathbb{R}^{3\times n_p}$ is a 3D shape satisfying $\mathbf{X1} = \mathbf{0}$[1], n_p is the number of landmarks, and K is the number of mixture components. The mixing probability for the kth component is defined as $\pi_k = p(c_k = 1|\pi_k)$, where $\pi_k \geq 0$, such that $\sum_k \pi_k = 1$ and $c_k \in \{0, 1\}$ indicates which mixture component has generated the sample. $p(\mathbf{X}|c_k = 1)$ is a PND corresponding to the kth component, which is defined as $\mathcal{N}_P(\mathbf{Y}|\overline{\mathbf{X}}_k, \mathbf{Q}_k\mathbf{\Sigma}_{R_k}\mathbf{Q}_k^T)$, where $\overline{\mathbf{X}}_k$, $\mathbf{\Sigma}_{R_k}$, and $\mathbf{Q}_k$ are the mean of aligned 3D shapes, the covariance matrix for non-rigid variations, and the projection matrix to the linear subspace of non-rigid shapes, respectively [18]. In addition, $\mathbf{Y} = s\mathbf{RX}$ is an aligned shape using scale s and rotation $\mathbf{R}$.

Since we do not know the true number of mixture components, we introduce a Dirichlet-type prior on π based on the minimum message length (MML) principle [42]: $p(\pi) \propto \exp\left(-\frac{n_l}{2}\sum_k \ln \pi_k\right)$. Then the parameters of a PNDMM can be learned with N training 3D shapes, similar to Chapter3, by maximizing the following expected complete log-posterior using the expectation-maximization (EM) algorithm:

$$\begin{aligned}\Upsilon(\Phi|\Phi^{old}) = &\sum_i \sum_k w_{ik} \ln(p(\mathbf{X}_i, c_{ik} = 1|\Phi)) \\ &+ \sum_k \ln(p(\pi_k)),\end{aligned} \tag{4.7}$$

where $\Phi = \{s_{ik}, \mathbf{R}_{ik}, \overline{\mathbf{X}}_k, \mathbf{\Sigma}_{R_k}, \mathbf{Q}_k, \pi_k | i = 1, \ldots, N, k = 1, \ldots, K\}$ are model parameters for a PNDMM, and the indeces i and k correspond to the ith training sample and kth PND component, respectively. The superscript old denotes the parameter set obtained from the previous M-step in the EM iteration procedure.

[1] In this chapter, we use $\mathbf{0}$ to denote both a matrix and a vector of zeros and $\mathbf{1}$ denote a vector of ones.

We denote $p(c_{ik} = 1|\mathbf{X}_i, \Phi^{old})$ as a weight w_{ik}, since the posterior distribution of c_{ik} plays the role as a weight for the component indicated by $\mathbf{c}_i$. Note that w_{ik} are computed in the E-step of the EM algorithm. The optimization in the M-step can be done similar to [18] and the learned parameters $(\overline{\mathbf{X}}_k, \mathbf{\Sigma}_{R_k}, \mathbf{Q}_k)$ of PND components using 3D training data are used as the parameters $(\overline{\mathbf{X}}_k^{train}, \mathbf{\Sigma}_{R_k}^{train}, \mathbf{Q}_k^{train})$ of a 3D prior model when we fit a PNDMM to a 2D pose candidate from a single image. (For more details, please see the Chapter C.)

4.3.2 Procrustean mixture model fitting

Unlike the prior model learning step discussed in the previous section, observations for our problem is not a 3D shape $\mathbf{X}$, but a 2D shape $\mathbf{D} \in \mathbb{R}^{2\times n_p}$. We treat $\mathbf{X}$ as a hidden variable and estimate $\mathbf{X}$ and regard the observation as a sample obtained by a noisy orthographic projection of $\mathbf{X}$ with a zero mean Gaussian noise with variance σ^2 in each coordinate. The fitting problem can be solved by using the EM algorithm based on the trained PNDMM. (For more details, please see the Chapter C.)

The overall EM procedure is similar to that in Chapter 3. However, in our case, the contribution of a single data sample has little effects on calculating the PND model parameters $(\overline{\mathbf{X}}_k^{train}, \mathbf{\Sigma}_{R_k}^{train}, \mathbf{Q}_k^{train})$. Hence, we fix those parameters and only update s_k, $\mathbf{R}_k$, π_k, and σ^2 in the M-step, as done in Chapter 3. If $\pi_k = 0$, the kth PND component is removed. After finishing EM iterations, the final posterior mean shape corresponding to the PND component with the maximum weight w_k is used as a reconstructed 3D shape $\widehat{\mathbf{X}}$.

4.4 Model Transformation

4.4.1 Model normalization

Given a novel image, the limb length of a subject may differ from the limb lengths of subjects in the training set. To handle this issue, we normalize the limb length information in the trained PNDMM model. Let $l_{ij}^{k} = \|\overline{\mathbf{X}}_k(i) - \overline{\mathbf{X}}_k(i)\|_2$ be the limb length between part i and part j, where $\overline{\mathbf{X}}_k(i)$ is the ith column vector of $\overline{\mathbf{X}}_k$. We calculate mean lengths between body parts as $\bar{l}_{ij} = \frac{1}{K}\sum_k l_{ij}^{k}$ and adjust lengths l_{ij}^{k} between body parts to $\bar{l}_{ij}$ in all components of the trained PNDMM. The length adjustment process uses the following fact.

Proposition 5. *Let* $\mathbf{J}$ *be an* $n_p \times (n_p - 1)$ *full column rank matrix satisfying* $\mathbf{1}^T\mathbf{J} = \mathbf{0}$. *Then* $\mathbf{J}\mathbf{J}^+ = \mathbf{I} - \frac{1}{n_p}\mathbf{1}\mathbf{1}^T$. [2]

Proof.

$$\mathbf{J} = \mathbf{U}\mathbf{S}\mathbf{V}^T \text{(sinular value decomposition).} \tag{4.8}$$

$$\mathbf{J}\mathbf{J}^+ = \mathbf{U}\mathbf{S}\mathbf{V}^T\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T = \mathbf{U}\mathbf{U}^T. \tag{4.9}$$

Since $\mathbf{1}^T\mathbf{J} = \mathbf{0}$, $\mathbf{1}$ is in the null space of $\mathbf{J}^T$. Then

$$\begin{bmatrix}\mathbf{U} & \frac{1}{\sqrt{n_p}}\mathbf{1}\end{bmatrix}^T \begin{bmatrix}\mathbf{U} & \frac{1}{\sqrt{n_p}}\mathbf{1}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{0} \\ 0 & 1\end{bmatrix} = \mathbf{I}, \tag{4.10}$$

where $\mathbf{U}$ is a left-singular vector of $\mathbf{J}$. Therefore, $\begin{bmatrix}\mathbf{U} & \frac{1}{\sqrt{n_p}}\mathbf{1}\end{bmatrix}$ is a full rank orthogonal matrix. Then

$$\mathbf{I} = \begin{bmatrix}\mathbf{U} & \frac{1}{\sqrt{n_p}}\mathbf{1}\end{bmatrix} \begin{bmatrix}\mathbf{U} & \frac{1}{\sqrt{n_p}}\mathbf{1}\end{bmatrix}^T = \mathbf{U}\mathbf{U}^T + \frac{1}{n_p}\mathbf{1}\mathbf{1}^T.$$

Hence, we have $\mathbf{J}\mathbf{J}^+ = \mathbf{U}\mathbf{U}^T = \mathbf{I} - \frac{1}{n_p}\mathbf{1}\mathbf{1}^T$. □

[2] $\mathbf{J}^+$ is the Psedudo-inverse of $\mathbf{J}$.

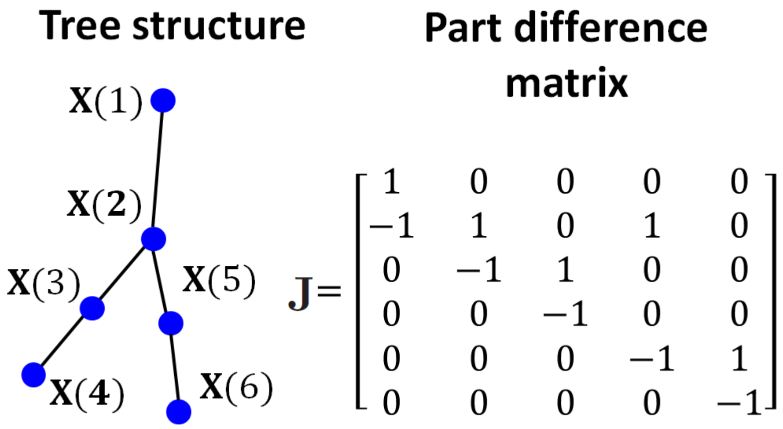


Figure 4.2: An example of a tree structure and the corresponding part different matrix $\mathbf{J}$.

Proposition 5 shows that $\mathbf{J}\mathbf{J}^+$ only removes a translation component of a given shape and preserves angles between body parts, thereby preserving the pose, *i.e.*, $\mathbf{X}\mathbf{J}\mathbf{J}^+ = \mathbf{X}(\mathbf{I} - \frac{1}{n_p}\mathbf{1}\mathbf{1}^T)$.

Let $\mathbf{J}$ be a part difference matrix defined by a tree structure (an example is shown in Figure 4.2). Then $\mathbf{J}$ is a full column rank matrix.

Let $\mathbf{T}_d^k \in \mathbb{R}^{(n_p-1)\times(n_p-1)}$ be a diagonal matrix, where each diagonal entry is a length scale between $\bar{l}_{ij}$ and l_{ij}^k with appropriate indices i and j given by $\mathbf{J}$. Let $\mathbf{T}_s^k$ be a swapped matrix of $\mathbf{T}_d^k$ by swapping entries based on physically symmetric pairs of a human body (in Figure 4.2, l_{23} and l_{25} can be swapped with l_{34} and l_{56}, respectively). Then we can obtain a physically symmetric diagonal matrix $\mathbf{T}^k = \frac{\mathbf{T}_d^k + \mathbf{T}_s^k}{2}$. Finally, the normalized mean shape for the kth PND is $\overline{\mathbf{X}}_k^{nor} = \overline{\mathbf{X}}_k^{train}(\mathbf{J}\mathbf{T}^k\mathbf{J}^+)$. Since the new mean shape matrix $\overline{\mathbf{X}}_k^{nor}$ does not satisfy the scale constraint of the modified generalised Procrustes analysis of the PND [18], we correct $\mathbf{T}^k$ by a scale factor $s_c^k = 1/\|\overline{\mathbf{X}}_k^{nor}\|_F$ and use $\mathbf{T}_c^k = s_c^k\mathbf{T}^k$ instead. Using the transformation matrix $\mathbf{T}_c^k$, we obtain the normalized mean shape $\overline{\mathbf{X}}_k^{nor} = \overline{\mathbf{X}}_k^{train}(\mathbf{J}\mathbf{T}_c^k\mathbf{J}^+)$ and the normalized covariance matrix for non-rigid variations $\mathbf{\Sigma}_{R_k}^{nor} = \mathbf{Q}_k^{norT}\left((\mathbf{J}\mathbf{T}_c^k\mathbf{J}^+)^T \otimes \mathbf{I}\right)\mathbf{\Sigma}_k^{train}\left((\mathbf{J}\mathbf{T}_c^k\mathbf{J}^+) \otimes \mathbf{I}\right)\mathbf{Q}_k^{nor}$, where

$\mathbf{\Sigma}_k^{train} = \mathbf{Q}_k^{train}\mathbf{\Sigma}_{R_k}^{train}\mathbf{Q}_k^{train^T}$ and $\mathbf{Q}_k^{train}$ is calculated from $\overline{\mathbf{X}}_k^{train}$ and $\mathbf{Q}_k^{nor}$ is calculated from $\overline{\mathbf{X}}_k^{nor}$ according to the definition of the PND [18]. The limb length adjustment is explained in Algorithm 2.

Algorithm 2 Limb Length Adjustment

```
Require:
  1: Part difference matrix $\mathbf{J}$
  2: Transformation matrix $\mathbf{T}$
  3: Model parameters: $\overline{\mathbf{X}}$, $\mathbf{\Sigma}_R$, and $\mathbf{Q}$
Ensure: Adjusted model parameters: $\overline{\mathbf{X}}'$, $\mathbf{\Sigma}'_R$, and $\mathbf{Q}'$
  1: $\overline{\mathbf{X}}' = \overline{\mathbf{X}}(\mathbf{J}\mathbf{T}\mathbf{J}^+)$
  2: $\mathbf{T}_c = s_c\mathbf{T}$, where $s_c = 1/\|\overline{\mathbf{X}}'\|_F$
  3: $\overline{\mathbf{X}}' = \overline{\mathbf{X}}(\mathbf{J}\mathbf{T}_c\mathbf{J}^+)$ and calculate $\mathbf{Q}'$ using [18]
  4: $\mathbf{J}' = (\mathbf{J}\mathbf{T}_c\mathbf{J}^+)^T \otimes \mathbf{I}$
  5: $\mathbf{\Sigma}'_R = \mathbf{Q}'^T\mathbf{J}'\mathbf{Q}\mathbf{\Sigma}_R\mathbf{Q}^T\mathbf{J}'\mathbf{Q}'$
```

4.4.2 Model adaptation

While the model normalization step adjusts limb lengths among PNDMM components, we also need to adjust limb lengths when reconstructing from a novel test image. While a PNDMM has s_k and $\mathbf{R}_k$ for rigid motion and $\mathbf{\Sigma}_{R_k}^{train}$ to handle non-rigid variations, they cannot effectively handle the limb length difference problem. We address this issue using Algorithm 2 as a pre-processing step by adjusting model parameters $\overline{\mathbf{X}}_k^{nor}$, $\mathbf{\Sigma}_{R_k}^{nor}$, and $\mathbf{Q}_k^{nor}$ to a reconstructed 3D shape $\widehat{\mathbf{X}}$ obtained from the first iteration of the EM algorithm described in Section 4.3.2.

4.5 Result Selection

We perform 3D reconstruction for $(2n_c - 1)$ 2D pose candidates and select the best reconstruction result among them using three measures described below.

Score of a reprojected 2D shape (r_S): Since the 3D reconstruction algorithm includes a parameter σ to handle the observation noise, it allows a reprojected 2D shape to differ from an input 2D shape. (Dot lines in the selection of Figure 4.1 are reprojected 2D part locations.) To check whether a reconstructed 3D shape is explained by a 2D part detector, we calculate scores of reprojected 2D shapes obtained from reconstructed 3D shapes using (4.1) and the score is denoted as r_S. A higher score means that the reprojected 2D shape is well explained by the 2D part detector and the 3D model.

Normalized reprojection error (r_R): We use different datasets to train a 2D part detector and a 3D model, moreover, the test dataset can be different from the training datasets. Therefore, there can be a bias caused by differences in landmark locations among datasets. It can cause a large reprojection error, even if the 3D reconstruction is close to the ground truth. To address such problem, we propose a normalized reprojection error as follows:

$$r_R(\mathbf{D}, \mathbf{X}) = \frac{1}{n_p} \sum_i \sqrt{\mathbf{\Gamma}(i)^T \mathbf{P}_{orth}^T \mathbf{\Sigma}_{r_i}^{-1} \mathbf{P}_{orth} \mathbf{\Gamma}(i)}, \tag{4.11}$$

where i is a body part index, $\mathbf{\Gamma}(i) = \mathbf{D}(i) - \mathbf{P}_{orth}\mathbf{X}(i)$, $\mathbf{D}(i)$ is the ith column vector of $\mathbf{D}$, $\mathbf{X}(i)$ is the ith column vector of $\mathbf{X}$, $\mathbf{P}_{orth} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is an orthographic projection matrix. That is, r_R corresponds to the mean of Mahalanobis distances calculated using 2D part reprojections. Since we are interested in reprojection errors only, we align the body part positions in training data to body part positions detected in a test image, *i.e.*, $\mathbf{D}^{train}$ is aligned to $\mathbf{D}^{test}$. The aligned 2D shapes and its 3D reconstruction result are denoted as $\widetilde{\mathbf{D}}^{train}$ and $\widetilde{\mathbf{X}}^{train}$, respectively. Since the error between $\widetilde{\mathbf{D}}^{train}$ and the projection of $\widetilde{\mathbf{X}}^{train}$ corresponds to an error caused by the bias, $(\widetilde{\mathbf{D}}^{train}(i) - \mathbf{P}_{orth}\widetilde{\mathbf{X}}^{train}(i))$ is used to compute the sample covariance matrix $\mathbf{\Sigma}_{r_i}$.

Model transformation error (r_T): Since we use a 3D model which is normalized by the mean limb lengths as discussed in Section 4.4.1 and it is adapted to a 2D shape, we calculate the model transformation error, which is defined by the Mahalanobis distance between the mean transformation matrix $\overline{\mathbf{T}}_c = \frac{1}{K}\sum_k \mathbf{T}_c^k$ obtained from model normalization and a transformation matrix $\mathbf{T}_a$ obtained from the current 3D model adaptation process of Section 4.4.2. Since the transformation matrix is a diagonal matrix, in which each element is a length scale, the Mahalanobis distance between $\mathrm{diag}(\overline{\mathbf{T}}_c)$ and $\mathrm{diag}(\mathbf{T}_a)$ can be calculated, where $\mathrm{diag}(\mathbf{A})$ denotes a column vector consisting of diagonal elements of a matrix $\mathbf{A}$.

$$r_T(\mathbf{T}_a, \overline{\mathbf{T}}_c) = \sqrt{\mathrm{diag}(\mathbf{T}_a - \overline{\mathbf{T}}_c)^T \mathbf{\Sigma}_{\mathbf{T}_c}^{-1} \mathrm{diag}(\mathbf{T}_a - \overline{\mathbf{T}}_c)}, \tag{4.12}$$

where $\mathbf{\Sigma}_{\mathbf{T}_c}$ is the sample covariance of $\mathbf{T}_c^k$ from training samples.

Result selection: We again normalize each error between 0 and 1. The normalized errors are denoted as r'_S, r'_R, and r'_T and the final error can be represented as

$$error = (1 - r'_S) + r'_R + r'_T. \tag{4.13}$$

This normalized sum can be interpreted as error voting and we choose a candidate with the lowest error for the final 3D reconstruction.

If there are significant biases in part locations between training and testing sets, a weighted sum of the 2D shape and projected 3D shape further improve the result:

$$\mathbf{D}^{new} = \eta\mathbf{D} + (1 - \eta)\mathbf{P}_{orth}\widehat{\mathbf{X}}, \tag{4.14}$$

where $0 \leq \eta \leq 1$ is a relative weight for emphasizing observation $\mathbf{D}$ and $\widehat{\mathbf{X}}$ is a 3D reconstruction result. With the new observation $\mathbf{D}^{new}$, we do 3D reconstruction described in Section 4.3 and Section 4.4, again. The observation correction and 3D reconstruction can iteratively proceed. If we select a big value for η, the correction will be slow and the 3D reconstruction result becomes close to the observation.

4.6 Experiments

4.6.1 Implementation details

There are some implementation issues that need to be addressed. One issue is related to the part detection. Since the part detector [6] was trained with twisted 2D shapes to capture appearance of poses consistently, *i.e.*, some left and right legs were swapped in the training set, it gives detections with twisted positions of legs. Since such a detection is not suitable for accurate 3D reconstruction, we detect twisted detections by comparing an angle θ_l between a left shoulder-neck and a left hip-neck with an angle θ_r between a left shoulder-neck and a right hip-neck. If $\theta_l > \theta_r$, we swapped two leg positions.

Another issue is related to 3D reconstruction. Since we use a single image, w_k has a small value, which makes proper PND components removed due to incomplete alignment parameters s and $\mathbf{R}$. If proper PND components are removed at the early stage of the EM iteration, the fitting process cannot give an accurate 3D reconstruction result. To solve this problem we performed the power normalization: $w_k = \frac{\sqrt{\pi_k p(\mathbf{D}|c_k=1,\Phi)}}{\sum_l \sqrt{\pi_l p(\mathbf{D}|c_{il}=1,\Phi)}}$. The weight does not affect PND parameters $\overline{\mathbf{X}}_k^{train}$ and $\mathbf{\Sigma}_{R_k}^{train}$ and the role of weight parameter w_k during the fitting process in Section 4.3.2 is to select a reconstruction result with the highest posterior. Since the power normalization does not change the order of weights among PND components, it does not affect the result. The sigma parameter to handle noisy observations in Section 4.3.2 was initialized 10^{-6}, which was updated using (3.23) when the 2D human poses is not ground truth, *i.e.*, Section 4.6.2 and Section 4.6.3.

4.6.2 Evaluation of the joint 2D and 3D pose estimation

Unlike other approaches [11, 69], we use different datasets to train a 2D part detector and a prior 3D model. The part detector [6] was trained using the PARSE dataset [74]. From N-best extension, we generated five initial candidates ($n_c = 5$) and four additional candidates were generated using recombination. The part detection results were converted from 26 part locations to 14 part locations. The 3D shape prior model was leaned using the CMU Mocap dataset[3] with 14 landmark points to match the number of landmark points between the PARSE and CMU Mocap datasets. We randomly selected five frames from each sequence of all available motion sequences for learning the 3D shape prior model, similar to [12]. The parameter n_l for a PNDMM were set to $2n_R$, where n_R is the dimension of a non-rigid shape space defined by a PND. The number of PND mixtures in Section 4.3.1 was initialized to $K = 120$ and reduced to $K = 88$ after training.

We calculated the sample covariance matrix of (4.11) using the nearest three samples after alignment, since samples far away are less predictive on the current observation. The post iteration was performed with $\eta = 0.9$ until the current reprojected score is less than the previous reprojected score and the minimum and maximum number of post iterations are 10 and 30, respectively. The 3D pose errors were evaluated using the mean error and its standard deviation in mm after the Procrustes alignment as done in [12]. We compared the proposed method to part detection based methods [11, 12, 69]. Following the experiment setup in [11, 12, 69], we evaluated our algorithm on the *walking* and *jogging* actions in the HumanEva dataset [75] with the same sequences used in [12].

Unlike the pose estimated from a single 2D pose candidate, our algorithm select the result that is well explained by both 2D and 3D models and it reduces the effects of inaccuracies in 2D part detection results, thereby improving the performance as shown

[3] http://mocap.cs.cmu.edu/subjects.php

in Table 4.1. In Table 4.1, 'N-best' is the case when N candidates from the N-best algorithm [73] are used and 'Recomb.' indicates the case which uses a combination of n_c best candidates from the N-best algorithm and $n_c - 1$ candidates from the recombination step of the proposed method. In our experiments, $N = 9$ and $n_c = 5$. In this table, we can see that the reconstruction results based on multiple 2D candidates are better than those based on a single detection. Moreover, the Table 4.1 shows that the results including recombined candidates are better than those using N-best algorithm alone.

We compared our results with the reported performance in [12, 11, 69]. Since [12] have evaluated their algorithm on the HumanEva dataset [75] with a model trained the CMU Mocap dataset, we first compared our algorithm with [12] in Table 4.2, which shows that our algorithm performs favorably compared to [12]. Since the results of 'Recomb. w/ MT' in Table 4.1 are obtained without post iterations and the results of 'Ours (CMU)' in Table 4.2 are obtained with post iterations, the difference corresponds to the error reduced by post iterations. Figure 4.3 shows examples of 3D pose estimation results obtained from the proposed method. The figure shows that we can better estimate 3D pose from a single image by considering multiple 2D pose candidates with a good 3D shape model.

Since [11, 69] are trained from the HumanEva dataset and tested on the same HumanEva set, the learned 3D models are biased toward subjects in the HumanEva dataset, resulting in smaller reconstruction errors. To confirm this, we tested the proposed method trained by the HumanEva dataset [75]. As shown in Table 4.3, the reconstruction errors are significantly reduced. Here, our algorithm shows excellent reconstruction results in all cases and performs favorably compared to [11, 69].

Table 4.1: Reconstruction errors (mm) according to multiple candidates (training set: CMU Mocap, test set: HumanEva).

Walking	S1	S2	S3
Single w/ MT †	75.1 (19.4)	87.7 (26.5)	114.6 (36.9)
Single w/o MT †	73.4 (15.4)	83.5 (23.8)	103.2 (29.4)
N-best w/ MT †	66.7 (20.0)	83.5 (23.5)	93.9 (14.9)
N-best w/o MT †	75.3 (22.1)	88.9 (31.6)	100.6 (27.8)
Recomb. w/ MT †	66.0 (16.3)	84.7 (20.7)	87.7 (19.0)
Recomb. w/o MT †	70.3 (17.4)	80.1 (20.3)	86.1 (19.5)

Jogging	S1	S2	S3
Single w/ MT †	95.2 (24.2)	92.2 (25.7)	111.9 (32.9)
Single w/o MT †	101.3 (23.4)	99.8 (25.7)	116.4 (33.4)
N-best w/ MT †	96.8 (27.9)	96.2 (23.3)	104.1 (32.1)
N-best w/o MT †	99.5 (20.7)	97.7 (24.9)	110.0 (30.0)
Recomb. w/ MT †	96.0 (27.0)	94.5 (25.7)	100.8 (29.3)
Recomb. w/o MT †	98.2 (22.7)	89.1 (17.2)	112.5 (29.7)

† Without post iterations.

Table 4.2: Reconstruction errors (mm) on the HumanEva dataset (training set: CMU Mocap).

Walking	S1	S2	S3
Ours (CMU) w/ MT ‡	66.3 (17.0)	80.9 (22.1)	83.9 (17.3)
[12]	75.1 (35.6)	99.8 (32.6)	93.8 (19.3)

Jogging	S1	S2	S3
Ours (CMU) w/ MT ‡	95.2 (26.9)	90.6 (20.3)	96.3 (26.6)
[12]	79.2 (26.4)	89.8 (34.2)	99.4 (35.1)

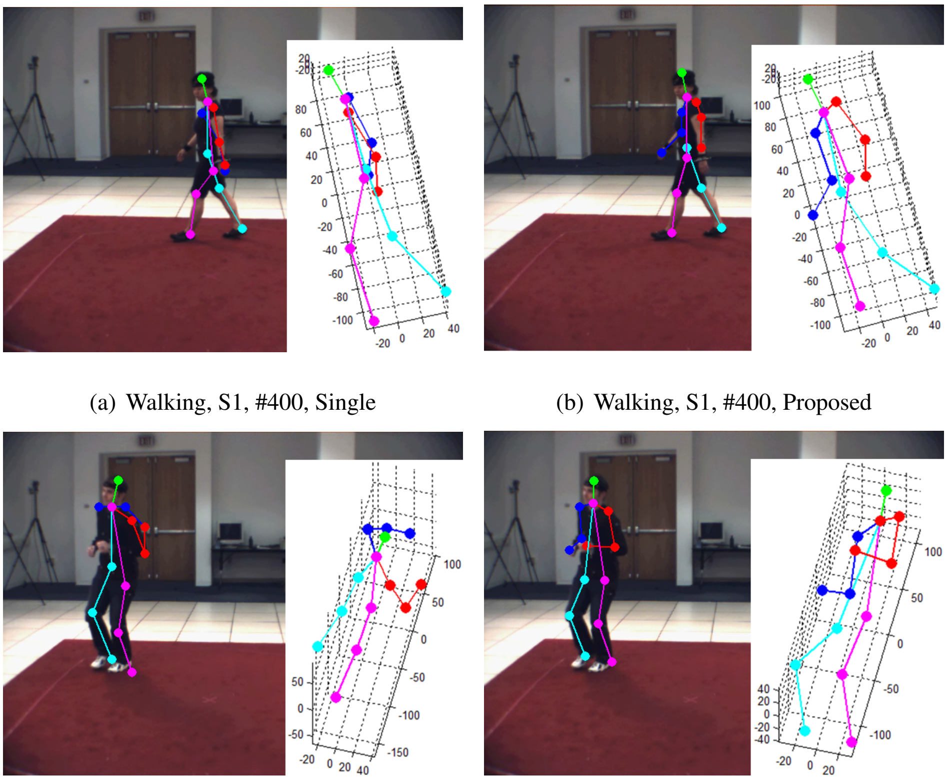
‡ With post iterations.

Table 4.3: Reconstruction errors (mm) on the HumanEva dataset (training set: HumanEva).

Walking	S1	S2	S3
Ours (HumanEva) ‡	49.9 (24.0)	64.5 (30.5)	77.9 (24.7)
[11]	99.6 (42.6)	108.3 (42.3)	127.4 (24.0)
[69]	71.9 (19.0)	75.7 (15.9)	85.3 (10.3)

Jogging	S1	S2	S3
Ours (HumanEva) ‡	66.3 (30.0)	55.8 (25.1)	72.7 (33.6)
[11]	107.2 (41.5)	93.1 (41.1)	115.8 (40.6)
[69]	62.6 (10.2)	77.7 (12.1)	54.4 (9.0)

‡ With post iterations.



(a) Walking, S1, #400, Single

(b) Walking, S1, #400, Proposed

(c) Jogging, S3, #225, Single

(d) Jogging, S3, #225, Proposed

Figure 4.3: Examples of 3D pose estimation results from the Human Eva dataset. (a) and (c) are single detection based results without model transformation and post iteration. (b) and (c) are results from the proposed method using multiple candidates, model transformation, and post iterations. '#' denotes the frame number.

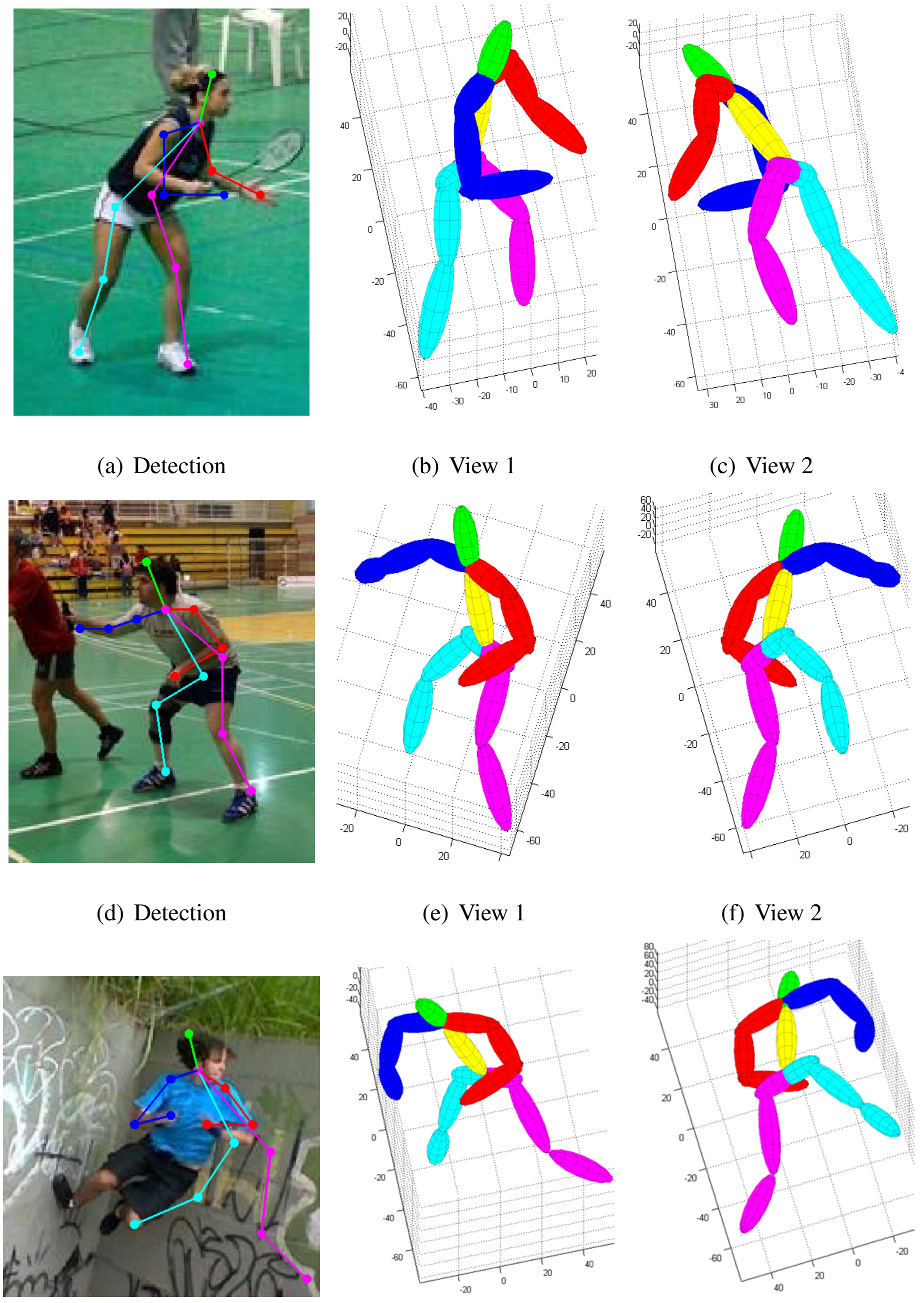
Table 4.4: 2D Part Detection Performance on LSP Dataset [76]

Method	Torso	Head	ULeg	LLeg	UArm	LArm	Total
Yang [6]	82.9	79.1	61.9	53.2	46.0	29.8	54.4
Ours	84.1	79.4	62.7	54.8	45.9	30.7	55.2

4.6.3 Evaluation of the 2D pose estimation

We evaluated the performance of 2D part detections obtained by the proposed 2D candidate selection on 1,000 test samples of the Leed Spart (LSP) dataset [76]. Note that the 2D part detector is trained on the PARSE dataset [74]. The performance measure is the percentage of correct parts (PCP) which is the standard evaluation metric [77]. Table 4.4 shows the performance of 2D part detections selected by the proposed method is better than [6]. We can conclude that the proposed 2D candidate selection improves the 2D part detection performances by considering 3D information. Figure 4.4 shows examples of reconstruction results for qualitative representation [4], where (a)–(f) show two successful cases and (g)–(i) show a failed case. Failed cases are mostly due to incorrect part detections. If we assume that the part detections are correct, the estimated 3D pose is plausible.

[4] To remove the sign ambiguity [50], we select a more plausible result between the reconstructed 3D shape and its depth inverted version and we made one virtual 3D landmark in a lower body for visualization.



(a) Detection (b) View 1 (c) View 2

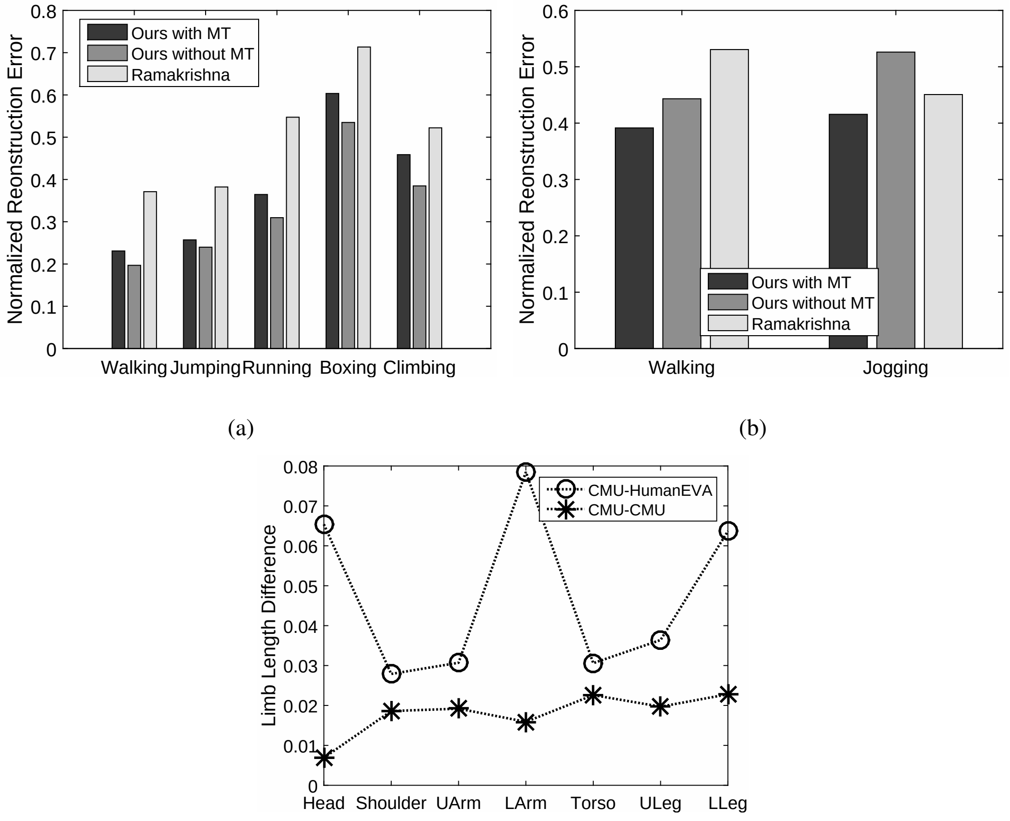
(d) Detection (e) View 1 (f) View 2

(g) Detection (h) View 1 (i) View 2

Figure 4.4: Examples of reconstruction results from the LSP dataset. (a)–(f) are successful cases and (g)–(i) show a failed case. Marker colors correspond to body parts according to the reconstruction result.

4.6.4 Evaluation of the 3D pose estimation

To check 3D reconstruction performances of the proposed method, we have performed additional experiments with known 2D landmark positions. In the CMU Mocap database, we randomly selected a subset of 3D human poses from five different action categories by 23 subjects: $walking$, $jumping$, $running$, $boxing$, and $climbing$ with 14 landmark points. For the generalizability evaluation of the proposed method, we performed 23 rounds of experiments by selecting a subject as testing data and using the remaining subjects for training data. We excluded $climbing$ from training and only use it for testing, since $climbing$ is performed by a subject like in [70]. For testing, we generated 2D landmark positions by orthogonally projecting 3D data into a 2D image plane with a random camera motion and reduced the frame rate to 20 frame per second (fps), since sequences in the CMU Mocap dataset have many redundant frames, *i.e.*, 120 fps. We compared our algorithm with a state-of-the-art algorithm developed by Ramakrishna *et al.*[67], which was retrained on the same training and testing data. For the performance evaluation of the 3D pose estimation, we performed the Procrustean alignment to the estimated 3D pose and ground truth and calculated the Euclidean distance at each landmark point and took the maximum reconstruction error over all the 14 landmark positions and normalized it over the distance between the chest and waist of the ground truth, where the waist landmark point was calculated as a middle value of left and right hip landmark points. We denote this measure as a normalized reconstruction error and Figure 4.5 shows that the proposed method outperforms [67].



(a)

(b)

(c)

Figure 4.5: Normalized reconstruction error. (a) and (b) show the normalized reconstruction errors in CMU Mocap databse and HumanEva dataset, respectively. ‘MT’ denotes model transformation. (c) The differences of mean limb lengths between training and testing data. ‘CMU-CMU’ denotes both training and testing data come from the CMU Mocap database. ‘CMU-HumanEVA’ denotes training and testing data come from the CMU Mocal databse and HumanEVA dataset, respectively. ‘U’ and ‘L’ in x-axis mean ‘Upper’ and ‘Lower’, respectively.

We also tested the 3D model trained by the five action categories of the CMU Mocap dataset to the $walking$ and $jogging$ actions in the HumanEva dataset [75] with the ground truth 2D landmark positions. The model transformation made negative effects on 3D posse estimation in Figure 4.5(a), while Figure 4.5(b) shows that the model transformation significantly decreases 3D pose estimation errors. To investigate what makes two different results, we analyzed differences of mean limb lengths between training and testing data after scale normalization. As can be seen in Figure 4.5(c), differences of limb lengths in the CMU Mocap database are significantly smaller than that between two different datasets, *i.e.*, the CMU Mocap database and HumanEva dataset. While in the same dataset, the limb lengths are varied by different subjects, in two different datasets, the different landmark setting of datasets makes a large amount of bias of limb lengths. We can conclude that the model transformation is useful in cases that there is a large amount of bias between training and testing data and the proposed model transformation method is useful in heterogenous datasets with different joint setting.

4.7 Chapter Summary

We have proposed a method for estimating a 3D human pose from a single novel image. The problem is challenging due to inaccuracies of 2D part detectors and the complexity of human poses. To address these issues, we consider multiple 2D pose candidates with respect to a sound 3D shape model using a Procrustean normal distribution mixture model (PNDMM). We have also introduced model transformation which is incorporated into the 3D shape prior model, such that the proposed method can be applied to a novel test image. Experimental results have shown that the proposed method can provide excellent 3D reconstruction results when tested on a novel test image, despite inaccuracies of 2D part detections and 3D shape ambiguities.

Chapter 5

Application to Action Recognition

Action recognition is an important problem in computer vision, which can be applied to many interesting applications, such as automatic video indexing and retrieval, human-computer interaction, and intelligent surveillance. Since Laptev [13] has introduced space-time interest points by extending the Harris detector, many classical descriptors [79, 72, 80, 81] used in object recognition have been extended from images to videos, *e.g.*, 3D-SIFT [82], HOG3D [83], extended SURF [84], and local trinary patterns [85]. With these local descriptors, bag-of-features (BoF) based methods of object recognition can be directly used for action classification and they have shown to be successful on many datasets [86, 87, 88, 89]. However, since an action in a video occupies in a joint space of 2D spatial domain and 1D time domain unlike an object in a 2D image, descriptors based on 2D spatial domain have many limitations to represent a human action in real world videos [90, 91].

To overcome limitations of 2D appearance based descriptors, many works have tried to enforce motion information using trajectories obtained from point trackers [92, 93, 94].

This chapter is based on the paper appeared in Pattern Recognition: 'Robust Action Recognition Using Local Motion and Group Sparsity [78]'.

Messing *et al.*[92] proposed velocity history features based on a sophisticated latent velocity model and side information, such as appearance, position, and high level semantic information. They have demonstrated the superiority of velocity history features on high resolution video sequences of complicated activities. Sun *et al.*[93] proposed an approach which hierarchically models the spatio-temporal context information about trajectories obtained by matching SIFT descriptors between consecutive frames and showed impressive results on realistic action and event recognition. Wang *et al.*[94] proposed a dense trajectory-based approach by combining point tracking and dense interest point sampling and achieved state-of-the-art results for action recognition compared to sparse interest point sampling techniques, such as the Kanade-Lucas-Tomasi (KLT) tracker [95].

Despite promising results on action recognition, low- and mid- level descriptors have still limited discriminative power in handling large and complex data. Recently, Jhuang *et al.*[4] have systematically analyzed a recognition algorithm to better understand the limitations and found that descriptors based on human poses estimated from [6], even without the ground truth pose, outperform low- and mid-level descriptors for action recognition where the full body is visible. Based on these findings, we extend pose descriptors from 2D spatial domain to 3D spatial domain as shown in Figure 5.1. Since real world consists of 3D objects, 2D observations including 2D human poses are perspectives of 3D objects and amount of information obtained from 3D observations is more than 2D observations. Usage of 3D human pose might give better features than 2D human poses in action recognition systems.

In addition, there is another issue to be solved, *i.e.*, the large variability in actions. When different subjects are performing the same action, they do not have the same appearance and their movements can be quite different for the same action. Even for a person performing the same action multiple times, each performance can be quite dif-

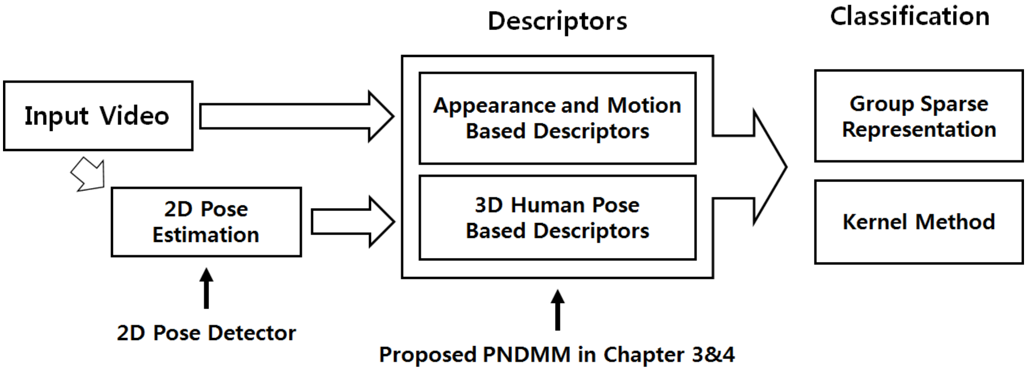

Descriptors
Classification
Input Video
Appearance and Motion Based Descriptors
3D Human Pose Based Descriptors
Group Sparse Representation
Kernel Method
2D Pose Estimation
2D Pose Detector
Proposed PNDMM in Chapter 3&4

Figure 5.1: Overview.

ferent from the previous one. Therefore, robust classification is an important issue in the human action recognition problem and it is necessary to develop a more robust alternative.

Recently, the concept of sparse representation has received significant attention and demonstrated promising performance in signal processing and computer vision [96, 97, 98]. It has been discovered in neuroscience [99] that the human vision system seeks a sparse representation of an incoming image using an overcomplete dictionary. In addition, recent studies go beyond sparsity and take into account additional information about the underlying structure of solutions [97]. Namely, the solution has a natural grouping of its components and the use of this group sparsity can reduce degree of freedom in a solution, thereby leading to a better solution [100]. In [97], a group sparsity method has been successfully applied to object recognition by kernelizing the accelerated proximal gradient (APG) method [101]. As shown in Figure 5.1, we classify action classes using the group sparse representation with the multiple kernel method, instead of a support vector machine (SVM), a popular classifier which is widely used in many action recognition algorithms. Our experimental results show that the proposed action recognition method with 3D pose based descriptors and group sparsity outperforms the baseline method using motion descriptors or 2D pose based descriptors with an SVM classifier [94].

5.1 Appearance and Motion Based Descriptors

We adopt the dense trajectory approach by Wang *et al.*[94] to generate motion descriptors and it is briefly introduced in this section. Feature points are sampled in eight spatial scales with a grid spaced by W pixels and each point $P_t = (x_t, y_t)$ at frame t is tracked to the next frame $t + 1$ by median filtering of a dense optical flow field $\omega_t = (u_t, v_t)$.

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t)|_{(\bar{x}_t, \bar{y}_t)}, \tag{5.1}$$

where M is the median filtering kernel whose size is $N_\omega \times N_\omega$ pixels and $(\bar{x}_t, \bar{y}_t)$ is the rounded position of P_t. Points of subsequent frames are concatenated to form a trajectory $\mathcal{T} = (P_t, P_{t+1}, P_{t+2}, \ldots)$. To extract a dense optical flow, the algorithm by Färneback [102] is adopted.

In the point tracking process, the effects of noise, light conditions, and other factors appear in the form of a drift which is an accumulation of small errors. To avoid this drifting problem, the maximum length of a trajectory is limited to L. Also, trajectories with sudden large or small displacements are removed, since trajectories with small displacements do not contain significant motion information and trajectories with sudden large displacements are most likely to be erroneous. A trajectory is considered to have a small displacement, if the diameter of the smallest region containing the trajectory is less than N_{min} pixels. A trajectory has a large displacement, if the diameter of the smallest region containing the trajectory is larger than N_{max} pixels or the displacement vector between two consecutive frames is larger than a threshold α.

After tracking feature points, the shape of a trajectory, called *TrajShape*, is described by concatenating a set of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} -$

y_t). In order to make a trajectory shape descriptor invariant to scale changes, a concatenated vector is normalized by the overall magnitude of motion displacements:

$$\mathbf{s} = \frac{[\Delta P_t, \cdots, \Delta P_{t+L-1}]}{\sum_{i=t}^{t+L-1} \|\Delta P_i\|}. \tag{5.2}$$

Also, the local motion and appearance in a video volume around a trajectory are described by a histogram of oriented gradients (HOG) [72], a histogram of optical flow (HOF), and a motion boundary histogram (MBH). HOG encodes the local appearance information, while HOF and MBH capture local motion patterns. A 3D video volume, which has the size of $N \times N$ pixels and L frames, is subdivided into $n_\sigma \times n_\sigma \times n_\tau$ cells and each feature is computed at each cell. For HOG, gradient orientations are quantized into eight bins. HOF has nine bins in total, with one extra bin for zero angle. Both descriptors are normalized with their l_2 norm. MBH computes a histogram based on the derivatives of optical flows on both horizontal and vertical components. Like HOG, eight bins are used to quantize orientations and values are normalized using the l_2 norm.

5.2 2D Pose Based Descriptors

Despite the insight that human poses are more high-level cues than weak visual cues for representing human actions, low- to mid-level descriptors have received more attention so far because pose estimation is a difficult problem. However, recent progress in pose estimation makes human pose revisited [103, 104, 105] as a descriptor for action recognition. We, here, introduce 2D pose descriptors used in [4], since we extended these 2D pose descriptors into a 3D pose space.

For action recognition with pose features, Jhuang *et al.*[4] used various types of descriptors derived from joint locations.

- $NTraj$**:** Given the x- and y- coordinates of 15 joints for each frame, they normalized the joint positions with respect to (w.r.t.) the scale obtained from [106] and pose based descriptors are designed as follows: the translation of the normalized joint positions along the x- and y- coordinates (cartesian_trajectory), the direction of the translational vector (radial_trajectory), and the relative positions of normalized joint positions w.r.t a torso joint position (norm_positions). The dimension of descriptors are 30 for translations, 15 for directions, and 30 for positions. Here, the translation is considered as the difference of positions between two adjacent frames along a trajectory. Since trajectories might have jitter caused by imperfect 2D pose estimation, they used differences between frame t and $t + s$, *i.e.*, the feature of type f is a sequence $(f_{t+s} - f_t, \ldots, f_{t+ks} - f_{t+(k\text{-}1)s})$, where $k = \frac{T\text{-}t}{s}$ where a small s is to handle noise. The user parameters T and s are set to 7 and 3, respectively, based on their experiments.

- $NTraj+$: Since relational features describing geometric relations between joints perform better than using normalized joint positions [2], Jhuang *et al.*[4] also extracted a set of relational features: ${}_{15}C_2 = 105$ distances between all the pairs of 15 joints (dist_relations), 105 orientations of the vector connecting two joints (ort_relations), and $3 \times_{15} C_3 = 1365$ inner angles spanned by two vectors connecting all the triples of joints (angle_relations). All possible relational features are computed for each frame, yielding 1,575 descriptor dimensions. In addition to using relational features, they also used the differences of relations between frame t and $t{+}s$ as described in $NTraj$, *i.e.*, dist_relation_trajectory, ort_relation_trajectory, and angle_relation_trajectory.

5.3 Bag-of-Features with a Multiple Kernel Method

We use a Bag-of-Features (BoF) approach which represents a video as an orderless distribution of visual words. We separately create a visual vocabulary for each descriptor type and fix the number of visual words per descriptor to 4,000 for appearance and motion based descriptors and 50 for posed based descriptors. In the case base on appearance and motion based descriptors, we cluster a subset of 100,000 randomly selected training descriptors using k-means to limit the complexity. We perform k-means eight times with random initials and keep the result with the lowest error. Descriptors are assigned to their closest vocabulary word using the Euclidean distance. The resulting frequency histograms of visual and pose word occurrences are used as features for action classification of a video clip. To combine multiple frequency histograms, we use a multiple kernel method from [107]. Each frequency histogram for each descriptor type corresponds to one channel. We compare feature distributions using the exponential χ^2 distance with the multiple kernel method [107] as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{N_c}\sum_{c=1}^{N_c}\frac{1}{\Omega_c}D_c(\mathbf{x}_i, \mathbf{x}_j)\right), \tag{5.3}$$

where $D_c(\mathbf{x}_i, \mathbf{x}_j)$ is χ^2 distance [108] for channel c, and Ω_c is the mean value of χ^2 distances between training samples for the c-th channel.

5.4 Classification - Kernel Group Sparse Representation

A work on image-based face recognition [96] has shown that the sparse representation is naturally discriminative as it selects only a small number of basis vectors that can most compactly represent the given signal. In [96], a single overcomplete dictionary is formed by concatenating vectorized training samples of all classes. Given a test image, its sparsest representation over the dictionary is found by l_1 minimization. The underlying assumption of this method is that a good number of training samples are available per class and they span the sample space well. Guha *et al.*[98] also explored the effectiveness of sparse representation obtained by learning a set of overcomplete dictionaries in the context of action recognition in videos. They proposed three different dictionary training frameworks:

(1) one dictionary for all classes (*shared*),

(2) one dictionary per class (*class-specific*), and

(3) a concatenation of class-specific dictionaries (*concatenated*).

When analyzing their experimental results, we find that the *shared* method shows lower performance than other two methods. It illustrates the fact that the solution has a certain group sparse structure. It is a motivation for the proposed classification method based on group sparsity.

5.4.1 Group sparse representation for classification

Let $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_J] \in \mathbb{R}^{m \times p}$ be a training feature matrix which is generated by concatenating training samples of J classes, *i.e.*, $\mathbf{X}_1 \in \mathbb{R}^{m \times p_1}, \ldots, \mathbf{X}_J \in \mathbb{R}^{m \times p_J}$, where m is the dimensionality of a training sample, p_j is the number of training samples in class j, and $\sum_{j=1}^{J} p_j = p$ is the total number of training samples. Given a test sample $\mathbf{y} \in \mathbb{R}^m$, the classification can be considered as the reconstructing problem of a test sample using training samples, which is represented as

$$\min_{\mathbf{w}_j} \frac{1}{2} \|\mathbf{y} - \mathbf{X}_j \mathbf{w}_j\|_2^2, \quad j = 1, 2, \ldots, J, \tag{5.4}$$

where $\mathbf{w}_j \in \mathbb{R}^{p_j}$ is the coefficient of training samples in the j-th class. The classification label can be assigned by a class with the minimum reconstruction error. However, it is often the case that $\mathbf{X}$ is ill-conditioned in many applications, regularization methods are required for stabilized the solution. The l_1 and $l_{2,1}$ norm have attracted increasing interest and shown promising performance in many application, such as face recognition [96] and object recognition [97].

We extend the group sparse representation approach to action recognition using the multiple kernel method. With group sparse representation, (5.4) can be formulated with $l_{2,1}$ norm regularization term as

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \mu\|\mathbf{w}\|_{2,1}, \tag{5.5}$$

where $\mathbf{w} = [\mathbf{w_1}^T, \ldots, \mathbf{w_J}^T]^T \in \mathbb{R}^p$ and $\|\mathbf{w}\|_{2,1} = \sum_{j=1}^{J} \|\mathbf{w}_j\|_2$ is definition of $l_{2,1}$ norm and μ is a regularization parameter. The coefficient vector $\mathbf{w}_j$ for the j-th class in training samples makes one group in (5.5).

The $l_{2,1}$ norm is indeed a general version of the l_1 norm since if $\mathbf{w}$ has only one group structure, then $\|\mathbf{w}\|_{2,1} = \|\mathbf{w}\|_1$. In addition, $\|\mathbf{w}\|_{2,1}$ is equivalent to $\|\mathbf{d}\|_1$ by constructing a new vector $\mathbf{d} \in \mathbb{R}^J$ with $d_j = \|\mathbf{w}_j\|_2$. Although there exist general optimization algorithms for solving (5.5), such as a subgradient based algorithm, the convergence rate can be quite slow since $\|\mathbf{w}\|_{2,1}$ is non-smooth. Recently, Beck *et al.*[101] proposed an efficient algorithm for solving a nonsmooth convex optimization problem with a guaranteed convergence rate of $O(1/K^2)$, where K is the number of iterations. Following the framework of [101], let us consider $f(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_F^2$ and $g(\mathbf{w}) = \mu\|\mathbf{w}\|_{2,1}$ and apply a proximal regularization of the linearized function of $f(\mathbf{w})$ at a given point $\mathbf{z}$:

$$Q_\eta(\mathbf{w}, \mathbf{z}) := f(\mathbf{z}) + \langle \nabla f(\mathbf{z}), \mathbf{w} - \mathbf{z} \rangle + \frac{\eta}{2}\|\mathbf{w} - \mathbf{z}\|_F^2 + \mu\|\mathbf{w}\|_{2,1}, \tag{5.6}$$

which has a unique minimizer $p_\eta(\mathbf{w}) := \arg\min_{\mathbf{w}}\{Q_\eta(\mathbf{w}, \mathbf{z})\}$. With simple algebra (ignoring constant terms in $\mathbf{z}$), we can obtain

$$p_\eta(\mathbf{w}) = \arg\min_{\mathbf{w}} \left\{ \frac{1}{\eta} g(\mathbf{w}) + \frac{1}{2} \left\| \mathbf{w} - \left(\mathbf{z} - \frac{1}{\eta} \nabla f(\mathbf{z}) \right) \right\|_F^2 \right\}, \tag{5.7}$$

where η is a Lipschitz constant of the gradient $\nabla f(\mathbf{z})$ and plays a role as a step size in optimization. We set η to $2\lambda_{\max}(\mathbf{X}^T\mathbf{X})$ according to [101], where $\lambda_{max}(\mathbf{A})$ is the maximum eigenvalue of $\mathbf{A}$. Finally, by representing $\left(\mathbf{z} - \frac{1}{\eta}\nabla f(\mathbf{z})\right)$ as a vector $\mathbf{r} =$

$[\mathbf{r}_1^T, \mathbf{r}_2^T, \ldots, \mathbf{r}_J^T]^T \in \mathbb{R}^p$ according to the group structure and $\frac{\mu}{\eta}$ as τ, the solution of (5.7) can be obtained as the following [109]:

$$\mathbf{w}_j = \Big[(1 - \tau / \|\mathbf{r}_j\|)\mathbf{r}_j\Big]_+ , \tag{5.8}$$

where $\mathbf{w}_j$ is the coefficient of the j-th group and $[\cdot]_+ = \max(\cdot, 0)$. The optimization for (5.5) is summarized in Algorithm 3.

Algorithm 3 Proximal Gradient Algorithm

```
Require: $\mathbf{X}, \mathbf{w}_0, \eta, \mu > 0$,
Ensure: $\mathbf{w}$
Initialize $\mathbf{z}_0 = \mathbf{r}_0, t_0 = 1, k = 0$.
for $k = 0, 1, 2, \ldots, K - 1$ do
    Calculate $\mathbf{w}_{k+1}$ by (5.7) and (5.8).
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $\mathbf{z}_{k+1} = \mathbf{w}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{w}_{k+1} - \mathbf{w}_k)$
end for
```

5.4.2 Kernel group sparse (KGS) representation for classification

Sparse representation is developed for a feature, while we have multiple features, *i.e.*, appearance, motion, and pose descriptors. Hence, it requires a method to combine multiple features. For the purpose of combining multiple features, we modify the general APG method using the kernel trick as done in [97] for object recognition.

A kernel approach uses a non-linear kernel function $\phi(\cdot)$ to map training and test samples from the original space to a higher dimensional feature space. The kernel trick enable us to operate in the feature space by computing inner products using a kernel function, instead of performing operations in the high-dimensional feature space, *i.e.*, $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ for a given kernel function K. Many algorithms, such as a nonlinear SVM [110] and kernel principal component analysis [111], have used this kernel trick and demonstrated better performance compared to non-kernel methods. In this chapter, we also apply the kernel trick to (5.5) with a kernel function $\phi(\cdot)$. It can be represented as

$$\min_{\mathbf{w}} \frac{1}{2} \left\| \phi(\mathbf{y}) - \sum_{j=1}^{J} \phi(\mathbf{X}_j)\mathbf{w}_j \right\|_2^2 + \mu \sum_{j=1}^{J} \|\mathbf{w}_j\|_2 , \tag{5.9}$$

where $\phi(\mathbf{X}_j) = [\phi(\mathbf{X}_{j,p_1}), \ldots, \phi(\mathbf{X}_{j,p_j})]$. When solving (5.9) using APG, there is a gradient mapping step, *i.e.*, $\nabla f(\cdot) = -\phi(\mathbf{X})^T\phi(\mathbf{y}) + \phi(\mathbf{X})^T\phi(\mathbf{X})$ which involves inner products of features. We can straightforwardly apply the kernel trick here. Let $\mathbf{G} = \phi(\mathbf{X})^T\phi(\mathbf{X})$ with $\phi(\mathbf{X}) = [\phi(\mathbf{X}_1), \cdots, \phi(\mathbf{X}_J)]$ be the training kernel matrix, and $\mathbf{h} = \phi(\mathbf{X})^T\phi(\mathbf{y})$ be the test kernel vector. Then we can have $\nabla f(\cdot) = -\mathbf{h} + \mathbf{G}$ instead of inner products of features.

Using only the optimal coefficients $\widehat{\mathbf{w}}_j$ associated with the j-th class, one can approximate $\mathbf{y}$ of a test sample as $\phi(\mathbf{y}) = \phi(\mathbf{X}_j)\widehat{\mathbf{w}}_j$ and the reconstruction error using training samples in the j-th class is determined as

$$\begin{aligned} E_j = & \quad \|\phi(\mathbf{y}) - \phi(\mathbf{X}_j)\widehat{\mathbf{w}}_j\|_2^2 \\ = & \quad K_{\max} - 2\mathbf{h}_j\mathbf{w}_j + \mathbf{w}_j^T\mathbf{G}_j\mathbf{w}_j, \end{aligned} \tag{5.10}$$

where $K_{\max}$ is the maximum value of the kernel function, $\mathbf{h}_j = \phi(\mathbf{y})^T\phi(\mathbf{X}_j)$ indicates elements of $\mathbf{h}$ associated with the j-th class, and $\mathbf{G}_j = \phi(\mathbf{X}_j)^T\phi(\mathbf{X}_j)$ is the block diagonal of $\mathbf{G}$ associated with the j-th class.

To use the spare representation method for video indexing and retrieval, we need a score function for a positive decision [78]. We define the score function as

$$f(E_j; E_{l\neq j}, \gamma) = \frac{K_{\max} - \min(E_j, \gamma)}{\sum_{l=1}^{J}(K_{\max} - \min(E_l, \gamma))}, \tag{5.11}$$

where $\min(E_j, \gamma)$ is a truncated error function with $0 < \gamma < K_{\max}$ for limiting the maximum error and robust classification. The score function returns a relative score on the j-th class compared to all classes in training samples. That is, the decision score of the j-th class increases if the reconstruction error of the j-th class, E_j, decreases or the

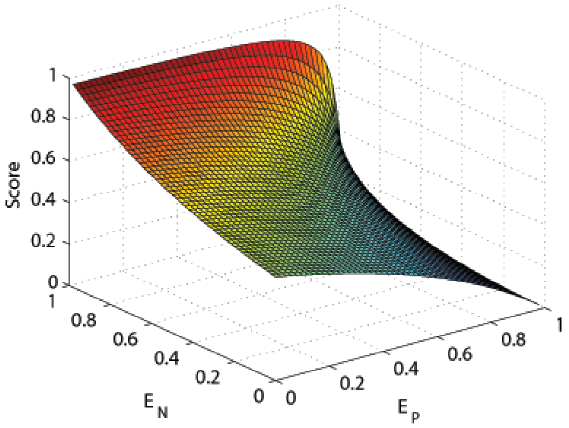


Figure 5.2: An example of a score function (5.11) for a binary classification. A binary classification can be done by thresholding the score. E_P is the reconstruction error using positive training samples and E_N is the reconstruction error using negative training samples.

reconstruction error of the remaining classes, $E_{l \neq j}$, increases, and vice versa. Figure 5.2 shows an example of the score function for the positive class in a binary classification problem. As shown in this figure, the score is proportional to the reconstruction error of a negative class, E_N, and inversely proportional to the reconstruction error of a positive class, E_P, as expected. For a multi-class problem, the class with the highest score is chosen as a solution, *i.e.*, $j^* = \arg\max_j f(E_j; \cdot)$.

5.5 Experiment on sub-JHMDB Dataset

5.5.1 Experimental setup

In this section, we show that 3D pose based descriptors can improve action recognition performances and a group sparse representation with the multiple kernel method outperforms a nonlinear SVM method in most cases. We followed the experiment setup of

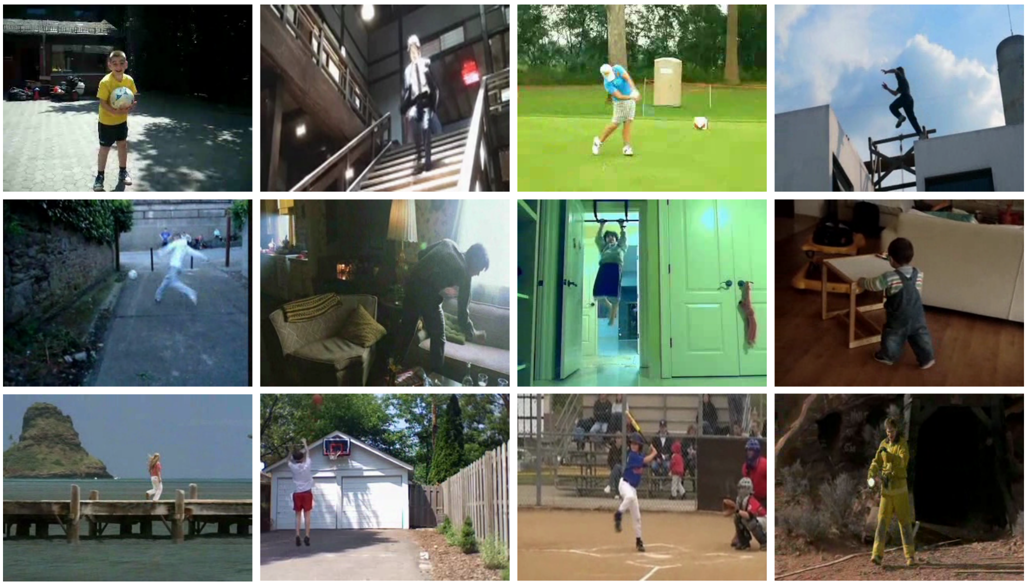


Figure 5.3: Examples of the sub-JHMDB dataset. There are twelve actions: catch, climb stairs, golf, jump, kick ball, pick, pull up, push ,run, shot ball, swing baseball, and walk.

[94] and compared our method to [94] (appearance and motion based descriptors) and [4] (2D pose based descriptors). For experiments, we used default parameters for dense trajectory as $N_\omega = 3$, $N_{min} = 3$, $N_{max} = 50$, $L = 15$, $N = 32$, $n_\sigma = 2$, $n_\tau = 3$, and the threshold α for removing a large displacement between two consecutive frames is 70% of the overall displacement of the trajectory. The sampling step size was set to $W = 5$ and $W = 10$ pixels. The regularization parameter μ in (5.5) was set to 0.001, and γ in (5.10) was set to 0.99.

To evaluate pose based features, we needed videos where the full body is visible and used the sub-JHMDB benchmark dataset. The sub-JHMDB Dataset [4] contains twelve human actions as shown in Figure 5.3: catch, climb stairs, golf, jump, kick ball, pick, pull up, push ,run, shot ball, swing baseball, and walk. The dataset consists of 216 video clips which show a large intra-class variability. We followed the three fold cross validation setting in [4]. For each action classes, video clips were randomly divided into two sets with a constraint that clips from the same video belong to the same set. The grouping was iterated until the ratio of the number of clips in the two sets were both close to 7:3. The 70% set was used for training and the 30% set for testing.

5.5.2 3D pose based descriptor

To generate 3D pose based descriptors, we perform the following procedure which is shown in Figure 5.4.

Obtaining a 2D human pose and its descriptor: We used human pose detector proposed in [3] with the N-best extension [73] and refined poses using dynamic programming. Given human poses obtained from a video clip, we generated nine types of 2D pose descriptors explained in Section 5.2, which are our baseline 2D pose based descriptors.

Obtaining a 3D human pose: Figure 5.4 shows the extension of 2D human pose based descriptors to their 3D version. Since using ground truth 3D human poses for video clips is not available, we estimated 3D human poses using ground truth 2D human poses in a training set before learning a 3D shape model by a PNDMM as shown in the leftmost of Figure 5.4:

(1) For each action we collect a set of ground truth joint locations of 2D human poses and learn a PND using [18], which makes twelve PNDs, *i.e.*, twelve actions.

(2) We generate a PNDMM by combining twelve PNDs in which a mixing probability of PNDMM is assigned to $\frac{1}{N}$ for each component, where N is the number actions.

(3) With estimated 2D human poses, we estimate 3D human poses using the learned PNDMM as done in Chapter 4.

3D human pose based descriptor: After estimating a 3D human pose from each frame, we projected a 3D human pose to x-y, y-z, and z-x image planes, respectively,

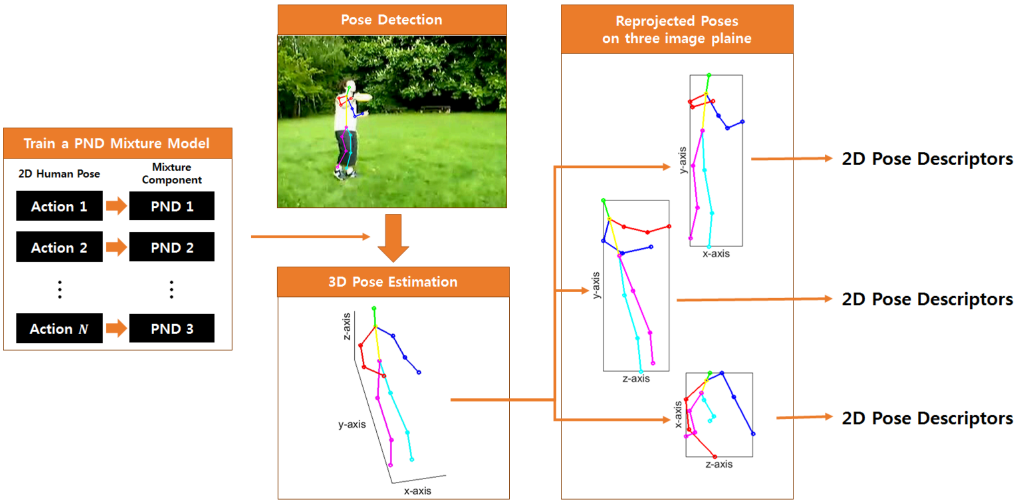


Figure 5.4: A graphical illustration of 3D human pose based descriptor.

which generates three 2D human poses as shown in the rightmost of Figure 5.4. Since a 3D human pose makes three 2D human poses on different image planes, we can obtain three sets of 2D human pose based descriptors from three image planes and they can be used as 3D pose based descriptors. We empirically found projected 2D human poses on y-z and z-x image planes were discontinuous in the temporal direction due to inaccurate 2D and 3D pose estimation. We only used four types of pose descriptors (norm_positions, dist_relations, ort_relations, and angle_relations) on y-z and z-x planes by excluding trajectory based pose descriptors. Therefore, 3D pose based descriptors consist of nine types of 2D pose based descriptors in the x-y image plane and four types of 2D pose based descriptors in the y-z and z-x image planes. All different descriptors were used with multiple kernel method in (5.3) in classification.

5.5.3 Experimental results

As explained in Section 5.2, 2D pose based descriptors used in [4] are the scale normalized version so that all human poses have the same scale. Since knowing the scale factor is difficult in real situations, we tested both of scale unnormalized pose based descriptors and scale normalized pose based descriptors, which are shown in Table 5.1 and Table 5.2. Regardless of the scale normalization, 2D pose based descriptors outperform dense trajectory based descriptors. Also, 3D pose based descriptors outperform 2D pose based descriptors in most cases excepting experimental results for specific split sets. Even when we combined the appearance and motion based descriptors with pose based descriptors, 3D pose based descriptors still outperform 2D pose based descriptors. It shows that estimated 3D poses have more information than 2D poses and they can give positive impacts to action recognition. Looking at the impact according to classification methods, in most cases the KGS is better than an SVM. We also investigated performances according to the sampling step size of dense trajectory [94], i.e, $W = 5$ and $W = 10$. As you can see in Table 5.1 and Table 5.2, performance gaps between 3D pose based descriptors and 2D pose based descriptors increased when we combined with dense trajectory based descriptors at the lower sampling step size. It is also evidence that the amount of information of 3D pose based descriptors is larger than 2D pose based descriptors.

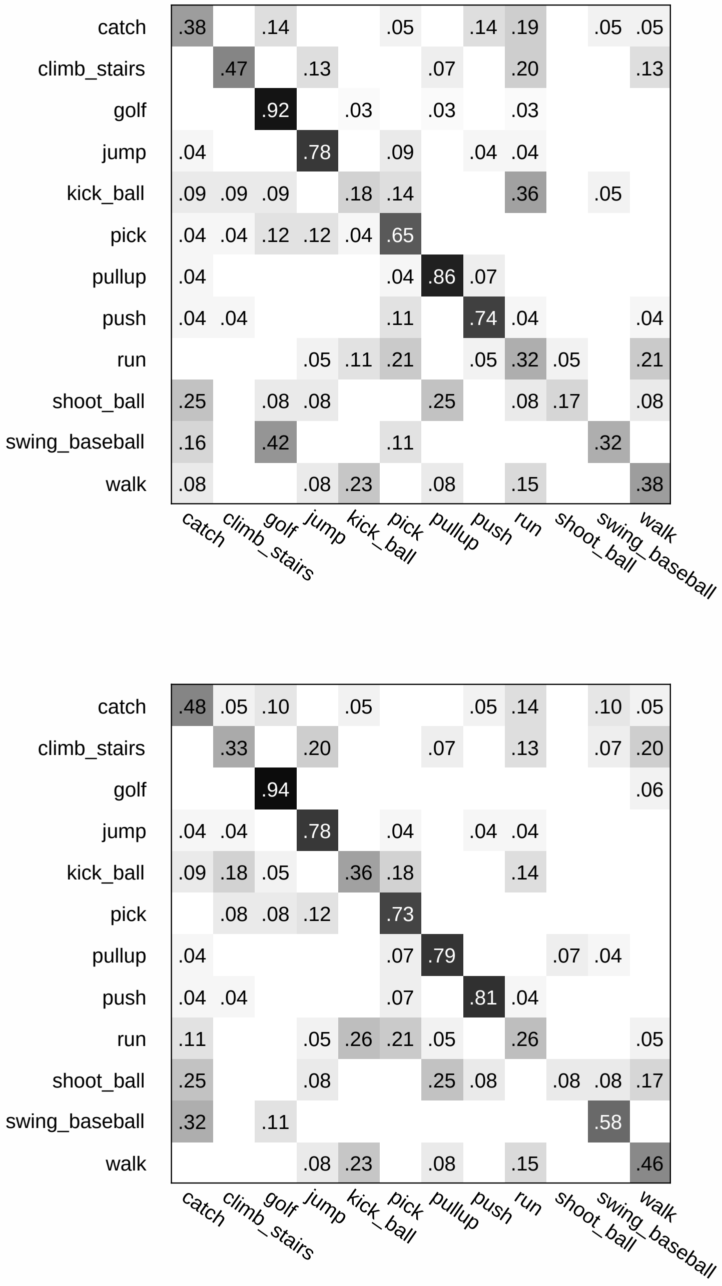
Figure 5.5 shows the confusion matrices on action classes with scale normalized poses. We can find that 3D information can be useful for classifying actions which have quite different shapes from other action in three dimensional space such as ‘catch’ and ‘push’, while the shapes of poses obtained from some action like ‘climb_stairs’ and ‘run’ may be included to other actions and there is a risk that performance decreases by 3D pose based descriptors. With more strong classifier, the risk can be reduced as shown in Figure 5.6, which shows the kernel group spare representation is better than an SVM classifier. We also compared the performance on ground truth 2D poses with their 3D poses estimated from ground truth 2D human poses. As you can see in Table 5.3, despite using 2D ground truth poses for action recognition, estimated 3D poses give still

Table 5.1: Evaluation of the proposed method on the sub-JHMDB dataset with estimated 2D and 3D human poses (w/o scale normalization)

	Descriptor	Split 1	Split 2	Split 3	Total
SVM	Dense Trajectory based descriptors (DT with $W = 10$)	32.6	47.5	45.7	41.8
	Dense Trajectory based descriptors (DT with $W = 5$)	39.3	51.3	48.9	46.4
	2D Pose based descriptors (2D Pose)	49.4	42.5	45.3	49.0
	3D Pose based descriptors (3D Pose)	57.3	50.0	52.2	53.3
	DT with $W = 10$ + 2D Pose based descriptors	59.6	51.3	60.9	57.5
	DT with $W = 10$ + 3D Pose based descriptors	68.5	58.8	58.7	62.1
	DT with $W = 5$ + 2D Pose based descriptors	64.0	52.5	63.0	60.2
	DT with $W = 5$ + 3D Pose based descriptors	70.8	60.0	59.8	63.6
KGS	Dense Trajectory based descriptors (DT with $W = 10$)	41.6	45.0	47.8	44.8
	Dense Trajectory based descriptors (DT with $W = 5$)	46.1	52.5	51.1	49.8
	2D Pose based descriptors (2D Pose)	56.2	45.0	51.1	51.0
	3D Pose based descriptors (3D Pose)	60.7	50.0	51.1	54.0
	DT with $W = 10$ + 2D Pose based descriptors	62.9	56.3	60.9	60.2
	DT with $W = 10$ + 3D Pose based descriptors	67.4	60.0	59.8	62.5
	DT with $W = 5$ + 2D Pose based descriptors	66.3	58.8	62.0	62.5
	DT with $W = 5$ + 3D Pose based descriptors	67.4	61.3	60.9	63.2

Table 5.2: Evaluation of the proposed method on the sub-JHMDB dataset with estimated 2D and 3D human poses (w/ scale normalization)

	Descriptor	Split 1	Split 2	Split 3	Total
SVM	2D Pose based descriptors (2D Pose)	52.8	47.5	50.0	50.2
	3D Pose based descriptors (3D Pose)	61.8	48.8	56.5	55.9
	DT with $W = 10$ + 2D Pose based descriptors	60.7	53.8	55.4	56.7
	DT with $W = 10$ + 3D Pose based descriptors	65.2	53.8	59.8	59.8
	DT with $W = 5$ + 2D Pose based descriptors	62.9	52.5	56.6	57.5
	DT with $W = 5$ + 3D Pose based descriptors	68.5	55.0	60.9	61.7
KGS	2D Pose based descriptors (2D Pose)	53.9	50.0	48.9	51.0
	3D Pose based descriptors (3D Pose)	57.3	48.8	53.3	53.3
	DT with $W = 10$ + 2D Pose based descriptors	65.2	56.3	59.8	60.5
	DT with $W = 10$ + 3D Pose based descriptors	68.5	56.3	59.8	61.7
	DT with $W = 5$ + 2D Pose based descriptors	65.2	56.3	60.9	60.9
	DT with $W = 5$ + 3D Pose based descriptors	69.7	60.0	62.0	65.0

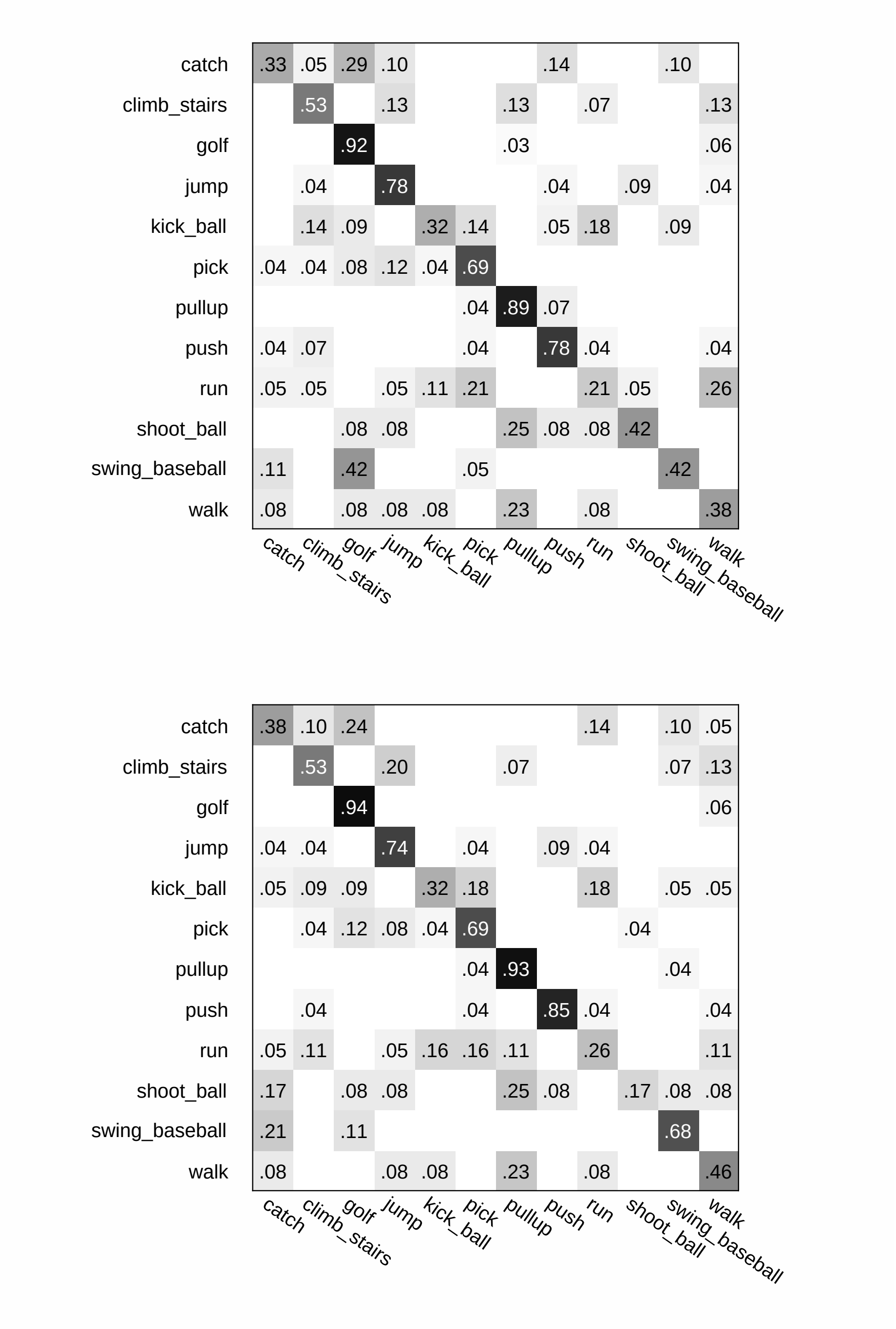


	catch	climb_stairs	golf	jump	kick_ball	pick	pullup	push	run	shoot_ball	swing_baseball	walk
catch	.38		.14			.05		.14	.19		.05	.05
climb_stairs		.47		.13			.07		.20			.13
golf			.92		.03		.03		.03			
jump	.04			.78		.09		.04	.04			
kick_ball	.09	.09	.09		.18	.14			.36		.05	
pick	.04	.04	.12	.12	.04	.65						
pullup	.04					.04	.86	.07				
push	.04	.04				.11		.74	.04			.04
run				.05	.11	.21		.05	.32	.05		.21
shoot_ball	.25		.08	.08			.25		.08	.17		.08
swing_baseball	.16		.42			.11					.32	
walk	.08			.08	.23		.08		.15			.38

	catch	climb_stairs	golf	jump	kick_ball	pick	pullup	push	run	shoot_ball	swing_baseball	walk
catch	.48	.05	.10		.05			.05	.14		.10	.05
climb_stairs		.33		.20			.07		.13		.07	.20
golf			.94									.06
jump	.04	.04		.78		.04		.04	.04			
kick_ball	.09	.18	.05		.36	.18			.14			
pick		.08	.08	.12		.73						
pullup	.04					.07	.79			.07	.04	
push	.04	.04				.07		.81	.04			
run	.11			.05	.26	.21	.05		.26			.05
shoot_ball	.25			.08			.25	.08		.08	.08	.17
swing_baseball	.32		.11								.58	
walk				.08	.23		.08		.15			.46

(b)

Figure 5.5: Confusion matrix obtained from a support vector machine. (a) Confusion matrix with 2D human poses. (b) Confusion matrix with 3D human poses.



	catch	climb_stairs	golf	jump	kick_ball	pick	pullup	push	run	shoot_ball	swing_baseball	walk
catch	.33	.05	.29	.10				.14			.10	
climb_stairs		.53		.13			.13		.07			.13
golf			.92				.03					.06
jump		.04		.78				.04		.09		.04
kick_ball		.14	.09		.32	.14		.05	.18		.09	
pick	.04	.04	.08	.12	.04	.69						
pullup						.04	.89	.07				
push	.04	.07				.04		.78	.04			.04
run	.05	.05		.05	.11	.21			.21	.05		.26
shoot_ball			.08	.08			.25	.08	.08	.42		
swing_baseball	.11		.42			.05					.42	
walk	.08		.08	.08	.08		.23		.08			.38

	catch	climb_stairs	golf	jump	kick_ball	pick	pullup	push	run	shoot_ball	swing_baseball	walk
catch	.38	.10	.24						.14		.10	.05
climb_stairs		.53		.20			.07				.07	.13
golf			.94									.06
jump	.04	.04		.74		.04		.09	.04			
kick_ball	.05	.09	.09		.32	.18			.18		.05	.05
pick		.04	.12	.08	.04	.69				.04		
pullup						.04	.93				.04	
push		.04				.04		.85	.04			.04
run	.05	.11		.05	.16	.16	.11		.26			.11
shoot_ball	.17		.08	.08			.25	.08		.17	.08	.08
swing_baseball	.21		.11								.68	
walk	.08			.08	.08		.23		.08			.46

(b)

Figure 5.6: Confusion matrix obtained from kernel group sparse representation. (a) Confusion matrix with 2D human poses. (b) Confusion matrix with 3D human poses.

Table 5.3: Evaluation of the proposed method on the sub-JHMDB dataset with ground truth 2D human poses and estimated 3D human poses

	Descriptor	Split 1	Split 2	Split 3	Total
SVM	2D Pose based descriptors (2D Pose)	74.2	75.0	76.1	75.1
	3D Pose based descriptors (3D Pose)	77.5	80.0	80.4	79.3
KGS	2D Pose based descriptors (2D Pose)	69.7	76.3	82.6	76.3
	3D Pose based descriptors (3D Pose)	78.7	76.3	78.3	77.8

positive impacts in most cases. Considering the fact that the ground truth 2D poses are very strong descriptors and we use simple 3D pose based descriptors, using estimated 3D poses for action recognition is promising and obtaining an accurate 3D pose from a 2D pose like in this dissertation is a very important research issue for reliable computer vision systems.

5.6 Chapter Summary

In this chapter, we have proposed an action recognition method using 3D human pose based descriptors. To estimate a 3D human pose, we have used the proposed method in Chapter 3 and Chapter 4. After obtaining 3D human poses, we have projected 3D human poses into three image planes, *i.e.*, x-y, y-z, and z-x, and generated 2D pose based descriptors in each image plane. The pose based descriptors using 3D human poses outperform those with 2D human poses, regardless of classification methods. Furthermore, we have used the group sparse representation with the multiple kernel method (KGS) for robust classification. Through extensive experiments, we have demonstrated that, in most cases, KGS can improve the performance of action recognition, especially when the discriminative power of features is low.

Chapter 6

Conclusion and Future Work

Since real world consists of 3D objects, usage of 3D information might give better solutions in many computer vision problems including action recognition systems using 3D human poses. In this dissertation, we have proposed a new method to reconstruct 3D shapes form 2D shapes, *i.e.*non-rigid structure from motion (NRSfM) problem.

Based on a Procrustean normal distribution (PND) which was recently proposed [18] and showed state-of-the-art on bench mark datasets, we have proposed a new mixture model for representing 3D shape variations, which is called Procrustean normal distribution mixture model (PNDMM). While all most existing methods for NRSfM used a single model, the proposed PNDMM is able to decompose a complex shape variations to a set of simpler ones, which enables the model learning to be more tractable and accurate. Given 2D observations of a non-rigid object, model learning is performed using the expectation-maximization (EM) algorithm and component-wise EM algorithm. Experimental results with various conditions using various datasets have shown that the PNDMM and *adaptive* PNDMM significantly outperform existing methods. The proposed PNDMM are not limited to the case of a human pose, but it can be easily extended to more general objects.

In addition, we have extended the proposed PNDMM to single view 3D human pose estimation. The problem is challenging due to inaccuracies of 2D part detectors and inherent ambiguity of 3D reconstruction from a single 2D observation. Moreover, the human poses are very complex shapes. In order to address inaccurate of 2D pose estimation on a single image, we have generated multiple 2D human pose candidates and reconstructed 3D human poses by using a sound 3D shape model, a PNDMM, learned by a CMU 3D motion capture dataset. After that we have selected the best one which can be explained by a 2D human pose detector and a 3D shape model. We have also introduced model transformation which is incorporated into the 3D shape prior model, such that the proposed method can be applied to a novel test image. Experimental results have shown that the proposed method can provide excellent 3D reconstruction results when tested on a novel test image, despite inaccuracies of 2D part detections and 3D shape ambiguities.

Finally, we have applied the proposed methods to action recognition from a video clip. Despite that using human poses has a lot of advantages, posed-based action recognition has not received attention over past few decades, which caused from the difficulty of human pose estimation on an image. However, current great progress in human pose estimation makes it possible to robustly estimate a human pose in images. Furthermore, current studies have pointed to high-level features obtained from 2D estimated human poses enable action recognition performance beyond current state-of-the-art methods using low- and mid-level features based on appearance and motion, despite inaccuracy of human pose estimation. Since the proposed PNDMM is able to reconstruct 3D shapes from 2D shapes, we have proposed an action recognition method based on 3D human pose based descriptors. Experimental results have shown that 3D pose based descriptors are better than 2D pose based descriptors for action recognition, regardless of classification methods. Considering the fact that we used simple 3D pose based descriptors based on a 3D shape model learned from 2D shapes, results in this dissertation are promising and obtaining accurate 3D information from 2D observations is a very important research issue for reliable computer vision systems.

The PNDMM assume that 2D observations are obtained from a orthographic camera model and in many cases it works well. However, a lot of cameras in real world are based on a perspective camera model. For future work, NRSfM based on a perspective camera model should be studied, which is more challenging. Moreover, a new mixture model using temporal information will be also studied, since a video stream is a kind of time series data.

Appendices

Appendix A

Proof of Propositions in Chapter 2

A.1 Proof of Proposition 1

To explain how to estimate the distribution of $\mathbf{h}_i$ in E-step, we use (2.3) and (2.7). From Bayes' theorem, the distribution of $\mathbf{h}_i$ can be written as

$$\begin{aligned} & p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}) \\ &= \frac{p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi})}{\int p(\mathbf{D}_i, \mathbf{h}_i|\mathbf{\Phi}) d\mathbf{h}_i} \\ &= \frac{\frac{1}{(2\pi)^{\frac{3(n_p-1)}{2}} |\mathbf{\Sigma}_R|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{v}_i^T \mathbf{P} {\mathbf{\Sigma}_R}^{-1} \mathbf{P}^T \mathbf{v}_i\right)}{\int \frac{1}{(2\pi)^{\frac{3(n_p-1)}{2}} |\mathbf{\Sigma}_R|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\mathbf{v}_i^T \mathbf{P} {\mathbf{\Sigma}_R}^{-1} \mathbf{P}^T \mathbf{v}_i\right) d\mathbf{h}_i}. \end{aligned} \tag{A.1}$$

Since $\mathbf{h}_i$ is a Gaussian distribution, it can be represented as

$$\begin{aligned} p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi}) &= \frac{1}{(2\pi)^{\frac{n_p-1}{2}} |\mathbf{C}_i'|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\tilde{\mathbf{h}}_i \mathbf{P}_h {\mathbf{C}_i'}^{-1} \mathbf{P}_h^T \tilde{\mathbf{h}}_i^T\right) \\ &= \frac{\frac{1}{(2\pi)^{\frac{n_p-1}{2}} |\mathbf{C}_i'|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\tilde{\mathbf{h}}_i \mathbf{P}_h {\mathbf{C}_i'}^{-1} \mathbf{P}_h^T \tilde{\mathbf{h}}_i^T + t_i\right)}{\int \frac{1}{(2\pi)^{\frac{n_p-1}{2}} |\mathbf{C}_i'|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\tilde{\mathbf{h}}_i \mathbf{P}_h {\mathbf{C}_i'}^{-1} \mathbf{P}_h^T \tilde{\mathbf{h}}_i^T + t_i\right) d\mathbf{h}_i}. \end{aligned} \tag{A.2}$$

An arbitrary constant t_i does not have any effect on $p(\mathbf{h}_i|\mathbf{D}_i, \mathbf{\Phi})$, because it is both on the numerator and denominator of the above equation. From (A.1) and (A.2), we can write

$$\begin{aligned}
&\mathbf{v}_i^T \mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\mathbf{v}_i \\
&= \mathbf{vec}\left(s_i\mathbf{R}_{i3}\mathbf{h}_i\right)^T \mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\mathbf{vec}\left(s_i\mathbf{R}_{i3}\mathbf{h}_i\right) \\
&+ 2\mathbf{vec}\left(s_i\left[\mathbf{R}_{i1}\ \mathbf{R}_{i2}\right]\mathbf{D}_i - \overline{\mathbf{X}}\right)^T \mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\mathbf{vec}\left(s_i\mathbf{R}_{i3}\mathbf{h}_i\right) \\
&+ \mathbf{vec}\left(s_i\left[\mathbf{R}_{i1}\ \mathbf{R}_{i2}\right]\mathbf{D}_i - \overline{\mathbf{X}}\right)^T \mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\mathbf{vec}\left(s_i\left[\mathbf{R}_{i1}\ \mathbf{R}_{i2}\right]\mathbf{D}_i - \overline{\mathbf{X}}\right).
\end{aligned} \tag{A.3}$$

and

$$\begin{aligned}
&\tilde{\mathbf{h}}_i\mathbf{P}_h{\mathbf{C}_i'}^{-1}\mathbf{P}_h^T\tilde{\mathbf{h}}_i^T + t_i \\
&= \mathbf{h}_i\mathbf{P}_h{\mathbf{C}_i'}^{-1}\mathbf{P}_h^T\mathbf{h}_i^T - 2\bar{\mathbf{h}}_i\mathbf{P}_h{\mathbf{C}_i'}^{-1}\mathbf{P}_h^T\mathbf{h}_i^T + \bar{\mathbf{h}}_i\mathbf{P}_h{\mathbf{C}_i'}^{-1}\mathbf{P}_h^T\bar{\mathbf{h}}_i^T + t_i.
\end{aligned} \tag{A.4}$$

Comparing the first term in (A.3) and (A.4) and using the relation $\mathbf{vec}(\mathbf{R}_{i3}\mathbf{h}_i) = (\mathbf{I}_{3n_p} \otimes \mathbf{R}_{i3})\mathbf{h}_i^T$ and $(\mathbf{I}_{3n_p} \otimes \mathbf{R}_{i3})\mathbf{P}_h = (\mathbf{P}_h \otimes \mathbf{R}_{i3}) = \boldsymbol{\Psi}_i$, we can express $\mathbf{C}_i'$ as

$$\mathbf{C}_i' = \frac{1}{s_i^2}\left(\boldsymbol{\Psi}_i^T\mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\boldsymbol{\Psi}_i\right)^{-1}.$$

Also, comparing the second term in (A.3) and (A.4), we can express $\bar{\mathbf{h}}_i$ as

$$\bar{\mathbf{h}}_i = s_i\mathbf{vec}\left(\overline{\mathbf{X}} - s_i\left[\mathbf{R}_{i1}\ \mathbf{R}_{i2}\right]\mathbf{D}_i\right)^T \mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\boldsymbol{\Psi}_i\mathbf{C}_i'\mathbf{P}_h^T.$$

A.2 Proof of Proposition 3

To find the optimal s_i^*, we rewrite the objective function (2.9) as

$$\begin{aligned}
J = &-\sum_i s_i^2 \mathrm{tr}\left(\boldsymbol{\Psi}_i^T\mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\boldsymbol{\Psi}_i\mathbf{C}_i'\right) - \sum_i \bar{\mathbf{v}}_i^T\mathbf{P}{\boldsymbol{\Sigma}_R}^{-1}\mathbf{P}^T\bar{\mathbf{v}}_i \\
&+ \lambda\left(\sum_i s_i^2\left(\left\|\begin{bmatrix}\mathbf{D}_i \\ \bar{\mathbf{h}}_i\end{bmatrix}\right\|_F^2 + \mathrm{tr}\left(\mathbf{C}_i'\right)\right) - 1\right).
\end{aligned} \tag{A.5}$$

By substituting (2.13) to (A.5), it becomes a generalized eigenvalue problem and $\mathbf{s}^*$ corresponds to the eigenvector of the smallest eigenvalue in the following.

$$\mathbf{s}^* = \arg\min_{\mathbf{s}} \mathbf{s}^T\mathbf{G}\mathbf{s} \quad \text{s.t.} \quad \mathbf{s}^T\mathbf{F}\mathbf{s} = 1, \tag{A.6}$$

where $\mathbf{G}$, $\boldsymbol{\Psi}_i$, $\mathbf{q}_i$, and $\mathbf{F}$ are given in (2.12). (A.6) gives (2.11).

A.3 Proof of Proposition 4

To ensure that $\mathbf{\Sigma}_R = \alpha\mathbf{Z} + (1-\alpha){\mathbf{\Sigma}_R}^{old}$ does increase the log-likelihood, we have to verify that the Frobenius inner product of $\left(\frac{\partial J}{\partial \mathbf{\Sigma}_R}\Big|_{\mathbf{\Sigma}_R=\alpha\mathbf{Z}+(1-\alpha){\mathbf{\Sigma}_R}^{old}}\right)$ and $\left(\mathbf{Z} - {\mathbf{\Sigma}_R}^{old}\right)$ is positive for all $0 \leq \alpha \leq 1$ (the inner product of the gradient and the direction to the optimal point from the current ${\mathbf{\Sigma}_R}^{old}$ should be positive.). Note that

$$\begin{aligned}&\mathrm{tr}\left(\left(\frac{\partial J}{\partial \mathbf{\Sigma}_R}\Big|_{\mathbf{\Sigma}_R=\alpha\mathbf{Z}+(1-\alpha){\mathbf{\Sigma}_R}^{old}}\right)^T\left(\mathbf{Z} - {\mathbf{\Sigma}_R}^{old}\right)\right)\\ &= \mathrm{tr}((\mathbf{Z} - {\mathbf{\Sigma}_R}^{old})^T(-n_s(\alpha\mathbf{Z} + (1-\alpha){\mathbf{\Sigma}_R}^{old})^{-1}\\ &\quad + n_s(\alpha\mathbf{Z} + (1-\alpha){\mathbf{\Sigma}_R}^{old})^{-1}\mathbf{Z}(\alpha\mathbf{Z} + (1-\alpha){\mathbf{\Sigma}_R}^{old})^{-1})).\end{aligned} \tag{A.7}$$

Let $\mathbf{W}$ be the generalized eigenvector matrix of $\mathbf{Z}$ and ${\mathbf{\Sigma}_R}^{old}$ such that

$$\mathbf{W}^T{\mathbf{\Sigma}_R}^{old}\mathbf{W} = \mathbf{I}, \qquad \mathbf{W}^T\mathbf{Z}\mathbf{W} = \Lambda. \tag{A.8}$$

Substituting (A.8) into (A.7), we can obtain

$$n_s(1-\alpha)\sum_i \frac{(\lambda_i - 1)^2}{(\alpha(\lambda_i - 1) + 1)^2},$$

where λ_i is the ith diagonal matrix of Λ. Since $n_s(1-\alpha) \geq 0$ and $\sum_i \frac{(\lambda_i-1)^2}{(\alpha(\lambda_i-1)+1)^2} \geq 0$ for $0 \leq \alpha \leq 1$, (A.7) is always positive for all $0 \leq \alpha \leq 1$.

Appendix B

Calculation of $p(\mathrm{X}_i|\mathrm{D}_i, \Phi_i)$ in Chapter 3

This appendix section describes the effect of ignoring the Dirac-delta term in (3.16).

B.1 Without the Dirac-delta term

We omit subscripts i, k, and ik if no confusion arises. Using Bayes' rule, the posterior distribution of $\mathbf{X}$ can be written as

$$\begin{aligned}
&p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi}) \propto p(\mathbf{D}|\mathbf{X}, \sigma)p(\mathbf{X}|\mathbf{\Phi}) \\
&\propto \exp\Bigg(-\frac{1}{2}\mathbf{vec}(\mathbf{X})^T\mathbf{H}\mathbf{vec}(\mathbf{X}) + \frac{1}{\sigma^2}\mathbf{vec}(\mathbf{D})^T\mathbf{F}^T\mathbf{vec}(\mathbf{X}) \\
&+ s\mathbf{vec}(\overline{\mathbf{X}})^T\mathbf{Q}\Sigma_R^{-1}\mathbf{Q}^T\left(\mathbf{I}\otimes\mathbf{R}\right)\mathbf{vec}(\mathbf{X})\Bigg) \\
&= \exp\left(-\frac{1}{2}\mathbf{vec}(\mathbf{X})^T\mathbf{H}\mathbf{vec}(\mathbf{X}) + \frac{1}{\sigma^2}\mathbf{vec}(\mathbf{D})^T\mathbf{vec}(\mathbf{X})\right),
\end{aligned} \tag{B.1}$$

where $\mathbf{H} = s^2(\mathbf{I}\otimes\mathbf{R}^T)\mathbf{\Sigma}^+(\mathbf{I}\otimes\mathbf{R}) + \frac{1}{\sigma^2}\mathbf{F}$ and we use $\mathbf{vec}(\overline{\mathbf{X}})^T\mathbf{Q} = 0$, $\mathbf{vec}(\mathbf{D}) = \mathbf{F}\mathbf{vec}(\mathbf{D})$, and $\mathbf{F}^2 = \mathbf{F}$.

We can also write $p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi})$ as:

$$\begin{aligned} &p(\mathbf{vec}(\mathbf{X})|\mathbf{D}, \mathbf{\Phi}) = \mathcal{N}\left(\mathbf{m}, \mathbf{\Omega}\right) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{vec}(\mathbf{X}) - \mathbf{m}\right)^T \mathbf{\Omega}^{-1}\left(\mathbf{vec}(\mathbf{X}) - \mathbf{m}\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\mathbf{vec}(\mathbf{X})^T\mathbf{\Omega}^{-1}\mathbf{vec}(\mathbf{X}) + \mathbf{m}^T\mathbf{\Omega}^{-1}\mathbf{vec}(\mathbf{X})\right). \end{aligned} \tag{B.2}$$

Comparing (B.2) with (B.1), we have

$$\begin{aligned} \mathbf{\Omega}^{-1} &= \mathbf{H}, \\ \mathbf{m}^T\mathbf{\Omega}^{-1} &= \frac{1}{\sigma^2}\mathbf{vec}(\mathbf{D})^T, \end{aligned} \tag{B.3}$$

Therefore, we can represent $p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi})$ as the following Gaussian distribution:

$$\begin{aligned} &p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi}) \\ &= p(\mathbf{vec}(\mathbf{X})|\mathbf{D}, \mathbf{\Phi}) \sim \mathcal{N}\left(\mathbf{m}, \mathbf{\Omega}\right), \\ &\mathbf{m} = \frac{1}{\sigma^2}\mathbf{\Omega}\mathbf{vec}(\mathbf{D}) \\ &= \frac{1}{\sigma^2}\left(s^2(\mathbf{I}\otimes\mathbf{R}^T)\mathbf{\Sigma}^+(\mathbf{I}\otimes\mathbf{R}) + \frac{1}{\sigma^2}\mathbf{F}\right)^+ \mathbf{vec}(\mathbf{D}), \\ &\mathbf{\Omega} = \left(s^2(\mathbf{I}\otimes\mathbf{R}^T)\mathbf{\Sigma}^+(\mathbf{I}\otimes\mathbf{R}) + \frac{1}{\sigma^2}\mathbf{F}\right)^+. \end{aligned} \tag{B.4}$$

B.2 With the Dirac-delta term

Let $\mathbf{v}$ be a random vector drawn from $\mathcal{N}(\mathbf{0}, \Sigma_R)$. Then, $\mathbf{vec}(\mathbf{X})$ can be represented as follows:

$$\mathbf{vec}(\mathbf{X}) = \frac{1}{s}\left(\mathbf{I}\otimes\mathbf{R}^T\right)\left(\mathbf{Q}\mathbf{v} + \mathbf{vec}(\overline{\mathbf{X}})\right). \tag{B.5}$$

By substituting (B.5) to $\mathbf{vec}(\mathbf{X})$ of (B.2) in Section B.1 and rearranging them with respect to $\mathbf{v}$, it can be written as

$$\begin{aligned} &p(\mathbf{v}|\mathbf{D}, \mathbf{\Phi}) \\ &\propto \exp\Bigg(-\frac{1}{2}\mathbf{v}^T\big(\mathbf{\Sigma}_R^{-1}\frac{1}{s^2\sigma^2}\mathbf{Q}^T\left(\mathbf{I}\otimes\mathbf{R}\right)\mathbf{F}\left(\mathbf{I}\otimes\mathbf{R}^T\right)\big)\mathbf{v} \\ &+ \frac{1}{s\sigma^2}\left(s\mathbf{vec}(\mathbf{D})^T\left(\mathbf{I}\otimes\mathbf{R}^T\right)\mathbf{Q} - \overline{\mathbf{X}}^T\left(\mathbf{I}\otimes\mathbf{R}\right)\mathbf{Q}\right)\mathbf{v}\Bigg). \end{aligned} \tag{B.6}$$

Since (B.6) has only a quadric term and a linear term of $\mathbf{v}$, and a constant term, it can be represented as

$$\begin{aligned} p(\mathbf{v}|\mathbf{D}, \mathbf{\Phi}) &= \mathcal{N}(\widehat{\mathbf{m}}, \widehat{\mathbf{\Omega}}) \\ &\propto \exp\left(-\frac{1}{2}(\mathbf{v}-\widehat{\mathbf{m}})^T\widehat{\mathbf{\Omega}}^{-1}(\mathbf{v}-\widehat{\mathbf{m}})\right) \\ &\propto \exp\left(-\frac{1}{2}\mathbf{v}^T\widehat{\mathbf{\Omega}}^{-1}\mathbf{v}+\widehat{\mathbf{m}}^T\widehat{\mathbf{\Omega}}^{-1}\mathbf{v}\right). \end{aligned} \tag{B.7}$$

Comparing (B.6) with (B.7), we can write

$$\begin{aligned} &p(\mathbf{v}|\mathbf{D}, \mathbf{\Phi}) = \mathcal{N}(\widehat{\mathbf{m}}, \widehat{\mathbf{\Omega}}), \\ &\widehat{\mathbf{m}} = \frac{1}{s\sigma^2}\widehat{\mathbf{\Omega}}\mathbf{Q}^T\big(\mathbf{vec}(\mathbf{D}) - \frac{1}{s}\mathbf{F}\left(\mathbf{I}\otimes\mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}})\big), \\ &\widehat{\mathbf{\Omega}} = s^2\left(s^2\mathbf{\Sigma}_R^{-1} + \mathbf{Q}^T\left(\mathbf{I}\otimes\mathbf{R}\right)\frac{\mathbf{F}}{\sigma^2}\left(\mathbf{I}\otimes\mathbf{R}^T\right)\mathbf{Q}\right)^+, \end{aligned} \tag{B.8}$$

where $\big(\mathbf{vec}(\mathbf{D}) - \frac{1}{s}\mathbf{F}\left(\mathbf{I}\otimes\mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}})\big)$ corresponds to non-rigid variations and we can see only non-rigid variations affect $\widehat{\mathbf{m}}$ since $\mathbf{Q}$ is orthogonal to rigid variations by the definition of PND.

$\mathbf{X}$ can be consider a linear transformed and translated version of $\mathbf{v}$ as shown in (B.5). By a linear property of a Gaussian distribution, we can also represent $p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi})$ as a

Gaussian distribution as

$$\begin{aligned}
& p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi}) = p(\mathbf{vec}(\mathbf{X})|\mathbf{D}, \mathbf{\Phi}) \sim \mathcal{N}(\mathbf{m}, \mathbf{\Omega}), \quad \text{where} \\
\mathbf{m} &= \mathbf{vec}(E[\mathbf{X}]) = \frac{1}{s}\left(\mathbf{I} \otimes \mathbf{R}^T\right)(\mathbf{Q}\widehat{\mathbf{m}} + \mathbf{vec}(\overline{\mathbf{X}})) \\
&= \frac{1}{s^2\sigma^2}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{Q}\widehat{\mathbf{\Omega}}\mathbf{Q}^T\left(\mathbf{I} \otimes \mathbf{R}\right) \\
&\quad \times \left(\mathbf{vec}(\mathbf{D}) - \frac{1}{s}\mathbf{F}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}})\right) \\
&\quad + \frac{1}{s}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}}), \qquad \text{(B.9)} \\
\mathbf{\Omega} &= E[\mathbf{vec}(\mathbf{X} - \mathbf{m})\mathbf{vec}(\mathbf{X} - \mathbf{m})^T] \\
&= \frac{1}{s^2}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{Q}\widehat{\mathbf{\Omega}}\mathbf{Q}^T\left(\mathbf{I} \otimes \mathbf{R}\right) \\
&= \left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{Q}\left(s^2\mathbf{\Sigma}_R^{-1} + \mathbf{Q}^T\left(\mathbf{I} \otimes \mathbf{R}\right)\frac{\mathbf{F}}{\sigma^2}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{Q}\right) \\
&\quad \times \mathbf{Q}^T\left(\mathbf{I} \otimes \mathbf{R}\right),
\end{aligned}$$

where $(\mathbf{vec}(\mathbf{D}) - \frac{1}{s}\mathbf{F}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}}))$ corresponds to non-rigid variations and $(\frac{1}{s}\left(\mathbf{I} \otimes \mathbf{R}^T\right)\mathbf{vec}(\overline{\mathbf{X}}))$ corresponds to rigid ones.

Since $\mathbf{Q}$ is a orthogonal to a subspace, $\mathbf{Q}_N(\overline{\mathbf{X}})$, on rigid motions, it means that we only consider an aligned prior mean shape $\frac{1}{s}\mathbf{R}^T\overline{\mathbf{X}}$ and non-rigid variations orthogonal to $\overline{\mathbf{X}}$ and other rigid variations are removed by $\mathbf{Q}$. However, we empirically found ignoring the Dirac-delta term makes the distribution $p(\mathbf{X}|\mathbf{D}, \mathbf{\Phi})$ close to the observation $\mathbf{D}$ and gives better reconstruction results, since s and $\mathbf{R}$ have inexact values in the early stage of the iteration process.

Appendix C

Procrustean Mixture Model Learning and Fitting in Chapter 4

C.1 Procrustean Mixture Model Learning

To model the deformation of 3D shapes, we use the Procrustean normal distribution (PND) [18], which makes 3D shapes closely aligned in a linear subspace. The PND can be extended to a mixture of PNDs as:

$$p(\mathbf{X}) = \sum_{k=1}^{K} \pi_k p(\mathbf{X}|c_k = 1), \tag{C.1}$$

where $\mathbf{X} \in \mathbb{R}^{3\times n_p}$ is a 3D shape satisfying $\mathbf{X1} = \mathbf{0}$ [1], n_p is the number of landmarks, and K is the number of mixture components. The mixing probability for the kth component is defined as $\pi_k = p(c_k = 1|\pi_k)$, where $\pi_k \geq 0$, such that $\sum_k \pi_k = 1$ and $c_k \in \{0, 1\}$ indicates which mixture component has generated the sample. $p(\mathbf{X}|c_k = 1)$ is a PND corresponding to the kth component, which is defined as $\mathcal{N}_P(\mathbf{Y}|\overline{\mathbf{X}}_k, \mathbf{Q}_k\mathbf{\Sigma}_{R_k}\mathbf{Q}_k^T)$ [18] where $\overline{\mathbf{X}}_k$, $\mathbf{\Sigma}_k$, and $\mathbf{Q}_k$ are the mean of aligned 3D shapes, the covariance matrix for

[1] In this chapter, we use $\mathbf{0}$ to denote both a matrix and a vector of zeros.

non-rigid variations, and the projection matrix to the linear subspace of non-rigid shapes, respectively. In addition, $\mathbf{Y} = s\mathbf{R}\mathbf{X}$ is an aligned shape using scale s and rotation $\mathbf{R}$.

Since we do not know the true number of mixture components, we introduce a Dirichlet-type prior on π based on the minimum message length (MML) principle [42]: $p(\pi) \propto \exp\left(-\frac{n_l}{2}\sum_k \ln \pi_k\right)$. Given N training 3D shapes, let us define the joint distribution $p(\mathbf{X}, \mathbf{c}, \pi)$ as

$$p(\mathbf{X}, \mathbf{c}, \pi) = \prod_{i=1}^{N}\prod_{k=1}^{K} \{p(\mathbf{X}_i|c_{ik} = 1)p(c_{ik} = 1|\pi_k)\}^{c_{ik}} p(\pi), \tag{C.2}$$

where i corresponds to the ith training sample. Then the parameters of (C.2) can be learned by maximizing the expected value of the following log-posterior function:

$$\begin{aligned}\Upsilon(\Phi|\Phi^{old}) = &\sum_i \sum_k w_{ik} \ln(p(\mathbf{X}_i, c_{ik} = 1|\Phi)) \\ &+ \ln(p(\pi)),\end{aligned} \tag{C.3}$$

where $\Phi = \{s_{ik}, \mathbf{R}_{ik}, \overline{\mathbf{X}}_k, \mathbf{\Sigma}_{R_k}, \mathbf{Q}_k, \pi_k | i = 1, \ldots, N, k = 1, \ldots, K\}$ is a set of model parameters, N is the number of samples, and K is the number of components.

E-step: We estimate $p(c_{ik} = 1|\mathbf{X}_i, \Phi^{old})$ given the current estimates of parameter Φ^{old} and observation $\mathbf{X}_i$[2]. Since the posterior distribution of c_{ik} plays the role as a weight for the component indicated by $\mathbf{c}_i$, we denote it as a weight w_{ik}. Using Bayes' rule, we have

$$w_{ik} = \frac{\pi_k p(\mathbf{X}_i|c_{ik} = 1, \Phi)}{\sum_l \pi_l p(\mathbf{X}_i|c_{il} = 1, \Phi_{il})}, \tag{C.4}$$

where $p(\mathbf{X}_i|c_{ik} = 1, \Phi)$ can be calculated by [18] with $\mathbf{Y}_{ik} = \mathbf{s}_{ik}\mathbf{R}_{ik}\mathbf{X}_i$.

M-step: The maximum posterior solutions of parameters are obtained using the posterior distribution of c_{ik}, *i.e.*, w_{ik} computed from the E-step. Since this optimization problem is the same as [18] except the prior term π, parameters s_{ik}, $\mathbf{R}_{ik}$, $\overline{\mathbf{X}}_k$, and $\mathbf{\Sigma}_{R_k}$

[2] The superscript *old* denotes the parameter set obtained from the previous M-step in the EM iteration procedure and we will omit the superscript (*old*) if no confusion arises.

are obtained by alternatively updating one parameter at a time as done in [18] for each component. The optimizing π_k is found as:

$$\pi_k = \max\Big(0, \sum_i w_{ik} - \frac{n_k}{2}\Big) \Big/ \sum_k \max\Big(0, \sum_i w_{ik} - \frac{n_k}{2}\Big). \tag{C.5}$$

An important feature of the M-step defined for finding π_k is that it performs component annihilation. To make more robust, we use the component-wise EM algorithm [42], such that each component is updated sequentially, *i.e.*, update π_1 and Φ_1, recompute all weights w_{ik}, update π_2 and Φ_2, recompute all weights w_{ik}, and so on.

The resulting PND components with parameters $\overline{\mathbf{X}}_k$, $\mathbf{\Sigma}_{R_k}$, and $\mathbf{Q}_k$ obtained from 3D training data $\mathbf{X}_i$ will be used as a prior model with parameters in Section C.2 of this supplementary material, *i.e.*, $\overline{\mathbf{X}}_k^{train} = \overline{\mathbf{X}}_k$, $\mathbf{\Sigma}_{R_k}^{train} = \mathbf{\Sigma}_{R_k}$, and $\mathbf{Q}_k^{train} = \mathbf{Q}_k$ when we fit a PNDMM to a 2D shape from a single image.

C.2 Procrustean Mixture Model Fitting

Unlike the prior model learning step discussed in the previous section, observations for our problem is not a 3D shape $\mathbf{X}$, but a 2D shape $\mathbf{D} \in \mathbb{R}^{2\times n_p}$. We regard the observation $\mathbf{D}$ as a sample obtained by a noisy orthographic projection of $\mathbf{X}$ with a zero mean Gaussian noise with variance σ^2 in each coordinate: $\mathbf{vec}(\mathbf{D}) = \mathbf{F}\mathbf{vec}(\mathbf{X}) + \mathbf{u}_i$, where $\mathbf{vec}$ is the vectorization operator, $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, and $\mathbf{F}$ is a projection matrix which removes the z-coordinate, the depth information (For more details, refer to Chapter 3). The joint distribution for a model fitting problem can be represented as

$$\begin{aligned} &p(\mathbf{D}, \mathbf{X}, \mathbf{c}|\sigma, \pi) \\ &= \prod_{k=1}^{K} \{p(\mathbf{D}|\mathbf{X}, \sigma) p(\mathbf{X}|c_k = 1) p(c_k = 1|\pi)\}^{c_k}, \end{aligned} \tag{C.6}$$

where the notation is the same as (C.2). We treat $\mathbf{X}$ in (C.6) as a hidden variable and estimate $\mathbf{X}$ using expectation-maximization (EM) algorithm based on the trained PNDMM.

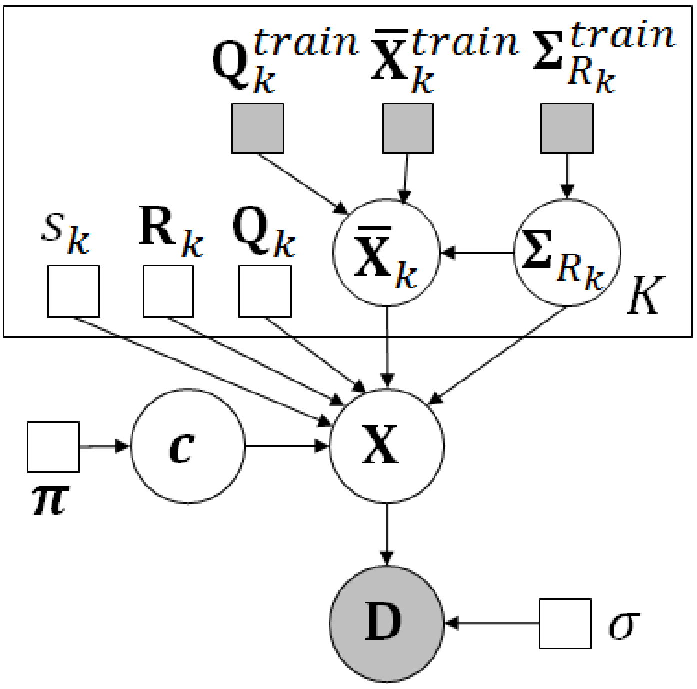


Figure C.1: A graphical representation of reconstruction process.

It is a detailed description in Section 4.3.2 of Chapter 4 and parameter notations follows that chapter. We extend the Procrustean mixture model to a single view 3D reconstruction problem with pre-learned prior information about $\overline{\mathbf{X}}_k^{train}$ and $\mathbf{\Sigma}_{R_k}^{train}$, which can be represented as in Figure C.1. Let Φ be a set of probabilistic parameters and let Θ be a set of deterministic parameters. Then $\Phi = \{\Phi_k | k = 1, \dots, K\}$, where $\Phi_k = \{\overline{\mathbf{X}}_k, \mathbf{\Sigma}_{R_k}\}$, and $\Theta = \{\Theta_k | k = 1, \dots, K\}$, where $\Theta_k = \{\sigma, s_k, \mathbf{R}_k, \mathbf{Q}_k, \pi_k\}$. Let Φ^{train} be a set of parameters related prior distributions, then $\Phi^{train} = \{\Phi_k^{train} | k = 1, \dots, K\}$, where $\Phi_k^{train} = \{\overline{\mathbf{X}}_k^{train}, \mathbf{\Sigma}_{R_k}^{train}, \mathbf{Q}_k^{train}, \beta, \nu\}$.

We formulate the model fitting to an 2D observation using the MAP-EM algorithm as

$$(\Phi, \Theta) = \arg\max_{\Phi,\Theta} \{\ln p(\mathbf{D}, \mathbf{X}, \mathbf{c} | \Phi, \Theta) + \ln p(\Phi | \Phi^{train})\} \tag{C.7}$$

and the cost function for the MAP-EM algorithm can be represented as

$$\begin{aligned} &J(\Phi, \Theta | \Phi^{old}, \Theta^{old}) \\ &\quad = \sum_k \Bigg(w_k \int \ln(p(\mathbf{D}, \mathbf{X}, c_k = 1 | \Phi_k, \Theta_k) \\ &\quad \times p(\mathbf{X} | c_k = 1, \mathbf{D}, \Phi_k^{old}, \Theta_k^{old}) d\mathbf{X} \\ &\quad + \ln p(\Phi_k | \Phi_k^{train}) \Bigg). \end{aligned} \tag{C.8}$$

Here, we use the chain rule as $p(\mathbf{X}, c_k = 1 | \mathbf{D}, \Phi_k^{old}, \Theta_k^{old}) = p(c_k = 1 | \mathbf{D}, \Phi_k^{old}, \Theta_k^{old}) p(\mathbf{X} | c_k = 1, \mathbf{D}, \Phi_k^{old} \Theta_k^{old})$ and denote $p(c_k = 1 | \mathbf{D}, \Phi_k^{old}, \Theta_k^{old})$ by w_k, since $p(c_k = 1 | \mathbf{D}, \Phi_k^{old}, \Theta_k^{old})$ plays the role as a weight for the component indicated by $\mathbf{c}$. The superscript old denotes the parameter set obtained from a previous M-step in the EM iteration procedure. From now on, we will omit the superscript (old) if no confusion arises.

The complete likelihood is represented by

$$\begin{aligned} &p(\mathbf{D}, \mathbf{X}, c_k = 1 | \Phi_k, \Theta_k) \\ &\quad = p(\mathbf{D} | \mathbf{X}, \sigma) p(\mathbf{X} | \overline{\mathbf{X}}_k, \mathbf{\Sigma}_{R_k}, c_k = 1) p(c_k = 1 | \pi_k), \end{aligned} \tag{C.9}$$

where $p(\mathbf{D}|\mathbf{X},\sigma)$ is for the observation noise and modeled by a Gaussian distribution, $p(\mathbf{X}|\overline{\mathbf{X}}_k,\boldsymbol{\Sigma}_{R_k},c_k=1)$ is for the kth PND component, and $p(c_k=1|\pi_k)$ is for the indicator variable, *i.e.*, $c_k=\{0,1\}$.

The prior distribution $p(\Phi_k|\Phi_k^{train})$ is represented by

$$\begin{aligned}&p(\Phi_k|\Phi_k^{train})\\&=p(\overline{\mathbf{X}}_k|\overline{\mathbf{X}}_k^{train},\beta,\boldsymbol{\Sigma}_{R_k})p(\boldsymbol{\Sigma}_{R_k}^{-1}|\boldsymbol{\Sigma}_{R_k}^{train^{-1}},\nu),\end{aligned}\tag{C.10}$$

where $p(\overline{\mathbf{X}}_k|\overline{\mathbf{X}}_k^{train},\beta,\boldsymbol{\Sigma}_{R_k})$, and $p(\boldsymbol{\Sigma}_{R_k}^{-1}|\boldsymbol{\Sigma}_{R_k}^{train^{-1}},\nu)$ are prior distributions for a mean shape $\overline{\mathbf{X}}_k$ of the kth PND, a precision matirx of non-rigid variations $\boldsymbol{\Sigma}_{R_k}^{-1}$, respectively. To apply pre-learned prior $\overline{\mathbf{X}}_k^{train}$ and $\boldsymbol{\Sigma}_{R_k}^{train}$, let $\overline{\mathbf{X}}_k$ have a PND as

$$\begin{aligned}&p(\overline{\mathbf{X}}_k|\overline{\mathbf{X}}_k^{train},\beta,\boldsymbol{\Sigma}_{R_k})\\&=\mathcal{N}_P(\overline{\mathbf{X}}_k|\overline{\mathbf{X}}_k^{train},\beta^{-1}\mathbf{Q}_k^{train}\boldsymbol{\Sigma}_{R_k}\mathbf{Q}_k^{train^T}),\end{aligned}\tag{C.11}$$

where $\mathbf{Q}_k^{train}$ is a PND parameter for the kth trained PND component, and let $\boldsymbol{\Sigma}_{R_k}$ have a Wishart distribution as

$$p(\boldsymbol{\Sigma}_{R_k}^{-1}|\boldsymbol{\Sigma}_{R_k}^{train^{-1}},\nu)=\mathcal{W}(\boldsymbol{\Sigma}_{R_k}^{-1}|\nu\boldsymbol{\Sigma}_{R_k}^{train^{-1}},\nu).\tag{C.12}$$

Given a test image, we compute mixture weights (E-step) using

$$w_k=\frac{\pi_k p(\mathbf{D}|c_k=1,\Phi_k)}{\sum_l \pi_l p(\mathbf{D}|c_l=1,\Phi_l)},\tag{C.13}$$

where $p(\mathbf{D}|c_k=1,\Phi_k)=\int p(\mathbf{D}|\mathbf{X},\sigma)p(\mathbf{X}|c_k=1,\Phi_k)d\mathbf{X}$. The posterior distribution of the true 3D shape, *i.e.*, $p(\mathbf{X}|c_k=1,\mathbf{D},\Phi_k,\Theta_k)$, can be represented by a Gaussian distribution [18].

Since the prior information about $\overline{\mathbf{X}}_k$, $\boldsymbol{\Sigma}_{R_k}$, and π_k affects M-step of EM procedure, the M-step is similar to [18] and the differences in the M-step are calculation of $\overline{\mathbf{X}}_k$ and $\boldsymbol{\Sigma}_{R_k}$.

For the optimization problem in the M-step for $\overline{\mathbf{X}}_k$ and $\mathbf{\Sigma}_{R_k}$ is

$$\begin{aligned}
&\max J(\Phi, \Theta|\Phi^{old}, \Theta^{old}) \\
&= -\sum_k w_k \Bigg(\frac{1}{2} \ln |\mathbf{\Sigma}_{R_k}| + \frac{1}{2} \mathbf{h}_k^T \mathbf{\Sigma}_{R_k}^{-1} \mathbf{h}_k \\
&+ \frac{s_k^2}{2} \mathrm{tr} \left((\mathbf{I} \otimes \mathbf{R}_k^T) \mathbf{Q}_k \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^T (\mathbf{I} \otimes \mathbf{R}_k) \mathbf{\Omega}_k \right) - \ln \pi_k \Bigg) \\
&- \sum_k \Bigg(\frac{1}{2} \ln |\mathbf{\Sigma}_{R_k}| + \frac{1}{2} \beta \mathbf{v}_k^T \mathbf{Q}_k^{train} \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^{train^T} \mathbf{v}_k \qquad \text{(C.14)} \\
&+ \frac{\nu - n_R - 1}{2} \ln |\mathbf{\Sigma}_{R_k}| + \frac{1}{2} \mathrm{tr}(\nu \mathbf{\Sigma}_{R_k}^{train} \mathbf{\Sigma}_{R_k}^{-1}) \Bigg), \\
&\text{subject to} \quad \sum_k \pi_k = 1, \quad \mathbf{R}_k^T \mathbf{R}_k = \mathbf{I}, \quad \left\| \overline{\mathbf{X}}_k \right\|_F^2 = 1, \\
&s_k \mathrm{tr}(\mathbf{R}_k \mathbf{M}_k \overline{\mathbf{X}}_k^T) = 1, \quad \mathbf{R}_k \mathbf{M}_k \overline{\mathbf{X}}_k^T \in \mathbf{S}_+^{n_d},
\end{aligned}$$

where $\mathbf{h}_k = \mathbf{Q}_k^T (s_k (\mathbf{I} \otimes \mathbf{R}_k) \mathbf{m}_k - \mathbf{vec}(\overline{\mathbf{X}}_k))$, $\mathbf{v}_k = \mathbf{vec}(\overline{\mathbf{X}}_k - \overline{\mathbf{X}}_k^{train})$, and $\mathbf{M}_k$ is the expectation of $\mathbf{X}$ with respect to its posterior distribution.

To update $\overline{\mathbf{X}}_k$, $\mathbf{Q}_k$ and $\mathbf{Q}_k^{train}$ are regarded as independent parameters with $\overline{\mathbf{X}}_k$ and $\overline{\mathbf{X}}_k^{train}$, differentiate (C.14) with respect to $\mathbf{X}_k$ and equate it to zero, and normalize the solution, as done in [18]. Then, the solution to a mean shape $\overline{\mathbf{X}}_k$ is

$$\overline{\mathbf{X}}_k = \frac{\Xi_k}{\|\Xi_k\|_F}, \qquad \text{(C.15)}$$

where Ξ_k is a matrix representation of $\overline{\mathbf{x}}_k$, *i.e.*, $\mathbf{vec}(\Xi_k) = \overline{\mathbf{x}}_k$ and $\overline{\mathbf{x}}_k = \big(w_k \mathbf{Q}_k \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^T + \beta \mathbf{Q}_k^{train} \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^{train}\big)^+ \big(w_k s_k \mathbf{Q}_k \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^T (\mathbf{I} \otimes \mathbf{R}_k) \mathbf{m}_k + \beta \mathbf{Q}_k^{train} \mathbf{\Sigma}_{R_k}^{-1} \mathbf{Q}_k^{train^T} \mathbf{vec}(\overline{\mathbf{X}}_k^{train})\big)$. $\beta > 0 \in \mathbb{R}$ plays a role as a prior weight in a posterior probability, and $\mathbf{m}_k$ is a posterior mean of $\mathbf{vec}(\mathbf{X}_k)$.

Also, a covariance matrix of non-rigid variation $\mathbf{\Sigma}_{R_k}$ can be obtained by solving the first-order necessary condition of (C.14), *i.e.*,

$$\begin{aligned}\mathbf{\Sigma}_{R_k} = \frac{1}{w_k + \nu - n_R}\Bigg(&w_k \mathbf{h}_k \mathbf{h}_k^T + \\ &w_k s_k^2 \mathbf{Q}_k^T \left(\mathbf{I} \otimes \mathbf{R}_k\right) \mathbf{\Omega}_k \left(\mathbf{I} \otimes \mathbf{R}_k^T\right) \mathbf{Q}_k \\ &\beta \mathbf{Q}_k^{train^T} \mathbf{vec}(\overline{\mathbf{X}}_k - \overline{\mathbf{X}}_k^{train}) \mathbf{vec}(\overline{\mathbf{X}}_k - \overline{\mathbf{X}}_k^{train})^T \mathbf{Q}_k^{train} \\ &+ \nu \mathbf{\Sigma}_{R_k}^{train}\Bigg),\end{aligned} \tag{C.16}$$

where $\nu > n_R - 1 \in \mathbb{R}$ and it also plays a role as a prior weight in a posterior probability, and $\mathbf{h}_k = \mathbf{Q}_k^T(s_k(\mathbf{I} \otimes \mathbf{R}_k)\mathbf{m}_k - \mathbf{vec}(\overline{\mathbf{X}}_k))$. However, in the case of a single view reconstruction based on noisy observation, the prior information has to be emphasized, *i.e.*, $w_k \ll \beta$ and $w_k \ll \nu$. Moreover, $w_k \ll 1$, thus the terms based on observed data in $\overline{\mathbf{X}}_k$ and $\mathbf{\Sigma}_{R_k}$ can be neglected as

$$\overline{\mathbf{X}}_k \approx \overline{\mathbf{X}}_k^{train}, \mathbf{\Sigma}_{R_k} \approx \frac{\nu \mathbf{\Sigma}_{R_k}^{train}}{w_k + \nu - n_R} \approx \mathbf{\Sigma}_{R_k}^{train}. \tag{C.17}$$

Since $\mathbf{Q}_k$ is calculated using the mean shape $\overline{\mathbf{X}}_k$, it can be calculated as $\mathbf{Q}_k \approx \mathbf{Q}_k^{train}$. It means that we can fit a PNDMM to a single 2D observation using a learned PND parameters $\overline{\mathbf{X}}_k^{train}$, $\mathbf{\Sigma}_{R_k}^{train}$, and $\mathbf{Q}_k^{train}$, and do not need to update them . If $\pi_k = 0$, the kth PND component is removed. After finishing EM iterations, the final posterior mean shape $\mathbf{M}_k$ of $\mathbf{X}$ corresponding to a PND component with the maximum weight w_k is used as a reconstructed 3D shape $\widehat{\mathbf{X}}$.

Bibliography

[1] A. Yao, J. Gall, G. Fanelli, and L. J. V. Gool, “Does human action recognition benefit from pose estimation?.” in *Proceedings of the British Machine Vision Conference*, 2011.

[2] A. Yao, J. Gall, and L. V. Gool, “Coupled action recognition and pose estimation from multiple views,” *International Journal of Computer Vision*, vol. 100, no. 1, pp. 16–37, 2012.

[3] C. Wang, Y. Wang, and A. L. Yuille, “An approach to pose-based action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[4] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, “Towards understanding action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[5] B. X. Nie, C. Xiong, and S.-C. Zhu, “Joint action recognition and pose estimation from video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[6] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[7] 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, 2006.

[8] 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.

[9] M. Paladini, A. Del Bue, M. Stošic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for Non-Rigid and Articulated Structure using Metric Projections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2898–2905.

[10] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure-from-motion factorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[11] E. Simo-Serra, A. Ramisa, G. Alenya, C. Torras, and F. Moreno-Noguer, "Single image 3D human pose estimation from noisy observations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[12] I. Radwan, A. Dhall, and R. Goecke, "Monocular image 3D human pose estimation under self-occlusion," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[13] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.

[14] D. G. Kendall, "The diffusion of shape," *Advances in Applied Probability*, vol. 9, no. 3, pp. 428–430, 1977.

[15] J. Gower, "Generalized Procrustes analysis," *Psychometrika*, vol. 40, pp. 33–51, 1975.

[16] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 53, no. 2, pp. 285–339, 1991.

[17] F. Jiao, S. Li, H.-Y. Shum, and D. Schuurmans, "Face alignment using statistical models and wavelet features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[18] M. Lee, J. Cho, C.-H. Choi, and S. Oh, "Procrustean normal distribution for non-rigid structure from motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[19] J. Cho, M. Lee, C.-H. Choi, and S. Oh, "EM-GPA: Generalized Procrustes analysis with hidden variables for 3D shape modeling," *Computer Vision and Image Understanding*, vol. 117, no. 11, pp. 1549–1559, 2013.

[20] W.-H. Cho, S.-W. Kim, M.-E. Lee, S.-H. Kim, S.-Y. Park, and C. bu Jeong, "Multimodality image registration using ordinary Procrustes analysis and entropy of bivariate normal kernel density," in *Proceedings of the IEEE International Conference on BioInformatics and BioEngineering*, 2008.

[21] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 587–602, 2011.

[22] B. Jian and B. C. Vemuri, “Robust point set registration using gaussian mixture models,” *IEEE IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1633–1645, 2011.

[23] L. Wang, H. Ning, W. Hu, and T. Tan, “Gait recognition based on Procrustes shape analysis,” in *Proceedings of the IEEE International Conference on Image Processing*, 2002.

[24] L. Wang, T. Tan, W. Hu, and H. Ning, “Automatic gait recognition based on statistical shape analysis,” *IEEE Transactions on Image Processing*, vol. 12, no. 9, pp. 1120–1131, sept. 2003.

[25] N. Duta, A. Jain, and M.-P. Dubuisson-Jolly, “Automatic construction of 2D shape models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 433–446, 2001.

[26] A. Patel and W. Smith, “3D morphable face models revisited,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[27] L. S. Shapiro, A. Zisserman, and M. Brady, “3D motion recovery via affine epipolar geometry,” *International Journal of Computer Vision*, vol. 16, pp. 147–182, 1995.

[28] P. Torr and D. Murray, “The development and comparison of robust methods for estimating the fundamental matrix,” *International Journal of Computer Vision*, vol. 24, pp. 271–300, 1997.

[29] L. S. Shapiro and J. M. Brady, “Feature-based correspondence: an eigenvector approach,” *Image and Vision Computing*, vol. 10, no. 5, pp. 283–288, 1992.

[30] 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.

[31] C. Bregler, A. Hertzmann, and H. Biermann, “Recovering non-rigid 3D shape from image streams,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2000.

[32] J. W. Demmel and M. T. H. Y, “Applied numerical linear algebra,” in *Society for Industrial and Applied Mathematics*, 1997.

[33] R. N. Bracewell, *The Fourier transform and its applications, 3rd ed.* New York: McGraw-Hill, 2000.

[34] R. A. Horn, *Topics in matrix analysis*. Cambridge University Press, 1986.

[35] J. Xiao, S. Baker, I. Matthews, and T. Kanade, “Real-time combined 2D+3D active appearance models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[36] I. Matthews, J. Xiao, and S. Baker, “2D vs. 3D deformable face models: Representational power, construction, and real-time fitting,” *International Journal of Computer Vision*, vol. 75, no. 21, pp. 93–113, 2007.

[37] I. Matthews and S. Baker, “Active appearance models revisited,” *International Journal of Computer Vision*, vol. 60, pp. 135–164, 2003.

[38] G. H. G. Ake Bjorck, “Numerical methods for computing angles between linear subspaces,” *Mathematics of Computation*, vol. 27, no. 123, pp. 579–594, 1973.

[39] H.-S. Lee, S. Park, B.-N. Kang, J. Shin, J.-Y. Lee, H. Je, B. Jun, and D. Kim, “The POSTECH face database (PF07) and performance evaluation,” in *Proceedings*

of the IEEE International Conference on Automatic Face Gesture Recognition, 2008.

[40] T. F. Cootes and C. J. Taylor, "An algorithm for tuning an active appearance model to new data," in *Proceedings of the British Machine Vision Conference*, 2006.

[41] J. Cho, M. Lee, and S. Oh, "Complex non-rigid 3D shape recovery using a procrustean normal distribution mixture model," *International Journal of Computer Vision (Online)*, pp. 1–21, 2015.

[42] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[43] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for mixtures," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, pp. 697–712, 2001.

[44] N. van der Aa, X. Luo, G. Giezeman, R. Tan, and R. Veltkamp, "Utrecht multi-person motion (UMPM) benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction," in *Proceedings of the Workshop on Human Interaction in Computer Vision*, 2011.

[45] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Trajectory space: A dual representation for nonrigid structure from motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1442–1456, 2011.

[46] W. Zhang, M. Zhu, and K. Derpanis, "From actemes to action: A strongly-supervised representation for detailed action understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013.

[47] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall Professional Technical Reference, 2002.

[48] P. F. Gotardo and A. M. Martinez, “Non-rigid structure from motion with complementary rank-3 spaces,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[49] Y. Zhu and S. Lucey, “Convolutional sparse coding for trajectory reconstruction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 529–540, 2015.

[50] I. Akhter, Y. Sheikh, and S. Khan, “In defense of orthonormality constraints for nonrigid structure from motion,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[51] Y. Zhu, D. Huang, F. D. L. Torre, and S. Lucey, “Complex non-rigid motion 3D reconstruction by union of subspaces,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[52] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[53] M. Lee, C.-H. Choi, and S. Oh, “A procrustean markov process for non-rigid structure recovery,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[54] M. Salzmann, R. Urtasun, and P. Fua, “Local deformation models for monocular 3D shape recovery,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[55] A. Varol, M. Salzmann, E. Tola, and P. Fua, "Template-free monocular reconstruction of deformable surfaces," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[56] J. Taylor, A. D. Jepson, and K. N. Kutulakos, "Non-rigid structure from locally-rigid motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[57] J. Fayad, L. Agapito, and A. Del Bue, "Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences," in *Proceedings of the European Conference on Computer Vision*, 2010.

[58] M. L. Zelditch, D. L. Swiderski, and H. D. Sheets, *Geometric Morphometrics for Biologists: A primer.* Elsevier/Academic Press, 2012.

[59] D. Pizarro and A. Bartoli, "Global optimization for optimal generalized Procrustes analysis," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.

[60] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[61] Y. Zhu, M. Cox, and S. Lucey, "3D motion reconstruction for real-world camera motion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[62] A. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[63] L. Sigal, A. Balan, and M. J. Black, “Combined discriminative and generative articulated pose and non-rigid shape estimation,” in *Proceedings of the Advances in Neural Information Processing Systems*, 2007.

[64] L. Bo and C. Sminchisescu, “Structured output-associative regression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[65] ——, “Twin Gaussian processes for structured prediction,” *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 28–52, 2010.

[66] C. Ionescu, J. Carreira, and C. Sminchisescu, “Iterated second-order label sensitive pooling for 3D human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[67] V. Ramakrishna, T. Kanade, and Y. Sheikh, “Reconstructing 3D human pose from 2D image landmarks,” in *Proceedings of the European Conference on Computer Vision*, 2012.

[68] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer, “A joint model for 2D and 3D pose estimation from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[69] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, “Robust estimation of 3D human poses from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[70] X. Fan, K. Zheng, Y. Zhou, and S. Wang, “Pose locality constrained representation for 3D human pose reconstruction,” in *Proceedings of the European Conference on Computer Vision*, 2014.

[71] X. H. K. D. Xiaowei Zhou, Spyridon Leonardos, "3d shape estimation from 2D landmarks: A convex relaxation approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[72] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[73] D. Park and D. Ramanan, "N-best maximal decoders for part models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[74] D. Ramanan, "Learning to parse images of articulated bodies," in *Proceedings of the Advances in Neural Information Processing Systems*, 2006.

[75] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1-2, pp. 4–27, 2010.

[76] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proceedings of the British Machine Vision Conference*, 2010.

[77] M. Eichner, V. Ferrari, and S. Zurich, "Better appearance models for pictorial structures." in *Proceedings of the British Machine Vision Conference*, 2009.

[78] J. Cho, M. Lee, H. J. Chang, and S. Oh, "Robust action recognition using local motion and group sparsity," *Pattern Recognition*, vol. 47, no. 5, pp. 1813–1825, 2014.

[79] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, 1999.

[80] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Speeded-up robust features (SURF),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[81] Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, 2010.

[82] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the International Conference on Multimedia*, 2007.

[83] A. Klaser, M. Marszałek, and C. Schmid, “A spatio-temporal descriptor based on 3d-gradients,” in *Proceedings of the British Machine Vision Conference*, 2008.

[84] G. Willems, T. Tuytelaars, and L. V. Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *Proceedings of the European Conference on Computer Vision*, 2008.

[85] L. Yeffet and L. Wolf, “Local trinary patterns for human action recognition,” in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2009.

[86] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local svm approach,” in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2004.

[87] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007.

[88] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[89] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[90] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: a large video database for human motion recognition," in *Proceedings of the IEEE International Conference on Pattern Recognition*, 2011.

[91] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv:1212.0402*, vol. abs/1212.0402, 2012. [Online]. Available: http://arxiv.org/abs/1212.0402

[92] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[93] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[94] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[95] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 1981.

[96] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[97] X.-T. Yuan and S. Yan, "Visual classification with multi-task joint sparse representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[98] T. Guha and R. Ward, "Learning sparse representations for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1576–1588, 2012.

[99] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[100] W. Deng, W. Yin, and Y. Zhang, "Group sparse optimization by alternating direction method," *Technical Report 11-06, Department of Computational and Applied Mathematics, Rice University*, 2011.

[101] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[102] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the Scandinavian Conference on Image Analysis*, 2003.

[103] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[104] V. K. Singh and R. Nevatia, "Action recognition in cluttered dynamic scenes using pose-specific part models," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[105] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[106] S. Zuffi and M. J. Black, "Puppet flow," in *Technical Report RT-IS-MPI-007, MPI for Intelligent Systems*, 2013.

[107] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, 2007.

[108] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proceedings of the British Machine Vision Conference*, 2009.

[109] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[110] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, 2011.

[111] B. Scholkopf, A. Smola, and K.-R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

초록

최근 컴퓨터 비전 기술은 증강현실, 비디오 이미지 분석 등 지능형 시스템에서 점점 더 중요한 역할을 하고 있으며, 마이크로소프트의 키넥트와 같이 비용효율이 높은 깊이카메라도 개발되고 있다. 하지만 여전히 많은 컴퓨터 비전 알고리즘들은 일반적인 컬러 카메라로부터 얻어진 2차원 관측을 가정하에 개발되고 있기 때문에 만약 우리가 2차원 관측으로부터 3차원 정보를 추론할 수 있다면 많은 컴퓨터 비전 문제에서 좀 더 좋은 해결책을 제시할 수 있을 것이다.

본 논문은 2차원 관측들로부터 3차원 관측을 추론하는 것에 집중한다. 이러한 것은 비강체의 구조와 움직임 복원(NRSfM)으로 잘 알려져 있으며, NRSfM은 하나의 물체에 대한 변형이 낮은 차원의 공간에서 모델링될 수 있다는 가정하에 다수의 이미지를 분석함으로써 물체의 3차원 구조를 얻는 방법이다. 그러나 오랜 시간 동안의 사람의 몸은 매우 복잡한 변화가 가능하고, 이로인한 자유도의 증가는 문제를 보다 복잡하게 만든다. 본 논문에서는 이러한 복잡한 형상의 움직임을 보다 효율적으로 모델링하기 위하여 최근에 제안된 3차원 형상의 강체 움직임과 비강체 움직임을 구분하여 비강체 움직임만을 효율적으로 모델링 Procrustean 정규 분포(PND)를 복합 형태로 확장하는 방법을 제안하고, Procrustean 정규 분포 혼합 모델로 명명하였다. 기존의 NRSfM 방법들이 3차원 형상을 추론하기 위하여 하나의 모델을 사용하는데 반해, 제안하는 방법은 복잡한 형상의 움직임을 보다 단순한 움직임으로 그룹화하고, 각 그룹을 Procrustean 정규 분포를 통해 모델링함으로써 모델 학습을 보다 간단하고 정확하게 하였다. 본 논문에서는 실험을 통하여 제안된 방법을 긴 시간동안 복잡한 움직임을 하는 사람의 2차원 형상에 적용하여 기존의 방법보다 3차원 형상을 더 잘 추론할 수 있음을 보였다.

또한, 본 논문은 제안된 Procrustean 정규 분포 혼합 모델을 한 장의 이미지에서 사람의 3차원 자세를 추론하는 문제로 확장하였다. 한 장의 이미지에서 사람의 3차원 형상을 복원하는 문제는 중요한 문제임에도 불구하고 문제의 해가 하나로 주어지지 않는 모호성이 존재한다. 더욱이 2차원 관측으로 3차원 형상을 복원하기에 앞서 정확한 2차원 관측을 얻는 것이 필요하다. 본 논문에서는 2차원 관측의 부정확성과 3

차원 복원의 모호성의 문제를 해결하기 위하여, 다수의 2차원 관측과 3차원 형상을 추론하고 2차원 자세 검출기와 3차원 형상 추론 모델 모두에서 잘 설명되는 하나의 2차원 관측 및 3차원 형상 추론 결과를 선택함으로써 해결하고자 하였다. 본 논문은 또한 학습 데이터와 시험 데이터가 다른 경우 모델을 이미지에서 추론한 관측 방향으로 변형시킴으로써 새로운 이미지에도 적용이 가능하도록 모델변환을 도입하였다. 본 논문은 실험을 통하여 제안하는 방법이 새로운 시험 이미지에 적용한 경우에도 좋은 결과를 보여줌을 확인하였다.

마지막으로 본 논문은 제안하는 3차원 형상 복원 방법을 동영상에 적용함으로써 행동인식 문제를 다루고 있다. 최근 행동인식에 관한 연구는 이미지에서 추정된 2차원 사람의 자세와 같은 고수준(high-level)의 특징이 이미지의 겉모습과 픽셀 단위의 움직임과 같은 저수준(low-level)의 특징보다 더 좋은 성능을 보여줄 수 있음을 말하고 있다. 앞서 본 논문에서 제안하는 방법들은 한 장의 이미지에서 사람의 3차원 자세를 추론할 수 있기 때문에, 본 논문은 이러한 최근 논문의 흐름에서 한발 더 나아가 동영상에서 3차원 자세를 추정하고 이를 이용한 행동 인식 방법을 제안한다. 시험 결과는 3차원 자세에 기반을 둔 행동 묘사자가 2차원 자세에 기반을 둔 행동 묘사자보다 더 좋은 성능을 보여주었다. 본 논문에서 제안한 3차원 행동묘사자는 다수의 2차원 영상으로부터 학습된 3차원 형상 모델을 통하여 얻어진 것임을 고려할 때, 본 논문에서 도출된 결과는 매우 유망하다고 볼 수 있으며, 2차원 정보로부터 3차원 정보를 추론하는 연구는 보다 신뢰성 있는 컴퓨터 비전 시스템 개발을 위해 여전히 중요한 연구 주제라고 할 수 있다.

주요어: 3차원 형상 복원, 비강체 형상 및 움직임 복원, 3차원 자세 추정, 행동인식

학 번: 2010-20902