Ph.D. DISSERTATION

3D Reconstruction, Weakly-Supervised Learning, and Supervised Learning Methods for 3D Human Pose Estimation

3차원 사람 자세 추정을 위한
3차원 복원, 약지도학습, 지도학습 방법

BY

SUNGHEON PARK

JANUARY 2019

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

3D Reconstruction, Weakly-Supervised Learning, and Supervised Learning Methods for 3D Human Pose Estimation

3차원 사람 자세 추정을 위한
3차원 복원, 약지도학습, 지도학습 방법

BY

SUNGHEON PARK
JANUARY 2019

Intelligent Systems
Department of Transdisciplinary Studies
Graduate School of Convergence Science and Technology
SEOUL NATIONAL UNIVERSITY

3D Reconstruction, Weakly-Supervised Learning, and Supervised Learning Methods for 3D Human Pose Estimation

3차원 사람 자세 추정을 위한
3차원 복원, 약지도학습, 지도학습 방법

지도교수 곽 노 준
이 논문을 공학박사 학위논문으로 제출함

2019년 1월

서울대학교 대학원

융합과학부 지능형융합시스템전공

박 성 헌

박성헌의 공학박사 학위 논문을 인준함

2019년 1월

위 원 장:	이 교 구	(인)
부위원장:	곽 노 준	(인)
위 원:	서 봉 원	(인)
위 원:	최 상 일	(인)
위 원:	이 민 식	(인)

Abstract

Estimating human poses from images is one of the fundamental tasks in computer vision, which leads to lots of applications such as action recognition, human-computer interaction, and virtual reality. Especially, estimating 3D human poses from 2D inputs is a challenging problem since it is inherently under-constrained. In addition, obtaining 3D ground truth data for human poses is only possible under the limited and restricted environments. In this dissertation, 3D human pose estimation is studied in different aspects focusing on various types of the availability of the data. To this end, three different methods to retrieve 3D human poses from 2D observations or from RGB images—algorithms of 3D reconstruction, weakly-supervised learning, and supervised learning—are proposed.

First, a non-rigid structure from motion (NRSfM) algorithm that reconstructs 3D structures of non-rigid objects such as human bodies from 2D observations is proposed. In the proposed framework which is named as *Procrustean Regression*, the 3D shapes are regularized based on their aligned shapes. We show that the cost function of the Procrustean Regression can be casted into an unconstrained problem or a problem with simple bound constraints, which can be efficiently solved by existing gradient descent solvers. This framework can be easily integrated with numerous existing models and assumptions, which makes it more practical for various real situations. The experimental results show that the proposed method gives competitive result to the state-of-the-art methods for orthographic projection with much less time complexity and memory requirement, and outperforms the existing methods for perspective projection.

Second, a weakly-supervised learning method that is capable of learning 3D structures when only 2D ground truth data is available as a training set is presented. Extending the Procrustean Regression framework, we suggest *Procrustean Regression Network*, a learning method that trains neural networks to learn 3D structures using training data with 2D ground truths. This is the first attempt that directly integrates an NRSfM algorithm into neural network training. The cost function that contains a low-rank function is also firstly used as a cost function of neural networks that reconstructs 3D shapes. During the test phase, 3D structures of human bodies can be obtained via a feed-forward operation, which enables the framework to have much faster inference time compared to the 3D reconstruction algorithms.

Third, a supervised learning method that infers 3D poses from 2D inputs using neural networks is suggested. The method exploits a relational unit which captures the relations between different body parts. In the method, each pair of different body parts generates relational features, and the average of the features from all the pairs are used for 3D pose estimation. We also suggest a dropout method called *relational dropout*, which can be used in relational modules to impose robustness to the occlusions. The experimental results validate that the performance of the proposed algorithm does not degrade much when missing points exist while maintaining state-of-the-art performance when every point is visible.

keywords: 3D human pose estimation, non-rigid structure from motion, relational networks, procrustean regression, 3D reconstruction, deep learning, weakly-supervised learning

student number: 2014-30814

Contents

List of Tables

List of Figures

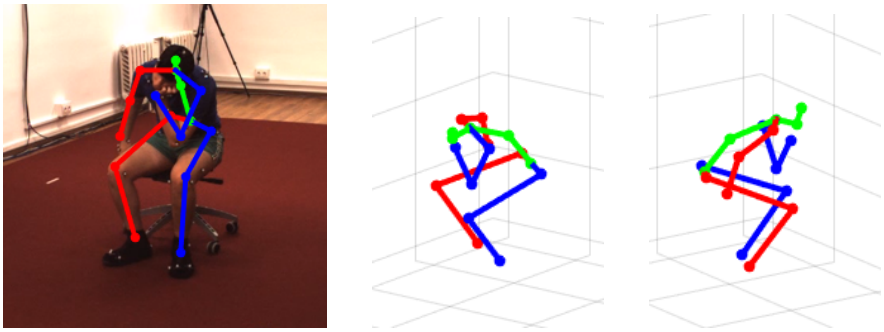
Chapter 1

Introduction

Human bodies are representative examples of non-rigid objects that has various shapes and poses. Since early computer vision literature, human pose estimation (HPE) has been one of the fundamental research areas. The importance of HPE is significant in that HPE is served as a preceding task for many computer vision applications such as virtual/augmented reality, human-computer interaction, surveillance systems, etc.

HPE has been actively researched for decades. Despite numerous researches conducted in the past, there is still plenty of room for improvement, especially for the pose estimation in 3D space. 3D HPE, which aims to recover 3D structures of human bodies from raw inputs such as RGB images or depth images, is treated as a more challenging task compared to the 2D HPE mainly due to the increased degree of freedom. In addition, when noisy or missing data is given as an input due to the reasons such as self-occlusion or illumination variations, it becomes even harder to estimate accurate 3D poses.

Humans have an ability to perceive depth of scenes using two eyes as a stereo vision system. Meanwhile, even in the case when the stereo vision sys-



(a) Input image and 2D pose (b) 3D pose - view 1 (c) 3D pose - view 2

Figure 1.1: Illustration of 3D human pose estimation. The task aims to retrieve human poses in 3D space when RGB images or 2D poses are given as inputs. In this dissertation, we are interested in finding 3D positions of joints of human bodies rather than reconstructing 3D mesh of humans.

tem cannot be applied, e.g., in the case of watching objects far away or watching 2D images, humans easily infer 3D structure of the objects they are watching in most cases. Those abilities are originated from exploiting background knowledge of depth information that humans have learned through their lives.

Aiming to provide algorithms reflecting the ability of 2D-to-3D conversion of humans, in this dissertation, 3D human poses are estimated from RGB images or 2D observations. Hence, 3D HPE using depth cameras is not considered, and only 2D information is used as an input of the algorithms proposed in this dissertation. As depicted in Figure 1.1, we are interested in estimating 3D position of the representative joints in human bodies rather than volumetric dense reconstruction of humans.

Although there exist 3D human pose datasets [97, 49, 72] which are acquired using motion capture systems, motion capture data can only be obtained under carefully controlled environments. This limits the diversity of RGB im-

ages, 2D poses, and 3D poses in the datasets. To fully exploit abundant RGB images on the web which contain various human poses in various situations, 3D reconstruction algorithms, unsupervised, and weakly-supervised learning methods also play an important role as well as supervised learning methods.

Thus, this dissertation scrutinizes 3D HPE under various configurations. Especially, we focus on how to recover 3D poses in underconstrained situations where infinite number of solution is possible. We solve this underconstrainedness by assuming low-rank shape variations or by learning from training data. We proposed three different methods for 3D HPE to provide solutions for various data configurations: a 3D reconstruction method, a weakly-supervised learning method, and a supervised learning method.

First, we propose a new method that reconstructs 3D shapes of human bodies from a sequence of 2D observations. To this end, we suggest a novel framework of non-rigid structure from motion algorithm which can be used to reconstruct any non-rigid objects as well as human bodies. A novel cost function that has both flexibility and generality are suggested, and it can be easily optimized using gradient descent methods with reduced time and space complexity.

Second, a weakly-supervised learning method that aims to learn 3D structures of human bodies via neural networks when only 2D annotations are available as ground truth is proposed. By extending the cost function of 3D reconstruction algorithm and applying it to the cost function of neural networks, the networks successfully learn to predict 3D human pose from a single 2D input.

Third, a supervised learning method that can be used when 3D ground truth data is available as a training set is suggested. A simple yet effective neural network structure based on the relational modules [93] is designed. A novel training and testing strategy are also proposed, which is able to effectively handle

the situation that missing observations due to occlusion exists.

The remainder of this chapter is organized as follows. In Section 1.1, we define the problems to solve throughout this dissertation. The differences of the three tasks dealt with in this dissertation is concretely explained. Then, motivation and applications of 3D HPE are discussed in Section 1.2. Challenges that makes 3D HPE difficult are enumerated in Section 1.3. Contributions of the algorithms proposed in this dissertation is argued in Section 1.4. Finally, an outline of the dissertation is given in Section 1.5.

1.1 Problem Definition

The aim of this dissertation is to develop various algorithms that estimate accurate 3D human poses. As previously stated, in this dissertation, 3D human pose is defined as a 3D position of representative joints in human bodies such as wrists, ankles, knees, hips, and so on. We tackled 3D HPE problem under three different perspectives, define the problem to solve for each perspective, and proposed the solution for each task. Inputs, outputs, and data settings are slightly different for each method. In this section, we provide detailed problem settings for each method. The three tasks address in this dissertation are 3D reconstruction, weakly-supervised learning, and supervised learning, and we illustrated the main differences of input and output settings, learning strategies, and types of available ground truth data for the three tasks in Figure 1.2.

The first task is to reconstruct 3D structures of human bodies from a sequence of 2D observations. 3D reconstruction of non-rigid objects, which is known as non-rigid structure from motion (NRSfM), is a research area we dove in for this task. Unlike rigid structure from motion, which reconstructs 3D

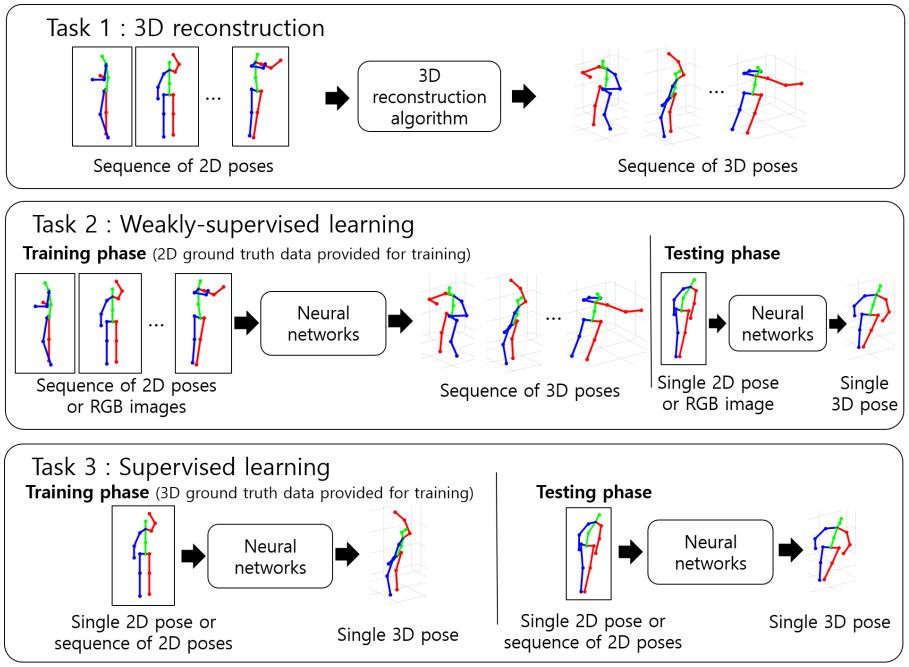


Figure 1.2: Illustration of the tasks addressed in this dissertation. We tackled 3D human pose estimation under three different viewpoints: 3D reconstruction, weakly-supervised learning, and supervised learning.

shapes of rigid and non-moving objects using 2D observations from multiple views [44], targets for NRSfM algorithms are non-rigid and moving ones. The problem of NRSfM is defined as follows: Given a sequence of 2D observations of non-rigid objects at multiple frames, reconstruct 3D shapes of the objects for all frames. Generally, NRSfM algorithms are not learning-based methods. Therefore, they does not require template or dictionary learning steps. Although most of experiments are conducted on datasets for human poses in this dissertation, NRSfM algorithms, including the one proposed in Chapter 3, is applicable to any non-rigid objects.

The second task is to provide a weakly-supervised learning method for

3D HPE. Weakly-supervised learning covers broad concepts of learning that use incomplete, inexact, and inaccurate supervision [133]. In this dissertation, we adopted the term weakly-supervised by meaning it incomplete supervision. Specifically, to learn a model that estimates 3D human pose, we use only 2D observations, i.e., x and y components, as ground truth data. During training phase, a sequence of 2D observations from multiple frames is given as input, and the model outputs estimated 3D structure of each input. Unlike the training inputs, test inputs does not use sequential information, and the inference of 3D pose is done using a single input.

Lastly, we aim to provide a fully supervised learning method for 3D HPE. Different from the above tasks, we use 3D pose information provided as a ground truth for the training data. The model accepts 2D pose estimation result from RGB images. We used off-the-shelf 2D pose estimator to locate the position of body joints in RGB images. The location of the joints are fed to the model that we proposed, and the model outputs 3D position of all joints.

To summarize the differences, the 3D reconstruction method does not require a learning step while the weakly-supervised and the supervised learning methods have split training and test datasets. We make use of neural networks to apply learning based methods. The supervised learning method fully exploits 3D ground truth data for training whereas only 2D ground truth data is given for training in the weakly-supervised learning method.

1.2 Motivation

We will discuss the importance of 3D HPE task and the proposed algorithms in this section. Understanding human behaviors is one of the most popular sub-

jects for computer vision research. In most of research topics related to human behavior analysis, human pose estimation plays an important role. In particular, 3D pose provides abundant information than 2D pose, which enables more detailed analysis of human behaviors. They can be further applied to the practical applications across various kinds of industries. In the field of human-computer interaction, 3D HPE helps to improve user experience by recognizing gestures or actions of humans. Augmented reality (AR) and virtual reality (VR) are also the major fields that 3D HPE plays a crucial role. If 3D avatar of a human that mimics behavior of users can be built in real-time by applying 3D HPE algorithms, it can be utilized to construct realistic AR/VR applications. On the other hand, gait analysis via 3D HPE can be used to identify individuals for surveillance purposes. In a nutshell, 3D HPE is a fundamental or preceding task for many computer vision applications.

While accurate 3D pose can be obtained from a motion capture system [49, 97, 72] or using multiple depth and RGB cameras [51], those systems require expensive equipments installed under carefully designed environments. This restricts acquisition of 3D pose information only to constrained situations. Meanwhile, a huge amount of RGB images containing a variety of human poses in various situations can be easily crawled from the Internet. Ground truth data for 2D pose estimation can be obtained by human efforts mostly via crowdsourcing. There are a few publicly available large-scale datasets [9, 66] that contain various poses in uncontrolled environments. Therefore, it is essential to develop a system that estimates 3D pose using only 2D observations in order to liberate 3D pose estimation from space and budget constraints.

Last but not least, understanding 3D structure of scenes is not only an important preceding task for many applications, but also provide meaningful insights

in terms of theoretical perspective. For NRSfM, research papers have different underlying assumptions of their own, and there is no consensus on which assumption is the most appropriate one. Finding appropriate assumption that shows good reconstruction results may be helpful for better understanding to non-rigid 3D reconstruction problems.

1.3 Challenges

There are a few obstacles that make 3D HPE difficult. First of all, the problem is inherently underconstrained. Large amount of information loss occurs when 3D points are project to 2D space. Since we have to infer 3D structures of humans from 2D poses, there are infinite number of configurations that satisfy 2D pose constraint. We will solve this underconstrained problem by introducing additional assumptions for 3D shapes over multiple frames or by learning 2D-to-3D mappings from training data.

The diversity of human poses is another challenging characteristics for 3D HPE. As representative non-rigid objects, shapes of human bodies can be deformed in various ways. There are large number of pose variations depending on actions of humans. Actions such as walking, sitting, and lying have quite different poses as illustrated in Figure 1.3. 3D HPE algorithms need to deal with those large pose variations and deformations. Since humans bodies are non-rigid but articulated objects, plausible configurations of 3D poses given 2D poses can be learned from training data.

Leaving those inherent difficulties aside, accurate pose estimation may be suffered by photometric variations in the input images. When 3D pose should be inferred from RGB images, illumination variations in the images may result

Figure 1.3: Pose variations on Human 3.6M dataset [49]. Self-occlusion of joints by other body parts can be observed. Also note that the images have different illumination conditions.

inaccurate estimation result. In addition, humans in the image may appeared in different scales and resolutions. Therefore, 3D HPE algorithms should cope with scale and illumination variations.

Another challenge is to handle occlusions. In real world images, people may be cluttered by another object or person. Also, when 3D body is projected into 2D by the camera, self-occlusion is unavoidable. Therefore, 3D HPE algorithms should robustly estimate 3D poses when part of input data is missing.

1.4 Contributions

The contributions of this dissertation are discussed for each task (3D reconstruction, weakly-supervised learning, supervised learning) in this section.

1.4.1 3D Reconstruction of Human Bodies

For the 3D reconstruction task, a non-rigid structure from motion (NRSfM) method that reconstructs 3D shapes of deformable objects is suggested. Motivated by the previously proposed method that takes 3D shape alignment into consideration [59], a novel framework called *Procrustean Regression* is pre-

sented. In the regression framework, NRSfM is formulated as a least squares problem with a regularization term. The proposed cost function can be easily casted into an unconstrained problem or a simple bounded-constrained problem without any relaxation or approximation step, which can be optimized using gradient descent methods.

In addition, the proposed framework is flexible in that various models can be directly integrated to the framework. For instance, the proposed method is able to cover both orthographic and perspective camera models by changing the data term, or it is possible to add a smoothness constraint to the regularization term.

Besides the ease of optimization and the flexibility of the framework, Procrustean Regression is advantageous over PND [59] in terms of time and space efficiency. The framework shows less memory consumption and faster running time compared to EM-PND [59] while maintaining the strength of shape alignment in NRSfM algorithms.

Extensive experiments on various datasets including synthetic and real world data prove that the proposed framework gives competitive result to state-of-the-art methods for orthographic projection with much less time complexity and memory requirement, and outperforms the existing methods for perspective projection. Besides the forms of the data term and the regularization term introduced in this dissertation, other kinds of cost functions can be easily integrated into the proposed framework with little modification. Hence, the flexibility of the proposed framework ease the process of applying newly designed cost functions for future research of NRSfM.

1.4.2 Weakly-Supervised Learning for 3D HPE

We extended Procrustean Regression to adopt it to the learning framework using neural networks. To this end, we propose a novel framework named *Procrustean Regression Network* (PRN) which learns to infer 3D structures of deformable objects when only 2D ground truth is available. The cost function of Procrustean Regression is slightly modified in order to use it directly as a cost function of neural networks. With analytical derivation of the gradients of the cost function, the neural networks are optimized using back-propagation. The whole training procedure is done in an end-to-end manner, and the reconstruction result of a single image or 2D shape is generated at the test phase via a simple forward propagation without requiring any post processing step for 3D reconstruction.

To the best of our knowledge, PRN is the first work that uses an NRSfM algorithm to deep neural networks directly as a loss function. In addition, PRN is the first work that contains low-rank optimization in the loss function of the network that reconstructs 3D shapes. The experimental results verify that PRN effectively reconstructs the 3D shapes of human skeletons, in that the prediction results from PRN are superior to the outputs of the network trained with the 3D data obtained from the conventional NRSfM methods. Although we limit the application of the algorithm to human pose estimation in this dissertation, the algorithm can be applied to any deformable object class since it does not explicitly learn any category-specific templates.

1.4.3 Supervised Learning for 3D HPE

Lastly, a supervised learning method that use 3D ground truth as a training data is proposed. A novel neural network structure is suggested which accepts 2D po-

sition of the joints as inputs, and outputs 3D positions of the joints. The network structure is based on the relational modules proposed in [93] which is designed to capture relations between two different objects. Human bodies are consists of arms, legs, a head, and a torso, each of which has distinctive behaviors and movements. Motivated by this idea, the network that learns the relations among different body parts is designed. The features from all pairs of groups are averaged to generate the feature vectors which are used for 3D pose regression. We found this simple structure outperforms the fully connected baseline network and shows competitive performance to the state-of-the-art methods.

In addition, a regularization method that can impose robustness to the missing points during the training is also proposed. The method, named as *relational dropout*, randomly drops one of the pair features when they are averaged, which simulates the case that certain groups of joints are missing during the training. Experiments validate that the model trained with relational dropout effectively produces plausible results even in the existence of missing joints.

1.5 Outline

The structure of this dissertation is composed as follows: In chapter 2, prior works related to non-rigid structure from motion and human pose estimation are reviewed. Then, the proposed algorithms for 3D HPE is discussed through Chapter 3 to Chapter 5. We presented the proposed algorithms in the order of increasing information availability. Chapter 3 proposes 3D reconstruction algorithm for human bodies and non-rigid objects when 2D observations are given. Chapter 4 presents a weakly-supervised learning method for 3D human pose estimation which learns to infer 3D human pose when only 2D ground truth

of training data is provided. Chapter 5 gives a supervised learning method for 3D human pose which implicitly learns relations between the body components when 3D ground truth data can be used for training. Finally, chapter 6 provides concluding remarks, limitations, and future directions of this research.

Chapter 2

Related Works

In this chapter, we will discuss prior researches related to the 3D human pose estimation across broad areas. First, in Section 2.1, a history of human pose estimation, focused on 2D HPE in images, is described. Then, researches that tackles 3D human pose estimation problem are discussed in Section 2.2. This section mainly focuses on supervised learning methods that use 3D ground truth poses data, which are closely related to Chapter 5. In Section 2.3, a thorough review of non-rigid structure from motion is provided. NRSfM is an area that related to Chapter 3 which reconstructs non-rigid objects from 2D observations. In Section 2.4, we review the works that tackled 3D reconstruction of rigid or non-rigid objects using neural networks, which are closely related to the weakly-supervised learning method that uses neural networks proposed in Chapter 4.

2.1 2D Human Pose Estimation

Human pose estimation from RGB images started from locating human body joints in the images. Early works for 2D human pose estimation which are

based on deformable parts model [31], pictorial structures [10, 25, 127], or poselets [13] train the relationship between body appearance and body joints using hand-crafted features. Ferrari et al. [32] improved pose estimation performance by reducing the search space. Johnson and Everingham [50] extended pictorial structure models to incorporate richer representation and prior knowledge of human poses. Various improvements such as occlusion-sensitive models [98] and cascaded models [95] further increase the pose estimation accuracy. Nevertheless, those hand-crafted or gradient-based features are inferior to neural network based methods which automatically learn informative features.

Recently proposed CNN-based methods drastically improve the pose estimation accuracy from the previous hand-crafted feature based methods. DeepPose [109] used CNN-based structure to regress joint locations with multiple iterations. It predicts an initial pose using holistic view and refine the currently predicted pose using relevant parts of the image. Xiaochuan et al. [27] integrated both the local part appearance and the holistic view of an image using dual-source CNN. Convolutional pose machine [116] proposed a systematic approach to improve prediction of each stage. By feeding the estimated heatmaps of all joints to the next stage, the refined results are obtained as the data passes through multiple stages. Carreira et al. [17] proposed a self-correcting method by a top-down feedback. It iteratively learns a human pose using a self-correcting CNN model which gradually improves the initial result by feeding back error predictions. Chu et al. [23] proposed an end-to-end learning system which captures the relationships among feature maps of joints. Geometrical transform kernels are introduced to learn features and their relationship jointly. Stacked hourglass network [76] is the structure that achieves state-of-the-art performance on 2D HPE. The network consists of multiple downsampling and

upsampling steps to learn features from multiple resolutions.

Different from the single-person pose estimation methods mentioned above, multi-person pose estimation aims to estimate poses of multiple people appeared in an image. Top-down approaches [28, 121] firstly finds bounding box of the people using object detectors, and then estimate pose of each person. Bottom-up approaches [16, 87, 75] detect candidate joints and connect them to build poses of multiple people. We only focused on single-person 3D pose estimation in this dissertation. The proposed methods can be easily extended to the multi-person scenario when combined with top-down multi-person pose estimation algorithms.

2.2 3D Human Pose Estimation

In this section, we will review supervised learning based methods that exploited 3D ground truth of human poses for training. Similar to the 2D HPE, early stage of 3D HPE is also based on the low-level features such as local shape context [1], histogram of gradients [78, 90], or segmentation results [48]. With the extracted features, 3D pose estimation is formulated as a regression problem using relevance vector machines [1], structured SVMs [48], KD-trees [128], Bayesian non-parametric models [94], or random forest classifiers [96].

Recently, CNNs have drew a lot of attentions also for the 3D human pose estimation tasks. Since search space in 3D is much larger than 2D image space, 3D human pose estimation is often formulated as a regression problem rather than a classification task. Li and Chan [64] firstly used CNNs to learn 3D human pose directly from input images. Relative 3D position to the parent joint is learned by CNNs via regression. They also used 2D part detectors of each joints

in a sliding window fashion. They found that the loss function which combines 2D joint classification and 3D joint regression helps to improve the 3D pose estimation results. Li et al. [65] improved the performance of 3D pose estimation by integrating a structured learning framework into CNNs. Tekin et al. [104] proposed a structured prediction framework which learns 3D pose representations using an auto-encoder. Temporal information from video sequences also helps to predict more accurate pose estimation result. In addition, there have been a few methods that exploit pose priors of human body from 3D data [12, 91, 58].

It has been proven that 2D pose information acts a crucial role for 3D pose estimation. Park et al. [83] directly propagated 2D pose estimation results to the 3D pose estimation part in a single CNN. Pavlakos et al. [85] proposed a volumetric representation that gradually increases the resolution of the depth from heatmaps of 2D pose. Mehta et al. [73] similarly regressed the position of each coordinate using heatmaps. There are a couple of works that directly regress 3D pose from an image using constraints on human joints [102] or combining weakly-supervised learning [130]. Tome et al. [107] lifted 2D pose heatmaps to 3D pose via probabilistic pose models. Tekin et al. [105] combined features from both RGB images and 2D pose heatmaps which were used for 3D pose estimation.

While 3D pose estimation from images have shown impressive performance, there is another approach that infers a 3D pose directly from the result of 2D pose estimation. It usually has a two-stage procedure: 1) 2D pose estimation using CNNs and 2) 3D pose inference via neural networks using the estimated 2D pose. Chen and Ramanan [19] found that a non-parametric nearest neighbor model that estimates a 3D pose from a 2D pose showed comparable performance when the precise 2D pose information is provided. Moreno-Noguer [74]

proposed a neural network that outputs 3D Euclidean distance matrices from 2D inputs. Martinez et al. [71] proposed a simple neural network that directly regresses a 3D pose from raw 2D joint positions. The network consists of two residual modules [45] with batch normalization [47] and dropout [101]. The method showed state-of-the-art performance despite its simple structure. The performance has been further improved by recent works. Fang et al. [29] proposed a pose grammar network that incorporates a set of knowledge learned from human body, which was designed as a bidirectional recurrent neural network. Yang et al. [126] used adversarial learning to implicitly learn geometric configuration of human body. Cha et al. [18] developed a consensus algorithm that generates a 3D pose from multiple partial hypotheses which are based on a non-rigid structure from motion algorithm [60]. It is found in [84] that ordinal depth supervision which gives relative depth information further improves the accuracy for 3D HPE.

There are a few approaches that exploit temporal information using various methods such as overcomplete dictionaries [131, 132], 3D CNNs [41], sequence-to-sequence networks [88], and multiple-view settings [86]. For the supervised method proposed in Chapter 5, we focus on the case that both training and testing are conducted on a single image although we also provided the extension of the proposed method to multi-frame settings.

2.3 Non-rigid Structure from Motion

Non-rigid structure from motion (NRSfM) is a research area that studies how to reconstruct 3D structures of non-rigid objects from 2D observations. The reconstruction method proposed in Chapter 3 also lies in this field. Hence, we

review the conventional NRSfM methods in this section. Existing NRSfM methods from the literature can be classified into three different approaches; utilizing shape bases, trajectory bases, or recently proposed force bases [3]. Bregler et al. [14] first solved NRSfM via factorization of a shape matrix. They suggested the assumption that each non-rigid shape is represented as a sum of weighted shape bases. Under the assumption, NRSfM could be easily solved by extending Tomasi-Kanade factorization [106]. Xiao et al. [122] argued that orthonormality constraint with fixed number of shape bases is not sufficient to obtain a unique solution. However, Akhter et al. [7] proved that the orhonormality constraint provides an unambiguous solution other than a 3×3 rotation. In [122] and [79], factorization algorithms that project the shape matrix to another subspace showed superior performance. Until now, the rank constraint is an essential concept for solving NRSfM, but choosing the number of basis K is a troublesome task since the number of shape bases is not known beforehand in most cases. Moreover, the performances are highly dependent on the number of bases. To resolve such a problem, Dai et al. [24] attempted to determine the rank of the shape matrix implicitly. They applied a rank minimization scheme rather than fixing the number of bases, which gave better results than the previous methods. Garg et al. [37] formulated NRSfM as a global variational energy minimization. The cost function they proposed contains low-rank term on a shape matrix and spatial smoothness term. Agudo et al. [2] used modal analysis from continuum mechanics to obtain shape bases. They proposed a low computational cost method which is able to process input images sequentially. Force-basis-based method [3] gives a better physical interpretation and can be considered as an alternative to the shape or trajectory bases.

Another main stream of non-rigid shape representation is trajectory basis

methods. Akhter et al. [8] argued that trajectory of each point can be represented as a weighted sum of trajectory bases, which are the dual representation of the shape bases. They verified that DCT can be used as a basis of natural motions. However, determining the number of trajectory bases is also a troublesome task. Zhu et al. [134] applied L_1 norm regularization to the DCT coefficients to obtain a sparse solution without fixing the number of trajectory bases. Fragkiadaki et al. [34] used motion trajectory clustering to segment the shape and reconstructed each cluster with different rank bounds. Agudo and Moreno-Noguer [5] proposed manifold NRSfM which is applicable to a multi-instance domain that does not require temporal consistency.

Some of prior works exploit both shape and trajectory bases. Simon et al. [99, 100] proposed a statistical model of human motion data based on the Kronecker structure of the spatiotemporal covariance of natural deformations. Li et al. [63] fit 3D shape trajectory space with wavelet basis and turned NRSfM into a matrix completion problem based on sparse representation. Gotardo and Martinez [38, 39] proposed a 3D shape trajectory approach by defining a complementary rank-3 space which is an alternative to the rank-$3K$ factorization.

Many NRSfM algorithms assume an orthographic camera to simplify the projection matrix. However, cameras in real world are usually represented by perspective projections. Orthographic projection is reasonable if a camera is far from the image plane, but errors will get significant as an object gets closer to the camera. Also, reflection ambiguities exist for the reconstructed shapes under orthographic projection. Xiao and Kanade [123] first addressed NRSfM under perspective projection. It recovers projective depths in the first step, then camera matrices are recovered based on the 3D points. Wang et al. [115] extended the orthographic camera model to a weak-perspective camera model, and the re-

constructed shapes are refined via bundle adjustment. Liado et al. [68] separated rigid and non-rigid points from deformable objects to reconstruct shapes from uncalibrated perspective camera, and Bartoli et al. [11] suggested a coarse-to-fine low rank shape model. Hartley and Vidal [43] proposed a closed-form solution using images from uncalibrated cameras, but the number of shape bases should be determined. The shape basis methods for perspective camera are sensitive to noise and work well only for the shapes with largely rigid motions such as human faces [82]. Prior knowledge such as inextensibility constraints [113] or physical priors [6] are also used for perspective NRSfM, but these methods does not fit well to the 3D models with large deformations such as human skeletons.

For perspective projections, trajectory-based approach showed more successful results. Park et al. [81, 82] developed a linear solution for reconstructing 3D trajectories when camera matrices are known. They assumed that relative locations of the view-points are known, which makes the problem easier than other NRSfM methods. Temporal smoothness assumption is often used for trajectory-based methods. Valmadre and Lucey [112] suggested trajectory filter which gives higher weights on the low-frequency bases of DCT by applying simple difference filters in the time domain. While trajectory-based methods work well in the case that the position of camera changes rapidly, they perform very poorly when there is a little motion on the camera. Also, 3D shapes cannot be reconstructed when pose and position of perspective cameras are unknown.

There have been efforts to apply shape alignment of non-rigid shapes to NRSfM algorithm. Cho et al. [20] suggested EM-GPA which exploited 3D shape alignment to reconstruct non-rigid shapes. They extended GPA to deal with missing points using EM algorithm. Lee et al. [59, 61] modified the opti-

mality condition of GPA to derive a constraint for non-rigid component, which is the first work that discovered the relation between shape alignment and separation of rigid and non-rigid components. Their method, EM-PND, showed superior performance compared to the previous NRSfM algorithms without choosing the number of bases explicitly. Procrustean Markov process (PMP) [62] further extended EM-PND by utilizing a hidden Markov process which imposes temporal dependency to the distribution of the non-rigid component. A PND mixture model [21], PNDMM, which probabilistically models the 2D shape generation process from a mixture of 3D shapes is proposed recently to deal with sequences that have complex temporal variations.

The superiority of the aligned shapes over the conventional NRSfM methods in terms of performance have been proved in the PND and its variants. Hence, the idea of shape alignment is also the underlying foundation of the proposed framework. On the other hand, the drawbacks of EM-PND, PMP, and PNDMM are their lack of flexibility and heavy, complicated designs, which restricts high resolution inputs. The memory consumption, as well as the time complexity, for the EM-algorithm grows quadratically with the number of points. Moreover, it is hard to impose additional constraints or extend the framework to perspective camera cases since EM-PND is strictly designed to meet the PND constraint under orthographic projection. In chapter 3, we propose a novel NRSfM framework that gives flexibility and efficiency without losing the advantage of rigid/non-rigid separation in PND [59, 61].

2.4 Learning to Reconstruct 3D Structures via Neural Networks

The weakly-supervised learning proposed in Chapter 4 provides the method to train neural network to reconstruct 3D human pose from 2D inputs. In this section, we review the existing methods that learn to infer 3D structures from 2D inputs via neural networks. Although our weakly-supervised method only uses 2D ground truth, the methods that exploits 3D ground truth data are also covered in this section. Along with recent improvements in deep learning, there have been efforts that solve 3D reconstruction using CNNs. Although there were a few data-driven approaches for 3D reconstruction of objects without CNNs [114, 56], CNNs can be used as a powerful tool for object reconstruction in terms of its performance and generalization ability. CNNs that generate depth maps from scene images [26, 33, 36, 124] require stereo image pairs or ground truth depth maps for training.

Object reconstruction from a single image with CNNs is an active field of research. The densely reconstructed shapes are often represented as 3D voxels or depth maps. While some works use ground truth 3D shapes [103, 22, 118], other works enable the networks to learn 3D reconstruction from multiple 2D observations [125, 111, 35, 129]. The networks used in aforementioned works include a transformation layer that estimates the viewpoint of observations and/or a reprojection layer to minimize the error between input images and projected images. However, they also deal with the classes that are rigid and have small amount of deformations within each class such as chairs and tables.

The 3D interpreter network [119] took similar approach to NRSfM methods in that it formulates 3D shapes as the weighted sum of base shapes, but it

used 3D synthetic models for network training. Warpnet [54] successfully reconstructs 3D shapes of non-rigid objects without supervision, but the results are only provided for birds datasets which has less deformations than human skeletons. Tulsiani et al. [110] provided the learning algorithm that automatically localize and reconstruct deformable 3D objects, and Kanazawa et al. [55] also infers 3D shapes as well as texture information from a single image. Although those methods outputs dense 3D meshes, the reconstruction is conducted on rigid objects or birds which does not contain large deformations. The framework proposed in chapter 4 provides a way to learn 3D structure of human skeleton which contains large deformations and pose variations.

Chapter 3

3D Reconstruction of Human Bodies via Procrustean Regression

3D reconstruction has been a fundamental research area since early computer vision literatures [106, 44]. Although 3D reconstruction of rigid objects has been studied thoroughly, recovering 3D shapes of non-rigid shapes remains as a challenging problem. NRSfM has been long considered to be an underconstrained problem, so additional constraints have been imposed to find a solution. Most existing algorithms impose rank constraint on the shape matrix. The underlying assumption for the rank constraint is that non-rigid shape is represented as a weighted sum of a few basis shapes [14]. Number of shape basis, or rank of a shape matrix in other words, was fixed in the early research of NRSfM. However, fixing the number of shape basis is not applicable for real world situations.

On the other hand, Akhter et al. [8] revealed the duality between the shape and trajectory bases, using discrete cosine transform (DCT) as a trajectory basis to reconstruct 3D trajectories of deformable objects. However, similar to the shape basis representation, the number of the trajectory bases is also difficult

to choose. To alleviate this problem, algorithms based on rank minimization scheme have been proposed [24, 37]. These algorithms which minimize rank of the shapes matrix implicitly via minimization of a nuclear norm showed superior performance to the ones that fix the rank of shape matrix.

Recently, Procrustean Normal Distribution (PND) [59, 61] was proposed to impose a probabilistic model only on the non-rigid components of shape variations. PND used modified version of generalized procrustes analysis (GPA) to align 3D non-rigid shapes, and then rigid components are separated from non-rigid components. An expectation-maximization (EM) algorithm was used to calculate the distribution of 3D shapes. Although PND suggested the importance of the shape alignment, the actual reconstruction method, EM-PND, lacks flexibility and requires heavy computation and large memory.

In order to maximize flexibility and practicality while maintaining the good effect of shape alignment, in this chapter, we propose a novel regression framework for NRSfM. In the regression framework, NRSfM is formulated as a least squares problem with a regularization term. Unlike the conventional methods which impose regularization constraints on the reconstructed 3D shapes directly [14, 24], the regularization term in our framework is on the aligned shapes which is inspired by PND. The data term is on the relation between 3D shapes in a camera coordinate system and the corresponding projected 2D observations. We show that the proposed cost function can be easily casted into an unconstrained problem or a simple bound-constrained problem without any relaxation or approximation step. The proposed method has much less time and memory complexities than EM-PND, by a factor of number of points in the data. Moreover, unlike EM-PND, many different error models can be easily integrated within our framework.

The rest of this chapter is organized as follows: We formally define the problem of NRSfM in Section 3.1, and the proposed framework is explained in Section 3.2. Then, experimental results on multiple datasets with various formulations are provided in Section 3.3. Additional discussions and conclusions of the proposed method is given in Section 3.4 and Section 3.5 respectively.

3.1 Formalization of NRSfM

NRSfM aims to recover 3D positions of the deformable and moving objects from 2D correspondences. Concretely, 2D observations of multiple frames are given as an input. Let n_p the number of points, and n_f the number of frames in the input. We denote $\mathbf{U}_i (1 \leq i \leq n_f)$ as the 2D observation in ith frame. $\mathbf{U}_i$ is a $2 \times n_p$ matrix which has the form of

$$\mathbf{U}_i = \begin{bmatrix} u_{i1} & u_{i2} & \cdots & u_{in_p} \\ v_{i1} & v_{i2} & \cdots & v_{in_p} \end{bmatrix}, \tag{3.1}$$

where $[u_{ij}, v_{ij}]^T$ is a 2D point of jth point in ith frame. NRSfM algorithm should reconstruct 3D shapes of all n_f frames from 2D observations. We denote $\mathbf{X}_i$ as the matrix of reconstructed 3D shapes in ith frame, which has the form of

$$\mathbf{X}_i = \begin{bmatrix} x_{i1} & x_{i2} & \cdots & x_{in_p} \\ y_{i1} & y_{i2} & \cdots & y_{in_p} \\ z_{i1} & z_{i2} & \cdots & z_{in_p} \end{bmatrix}. \tag{3.2}$$

$\mathbf{X}_i$ is a $3 \times n_p$ matrix and $[x_{ij}, y_{ij}, z_{ij}]^T$ is a 3D point of jth point in ith frame.

Many NRSfM algorithms assumes simple camera models such as orthographic projection. The algorithm proposed in this chapter deals with both orthographic and perspective projections. In the case of perspective projection, we

assume that intrinsic camera matrix is known. Though some NRSfM algorithms estimates camera parameters in addition to the 3D shapes of objects, we only focus on the reconstruction performance of 3D shapes.

NRSfM algorithms does not target specific object classes, but it is applicable to any non-rigid object classes. Hence, NRSfM algorithms does not require pre-learning steps such as template or dictionary learning about targeted classes. Human bodies are one of the well known non-rigid objects. Sequence of human actions contains large deformations and pose variations as well as the movements. We mainly focus on the reconstruction of 3D human pose from 2D poses or images in this dissertation.

3.2 Procrustean Regression

A novel regression framework for NRSfM is illustrated in this section. One of the strong points of the proposed framework is that various form of data term and regularization term can be easily integrated. Hence, the framework provides a general and flexible method for NRSfM.

Overall process of the proposed framework is illustrated in Fig. 3.1. The reconstructed 3D shapes and the reference shape is updated by minimizing the proposed cost function. We emphasize that the aligned 3D shapes are obtained automatically within the optimization framework since the cost function is updated considering the changes in alignment. The regularization term is able to incorporate various types of regularizer. For instance, the regularizer is able to impose not only a low-rank constraint on aligned shapes but also temporal dependency of points as in [8, 62]. Hence, the proposed framework is flexible in that various models can be directly integrated to the framework. For

the data term, the proposed method is able to cover both orthographic and perspective camera models. Unlike previous works of trajectory-based perspective NRSfM [81, 112], our framework is able to reconstruct 3D shapes when relative camera motion is unknown, i.e., extrinsic parameters of the camera are not known.

In section 3.2.1, we suggest a general form of the cost function for NRSfM that is basically a regularized regression. In section 3.2.2, it is explained how the cost function is casted into an unconstrained or bound-constrained problem. We verify that gradients can be analytically derived in a closed-form, which enables us to utilize any gradient-based solver. In section 3.2.3, examples of various forms of data term, regularization term, and their gradients are illustrated. Section 3.2.4 provides a way to handle the case when missing points exist in the inputs. Section 3.2.5 provides a way to handle the case when missing points exist in the inputs. Section 3.2.6 explains the initialization algorithm that determines starting values from the inputs.

3.2.1 The Cost Function of Procrustean Regression

Under the assumption that non-rigid shape can be represented as a linear combination of a small number of basis shapes or trajectories, the rank of a shape matrix or trajectory matrix is often used as a constraint in NRSfM algorithms [14, 8]. We followed the idea of shape alignment suggested in [59] rather than using unaligned shapes. As a consequence, our cost function consists of a data term which depends on the unaligned 3D shapes and the regularization term which depends on the aligned 3D shapes. Therefore, solving NRSfM is inter-

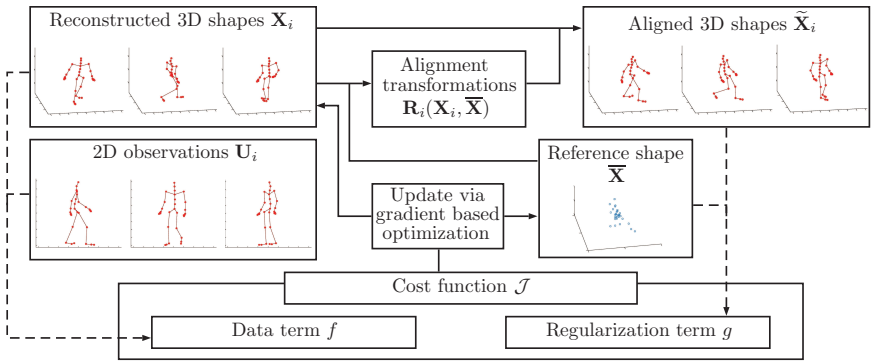


Figure 3.1: Graphical illustration of the proposed framework. The cost function $\mathcal{J}$ (equation (3.3) in Section 3.2.1) consists of a data term f and a regularization term g. The data term depends on the reconstructed 3D shape $\mathbf{X}_i$ while the regularization depends on its aligned shape $\widetilde{\mathbf{X}}_i$. The alignment $\mathbf{R}_i$ depends on $\mathbf{X}_i$ and the reference shape $\overline{\mathbf{X}}$ (equation (3.4) in Section 3.2.1). The proposed method can update this cost function considering the change in alignment (equation (3.16) in Section 3.2.2), based on existing gradient-based solvers.

preted as minimizing the following form of objective function:

$$\mathcal{J} = \sum_{i=1}^{n_f} f(\mathbf{X}_i, \mathbf{h}_i) + \lambda g(\widetilde{\mathbf{X}}, \overline{\mathbf{X}}). \tag{3.3}$$

Function f is the data term which aims to minimize the error between projected 2D points from recovered 3D shape and the points from input images. $\mathbf{X}_i$ is a $3 \times n_p$ matrix that represents 3D shapes on the ith frame, $\mathbf{h}_i$ is an additional parameter vector that are used in the function f. One example of $\mathbf{h}_i$ can be projective depths. n_p and n_f are the number of points in each frame and the number of frames, respectively. λ is a weight parameter that balances the two terms, f and g. Function g is the regularization term which imposes additional prior knowledge on the aligned 3D shapes, such as low-rankness. Especially,

in the proposed framework, regularization is imposed on the aligned shapes so that the prior knowledge can be applied only to the non-rigid components. $\widetilde{\mathbf{X}}$ is a $3n_p \times n_f$ matrix defined as $\widetilde{\mathbf{X}} \triangleq [\mathrm{vec}(\widetilde{\mathbf{X}}_1)\, \mathrm{vec}(\widetilde{\mathbf{X}}_2) \cdots \mathrm{vec}(\widetilde{\mathbf{X}}_{n_f})]$, where $\mathrm{vec}(\cdot)$ is a vectorization operator, and $\widetilde{\mathbf{X}}_i$ is an aligned shape of the ith frame. $\overline{\mathbf{X}}$ is the reference shape for alignment. Commonly used method to align non-rigid shapes is Procrustes analysis or Generalized Procrustes Analysis (GPA) [40]. Accordingly in this chapter, the alignment transform $\mathbf{R}_i$, which is a 3×3 rotation matrix, is determined based on $\mathbf{X}_i$ and $\overline{\mathbf{X}}$:

$$\mathbf{R}_i = \underset{\mathbf{R'}_i}{\operatorname{argmin}} \|\mathbf{R'}_i \mathbf{X}_i \mathbf{T} - \overline{\mathbf{X}}\| \quad \text{s.t.} \quad \mathbf{R'}_i^T \mathbf{R'}_i = \mathbf{I}, \tag{3.4}$$

where $\mathbf{T} = \mathbf{I} - \frac{1}{n_p}\mathbf{1}\mathbf{1}^T$ is a matrix that removes the translation component. Based on (3.4), the aligned shape is defined as $\widetilde{\mathbf{X}}_i \triangleq \mathbf{R}_i \mathbf{X}_i \mathbf{T}$. The rotation of a non-rigid shape is often incorporated as an extrinsic parameter in the conventional shape-basis NRSfM methods. The rotations in existing schemes do not strictly satisfy (3.4). The advantages of formulating rotation matrix as an alignment matrix rather than a free parameter are well explained in [61]. Note that, in this formulation, the scale constraint in GPA is not considered for alignment. The reason for the above alignment formulation, which does not contain a scale adjustment, will be explained later. Note that because of (3.4), this problem becomes a complex problem with nonlinear constraints.

Nevertheless, we show that the gradient of the proposed function can be efficiently calculated, which turns the problem into an unconstrained or bound-constrained optimization. Fig. 3.1 describes the proposed framework. In this dissertation, only examples of differentiable f and g are used for simplicity. However, the proposed framework can be easily extended to non-smooth functions if appropriate solvers are used. In the following section, it will be explained

how the shape alignment constraint is integrated into the cost function to exploit conventional solvers.

3.2.2 Derivatives of the Cost Function

An unconstrained optimization framework, such as BFGS, cannot be applied directly to the cost function (3.3) since the solution should also satisfy the constraint (3.4). To cast the proposed formulation into an unconstrained optimization, the constraint needs to be integrated into the cost function. First, we assume that f is designed so that $\mathbf{h}_i$ can be determined in a closed-form once $\mathbf{X}_i$ is fixed, i.e.,

$$\mathbf{h}_i = \underset{\mathbf{h}}{\operatorname{argmin}} f(\mathbf{X}_i, \mathbf{h}). \tag{3.5}$$

In other words, $\mathbf{h}_i$ can be represented as the function of $\mathbf{X}_i$. Examples for f are suggested in Section 3.2.3. Second, parameters for the aligned shapes such as $\mathbf{R}_i$ and $\widetilde{\mathbf{X}}$ can be calculated once $\mathbf{X}_i$ and $\overline{\mathbf{X}}$ are determined. Hence, (3.3) can be reformulated as a function of two variables, $\mathbf{X}_i$ and $\overline{\mathbf{X}}$.

To utilize an existing gradient-based solver, gradient of $\mathcal{J}$ with respect to (w.r.t.) $\mathbf{X}_i$ and $\overline{\mathbf{X}}$ is needed. By the chain rule, $\partial\mathcal{J}/\partial\overline{\mathbf{X}}$ is expressed as

$$\frac{\partial \mathcal{J}}{\partial \overline{\mathbf{X}}} = \lambda\left(\frac{\partial g}{\partial \overline{\mathbf{X}}} + \sum_i \left\langle \frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}, \frac{\partial \widetilde{\mathbf{X}}_i}{\partial \overline{\mathbf{X}}} \right\rangle\right), \tag{3.6}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product, and differentiation w.r.t. $\mathbf{X}_i$ yields

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \mathbf{X}_i} &= \frac{\partial f}{\partial \mathbf{X}_i} + \left\langle \frac{\partial f}{\partial \mathbf{h}_i}, \frac{\partial \mathbf{h}_i}{\mathbf{X}_i} \right\rangle + \lambda \left\langle \frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}, \frac{\partial \widetilde{\mathbf{X}}_i}{\partial \mathbf{X}_i} \right\rangle \\ &= \frac{\partial f}{\partial \mathbf{X}_i} + \lambda \left\langle \frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}, \frac{\partial \widetilde{\mathbf{X}}_i}{\partial \mathbf{X}_i} \right\rangle, \end{aligned} \tag{3.7}$$

since $\partial f/\partial \mathbf{h}_i = 0$ due to (3.5). $\frac{\partial g}{\partial \overline{\mathbf{X}}}$, $\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}$, and $\frac{\partial f}{\partial \mathbf{X}_i}$ can be analytically derived once f and g are defined. Hence, we need analytic derivations of $\frac{\partial \widetilde{\mathbf{X}}_i}{\partial \overline{\mathbf{X}}}$ and $\frac{\partial \widetilde{\mathbf{X}}_i}{\partial \mathbf{X}_i}$.

To integrate the alignment constraint in (3.4) to the cost function, let us introduce an orthogonal matrix $\mathbf{Q}_i$ which satisfies $\mathbf{R}_i = \mathbf{Q}_i\hat{\mathbf{R}}_i$. $\mathbf{Q}_i$ means a relative change of the aligning rotation from previous state $\hat{\mathbf{R}}_i$ to current state $\mathbf{R}_i$. We can assume that $\mathbf{Q}_i = \mathbf{I}$ at the time of gradient evaluation without loss of generality. Then, differentiating the orthogonality condition $\mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I}$ and evaluating at $\mathbf{Q}_i = \mathbf{I}$ yields

$$\partial\mathbf{Q}_i^T\mathbf{Q}_i + \mathbf{Q}_i^T\partial\mathbf{Q}_i = \partial\mathbf{Q}_i^T + \partial\mathbf{Q}_i = 0. \tag{3.8}$$

Hence, $\partial\mathbf{Q}_i$, which can be interpreted as an infinitesimal generator of rotation, is a skew-symmetric matrix. Note that the derivation is the Lie algebra of the rotation group, $\mathrm{SO}(3)$ [42]. Let us denote $\partial\mathbf{Q}_i$ as

$$\partial\mathbf{Q}_i = \begin{bmatrix} 0 & \partial q_{iz} & -\partial q_{iy} \\ -\partial q_{iz} & 0 & \partial q_{ix} \\ \partial q_{iy} & -\partial q_{ix} & 0 \end{bmatrix}, \tag{3.9}$$

and $\partial\mathbf{q}_i = [\partial q_{ix} \quad \partial q_{iy} \quad \partial q_{iz}]^T$. Then $\mathrm{vec}(\partial\mathbf{Q}_i) = \mathbf{L}\partial\mathbf{q}_i$ holds where

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{3.10}$$

Given an arbitrary $3 \times n_p$ matrix $\mathbf{A}$ and its column vectors $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{n_p}$, one can easily verify that $(\mathbf{A}^T \otimes \mathbf{I}_3)\mathbf{L} = \begin{bmatrix}[\mathbf{a}_1]_\times^T & [\mathbf{a}_2]_\times^T \cdots [\mathbf{a}_{n_p}]_\times^T\end{bmatrix}^T$ where $\otimes$ denotes the Kronecker product, $\mathbf{I}_3$ is a 3×3 identity matrix, and $[\mathbf{a}]_\times$ is a skew-symmetric matrix that is related to cross product of the vector. Let us denote $(\mathbf{A}^T \otimes \mathbf{I}_3)\mathbf{L}$ as $C(\mathbf{A})$. Note that $C(\mathbf{A})$ is the same as the second to fourth columns of the PND constraint proposed in [59].

Also, from (3.4) and from the condition $\mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I}$, the following equation is derived:

$$\partial(\frac{1}{2}\|\mathbf{Q}_i\hat{\mathbf{R}}_i\mathbf{X}_i\mathbf{T} - \overline{\mathbf{X}}\|^2 + \frac{1}{2}\langle\mathbf{\Lambda}_i, \mathbf{Q}_i^T\mathbf{Q}_i - \mathbf{I}\rangle) = 0, \tag{3.11}$$

where $\mathbf{\Lambda}_i$ is a lagrange multiplier. Let us denote $\hat{\mathbf{R}}_i\mathbf{X}_i\mathbf{T}$ as $\mathbf{X}_i'$, then (3.11) becomes

$$(\mathbf{Q}_i\mathbf{X}_i' - \overline{\mathbf{X}})\mathbf{X}_i'^T + \mathbf{Q}_i\mathbf{\Lambda}_i = \mathbf{0}. \tag{3.12}$$

Rearranging (3.12) and multiplying $\mathbf{Q}_i^T$ on both sides yields

$$\mathbf{Q}_i(\mathbf{X}_i'\mathbf{X}_i'^T + \mathbf{\Lambda}_i)\mathbf{Q}_i^T = \overline{\mathbf{X}}\mathbf{X}_i'^T\mathbf{Q}_i^T. \tag{3.13}$$

Here, $\mathbf{\Lambda}_i$ is symmetric because the constraint $\mathbf{Q}_i^T\mathbf{Q}_i = \mathbf{I}$ is symmetric. Therefore, one can verify that left hand side is a symmetric matrix and also the right hand side, i.e.,

$$\overline{\mathbf{X}}\mathbf{X}_i'^T\mathbf{Q}_i^T = \mathbf{Q}_i\mathbf{X}_i'\overline{\mathbf{X}}^T. \tag{3.14}$$

Vectorizing (3.14) yields

$$\begin{aligned}&\mathrm{vec}(\mathbf{Q}_i\mathbf{X}_i'\overline{\mathbf{X}}^T - \overline{\mathbf{X}}\mathbf{X}_i'^T\mathbf{Q}_i^T)\\ &= [(\overline{\mathbf{X}}\otimes\mathbf{I}) - (\mathbf{I}\otimes\overline{\mathbf{X}})\mathbf{E}]\mathrm{vec}(\mathbf{Q}_i\mathbf{X}_i') = \mathbf{0},\end{aligned} \tag{3.15}$$

where $\mathbf{E}$ is a permutation matrix which satisfies $\mathbf{E}\mathrm{vec}(\mathbf{A}) = \mathrm{vec}(\mathbf{A}^T)$. Note that (3.15) is a degenerate equation. $C(\overline{\mathbf{X}})^T$ can be directly obtained from $[(\overline{\mathbf{X}}\otimes\mathbf{I}) - (\mathbf{I}\otimes\overline{\mathbf{X}})\mathbf{E}]$. There are only three independent rows in this matrix, i.e., many are trivial or identical. The 8th, 3rd, 4th rows of $[(\overline{\mathbf{X}}\otimes\mathbf{I}) - (\mathbf{I}\otimes\overline{\mathbf{X}})\mathbf{E}]$ are equal to the 1st, 2nd, 3rd rows of $C(\overline{\mathbf{X}})^T$, respectively. Hence, the following condition is derived.

$$C(\overline{\mathbf{X}})^T\mathrm{vec}(\mathbf{Q}_i\mathbf{X}_i') = \mathbf{L}^T\mathrm{vec}(\mathbf{Q}_i\mathbf{X}_i'\overline{\mathbf{X}}^T) = \mathbf{0}. \tag{3.16}$$

The condition (3.16) implies the relationship between the reference shape $\overline{\mathbf{X}}$

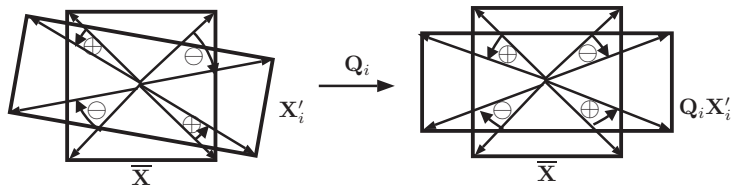


Figure 3.2: Geometric interpretation of the condition (3.16). Any misalignment caused by the change of $\mathbf{X}'_i$ is automatically adjusted by the rotation matrix $\mathbf{Q}_i$ because (3.16) makes the sum of the vectors from cross products zero.

and the aligned shape $\widetilde{\mathbf{X}}_i = \mathbf{Q}_i\mathbf{X}'_i$. The geometric interpretation of (3.16) is illustrated in Fig. 3.2. We illustrated the case that $\mathbf{X}'_i$ and $\overline{\mathbf{X}}$ are laid on the same plane for ease of explanation. The cross product of each point correspondence between $\mathbf{X}'_i$ and $\overline{\mathbf{X}}$ results the vector either pointing out of the page (denoted as $\oplus$) or pointing into the page (denoted as $\ominus$). Because of the constraint (3.16), the rotation $\mathbf{Q}_i$ has to be determined so that the sum of the magnitude of pointing-out vectors and pointing-in vectors are equal, which compensates any misalignment caused by the change in $\mathbf{X}'_i$.

Now, it is able to express $\partial\mathbf{q}_i$ in terms of $\mathbf{X}'_i$ and $\overline{\mathbf{X}}$. Differentiating (3.16) and evaluating at $\mathbf{Q}_i = \mathbf{I}$ yields

$$\begin{aligned}
&\mathbf{L}^T\mathrm{vec}(\partial\mathbf{Q}_i\mathbf{X}'_i\overline{\mathbf{X}}^T + \mathbf{Q}_i\partial\mathbf{X}'_i\overline{\mathbf{X}}^T + \mathbf{Q}_i\mathbf{X}'_i\partial\overline{\mathbf{X}}^T) \\
&= \mathbf{L}^T\mathrm{vec}(\partial\mathbf{Q}_i\mathbf{X}'_i\overline{\mathbf{X}}^T + \partial\mathbf{X}'_i\overline{\mathbf{X}}^T + \mathbf{X}'_i\partial\overline{\mathbf{X}}^T) \\
&= \mathbf{L}^T[(\overline{\mathbf{X}}\mathbf{X}_i'^T \otimes \mathbf{I})\mathrm{vec}(\partial\mathbf{Q}_i) + \mathrm{vec}(\partial\mathbf{X}'_i\overline{\mathbf{X}}^T + \mathbf{X}'_i\partial\overline{\mathbf{X}}^T)] = \mathbf{0}.
\end{aligned} \tag{3.17}$$

Note that the property of the Kronecker product $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^T\otimes\mathbf{A})\mathrm{vec}(\mathbf{B})$ is used here. Then, $\mathbf{q}_i$ is calculated as

$$\partial\mathbf{q}_i = -\mathbf{\Psi}_i^{-1}\mathbf{L}^T\mathrm{vec}(\partial\mathbf{X}'_i\overline{\mathbf{X}}^T + \mathbf{X}'_i\partial\overline{\mathbf{X}}^T), \tag{3.18}$$

where $\mathbf{\Psi}_i = \mathbf{L}^T(\overline{\mathbf{X}}\mathbf{X}_i'^T \otimes \mathbf{I})\mathbf{L} = \mathrm{tr}(\overline{\mathbf{X}}\mathbf{X}_i'^T)\mathbf{I} - \overline{\mathbf{X}}\mathbf{X}_i'^T$. Also, differentiating

$\mathbf{Q}_i\mathbf{X}_i' = \widetilde{\mathbf{X}}_i$ yields

$$\partial\mathbf{Q}_i\mathbf{X}_i' + \mathbf{Q}_i\partial\mathbf{X}_i' = \partial\widetilde{\mathbf{X}}_i. \tag{3.19}$$

From (3.18) and (3.19), we are able to derive the gradients we need. By vectorizing (3.19), we get

$$(\mathbf{X}_i'^T \otimes \mathbf{I})\mathbf{L}\partial\mathbf{q}_i = \mathrm{vec}(\partial\widetilde{\mathbf{X}}_i - \partial\mathbf{X}_i'). \tag{3.20}$$

By substituting (3.18) to (3.20), we get

$$\begin{aligned}
&\mathrm{vec}(\partial\widetilde{\mathbf{X}}_i) - \mathrm{vec}(\partial\mathbf{X}_i') \\
&= -C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}\mathbf{L}^T\mathrm{vec}(\partial\mathbf{X}_i'\overline{\mathbf{X}}^T + \mathbf{X}_i'\partial\overline{\mathbf{X}}^T) \\
&= -C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}\mathbf{L}^T[(\overline{\mathbf{X}} \otimes \mathbf{I})\mathrm{vec}(\partial\mathbf{X}_i') + (\mathbf{I} \otimes \mathbf{X}_i')\mathbf{E}\mathrm{vec}(\partial\overline{\mathbf{X}})].
\end{aligned} \tag{3.21}$$

Dividing both sides of (3.21) by $\partial\mathrm{vec}(\overline{\mathbf{X}})$ yields

$$\begin{aligned}
\frac{\mathrm{vec}(\partial\widetilde{\mathbf{X}}_i)}{\partial\mathrm{vec}(\overline{\mathbf{X}})} &= -C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}\mathbf{L}^T(\mathbf{I} \otimes \mathbf{X}_i')\mathbf{E} \\
&= C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}C(\mathbf{X}_i')^T,
\end{aligned} \tag{3.22}$$

and dividing both sides of (3.21) by $\partial\mathrm{vec}(\mathbf{X}_i)$ yields

$$\begin{aligned}
\frac{\mathrm{vec}(\partial\widetilde{\mathbf{X}}_i)}{\partial\mathrm{vec}(\mathbf{X}_i)} &= (\mathbf{T} \otimes \hat{\mathbf{R}}_i) - C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}C(\overline{\mathbf{X}})^T(\mathbf{T} \otimes \hat{\mathbf{R}}_i) \\
&= (\mathbf{I} - C(\mathbf{X}_i')\mathbf{\Psi}_i^{-1}C(\overline{\mathbf{X}})^T)(\mathbf{T} \otimes \hat{\mathbf{R}}_i).
\end{aligned} \tag{3.23}$$

Note that $\mathrm{vec}(\mathbf{X}_i') = (\mathbf{T} \otimes \hat{\mathbf{R}}_i)\mathrm{vec}(\mathbf{X}_i)$. Substituting (3.22) and (3.23) to (3.6)

and (3.7) yields the following equations.

$$\begin{aligned}
&\mathrm{vec}(\frac{\partial \mathcal{J}}{\partial \overline{\widetilde{\mathbf{X}}}}) = \lambda(\frac{\partial g}{\partial \overline{\widetilde{\mathbf{X}}}} + \sum_i C(\mathbf{X}'_i)\mathbf{\Psi}_i^{-T} C(\mathbf{X}'_i)^T \mathrm{vec}(\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i})) \\
&= \lambda(\frac{\partial g}{\partial \overline{\widetilde{\mathbf{X}}}} + \sum_i C(\mathbf{X}'_i)\mathbf{\Psi}_i^{-T} \mathbf{L}^T \mathrm{vec}(\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i} \mathbf{X}_i'^T)) \\
&\mathrm{vec}(\frac{\partial \mathcal{J}}{\partial \mathbf{X}_i}) \\
&= \mathrm{vec}(\frac{\partial f}{\partial \mathbf{X}_i}) + (\mathbf{T} \otimes \hat{\mathbf{R}}_i^T)(\mathbf{I} - C(\overline{\mathbf{X}})\mathbf{\Psi}_i^{-T} C(\mathbf{X}'_i)^T) \mathrm{vec}(\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}) \\
&= \mathrm{vec}(\frac{\partial f}{\partial \mathbf{X}_i}) + (\mathbf{T} \otimes \hat{\mathbf{R}}_i^T)(\mathrm{vec}(\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}) - C(\overline{\mathbf{X}})\mathbf{\Psi}_i^{-T} \mathbf{L}^T \mathrm{vec}(\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i} \mathbf{X}_i'^T).
\end{aligned} \tag{3.24}$$

Note that $C(\mathbf{A})^T \mathrm{vec}(\mathbf{B}) = \mathbf{L}^T \mathrm{vec}(\mathbf{B}\mathbf{A}^T)$. $\frac{\partial g}{\partial \overline{\widetilde{\mathbf{X}}}}$, $\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}$, and $\frac{\partial f}{\partial \mathbf{X}_i}$ are derived based on how f and g are formed. The regularization terms proposed in this dissertation only depend on $\widetilde{\mathbf{X}}_i$, in which $\frac{\partial g}{\partial \overline{\widetilde{\mathbf{X}}}}$ becomes zero. Based on (3.24), which are simple analytic expressions, we can easily minimize our cost function based on existing solvers. Note that the gradient (3.24) implicitly considers the alignment process without explicitly handling $\mathbf{R}_i$. If the actual values of $\mathbf{R}_i$ (based on $\mathbf{X}_i$ and $\overline{\mathbf{X}}$) have to be computed, then it can be computed as:

$$\mathbf{R}_i = \mathbf{U}\mathbf{V}^T, \tag{3.25}$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition of $\overline{\mathbf{X}}\mathbf{X}_i^T$.

Since the proposed framework considers shape alignment similar to that in [59], one may seek the possibility of using the exact same alignment used in [59]. However, our formulation in (3.4) does not have a scale adjustment, which is to avoid a trivial solution. If we introduce the scale component s_i, alignment formulation becomes

$$\begin{aligned}
(s_i, \mathbf{R}_i) = \underset{s_i, \mathbf{R}_i}{\mathrm{argmin}} \, &\|s_i \mathbf{R}_i \mathbf{X}_i \mathbf{T} - \overline{\mathbf{X}}\|, \\
\text{s.t.} \quad &\mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}, \langle s_i \mathbf{R}_i \mathbf{X}_i \mathbf{T}, \overline{\mathbf{X}} \rangle = \|\overline{\mathbf{X}}\|^2.
\end{aligned} \tag{3.26}$$

The last condition is the scale constraint. If the regularization term g is a function of $\mathbf{Y}_i = \widetilde{\mathbf{X}}_i - \overline{\mathbf{X}}$, then using the alignment in (3.26) will exhibit a trivial solution. The trivial solution occurs when the scale parameter is infinitely small and the depth of some points become infinitely large. For instance, under orthographic projection, given 2D observation on ith frame $\mathbf{W}_i$, trivial solution is given as $\mathbf{X}_i = \begin{bmatrix} \mathbf{W}_i \\ \mathbf{v} \end{bmatrix}$, $s_i \rightarrow 0$, $s_i\|\mathbf{v}\| \rightarrow 1$, $\overline{\mathbf{X}} = \begin{bmatrix} \mathbf{0} \\ \frac{\mathbf{v}}{\|\mathbf{v}\|} \end{bmatrix}$, and $\mathbf{R}_i = \mathbf{I}$. This has been overlooked in [61], where $\mathbf{X}_i$ is a distribution rather than a deterministic variable, and this problem has not arisen explicitly.

This can be avoided by eliminating the scale term. To cope with scale difference between frames, 2D input points are normalized under the orthographic projection. For the case of perspective projection, large values of projective depth are prohibited by the regularization term because it grows as the projective depth gets larger, which will be explained in Section 3.2.3. However, projective depth can still have arbitrarily small value, which causes similar effect as $s_i \rightarrow 0$ in the orthographic case. Therefore, we impose a bound on the projective depth as a constraint.

3.2.3 Example Functions for f and g

Most representative examples of the data term and the regularization term are suggested in this section. After f and g are appropriately defined, $\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}$, $\frac{\partial f}{\partial \mathbf{X}_i}$ are derived and substituted into (3.24). Besides the functions suggested in this section, new ideas for the data term or regularization term can be easily integrated in the proposed framework.

Data term

Function f plays a role of penalizing the error between observed 2D points and recovered 3D shapes. Both orthographic and perspective camera can be incorporated into the function f. Orthographic projection matrix is defined as $\mathbf{P}_{ortho} = \mathrm{diag}([1, 1, 0]^T)$, where $\mathrm{diag}(\mathbf{x})$ is a diagonal matrix whose diagonal elements are $\mathbf{x}$. Then, the reprojection error function f_{ortho} is defined as

$$f_{ortho}(\mathbf{X}_i) = \frac{1}{2}\|\mathbf{U}_i - \mathbf{P}_{ortho}\mathbf{X}_i\|_F^2, \tag{3.27}$$

where $\mathbf{U}_i$ is the 2D observation of the ith frame whose dimension is $3 \times n_p$ and the third row is zero. Differentiating w.r.t. $\mathbf{X}_i$ yields

$$\frac{\partial f_{ortho}}{\partial \mathbf{X}_i} = \mathbf{P}_{ortho}^T(\mathbf{P}_{ortho}\mathbf{X}_i - \mathbf{U}_i) = \mathbf{P}_{ortho}(\mathbf{X}_i - \mathbf{U}_i). \tag{3.28}$$

The perspective camera case is similar to the orthographic case. In this chapter, we assume that the intrinsic matrix of camera is known and 3D shapes are recovered up to scale for each shape. We can consider the projective depth as a parameter which is multiplied to 2D observations. There are a couple of possible cost functions that utilize projective depths. First, the algebraic error is defined as a mean squared error between 2D homogenous projected points and 2D homogenous observations multiplied by projective depths. Then, the cost function that minimizes the algebraic error is defined as

$$f_{alg}(\mathbf{X}_i, \mathbf{D}_i) = \frac{1}{2}\|\hat{\mathbf{U}}_i\mathbf{D}_i - \mathbf{P}_{persp_i}\mathbf{X}_i\|_F^2, \tag{3.29}$$

where $\hat{\mathbf{U}}_i$ is a $3 \times n_p$ matrix which is the homogenous representation of observations and parameter $\mathbf{D}_i$ is a $n_p \times n_p$ diagonal matrix whose jth diagonal element is a projective depth of jth point on ith frame. $\mathbf{P}_{persp_i}$ is a perspective camera matrix on ith frame. It is 3×4 matrix when both intrinsic and extrinsic parameters are known. If we use the intrinsic matrix only, the 3D shape is

reconstructed w.r.t. the canonical camera coordinate. We used canonical camera position throughout this chapter. Therefore, $\mathbf{P}_{persp_i}$ is a 3×3 matrix, and $\mathbf{X}_i$ is a $3 \times n_p$ matrix as in the orthographic case. The derivative of f_{alg} w.r.t. $\mathbf{X}_i$ is

$$\frac{\partial f_{alg}}{\partial \mathbf{X}_i} = \mathbf{P}_{persp_i}^T(\mathbf{P}_{persp_i}\mathbf{X}_i - \hat{\mathbf{U}}_i\mathbf{D}_i). \tag{3.30}$$

As a different normalization, the inverse of a camera matrix can be multiplied to the data term. The data term in this case can be considered as a Euclidean distance in 3D space, which has the following form:

$$f_{3D}(\mathbf{X}_i, \mathbf{D}_i) = \frac{1}{2}\|\mathbf{P}_{persp_i}^{-1}\hat{\mathbf{U}}_i\mathbf{D}_i - \mathbf{X}_i\|_F^2. \tag{3.31}$$

Another way of error calculation is to consider the Euclidean distance between 2D input points and reprojected 2D points. The cost function for reprojection error in 2D space has the following form:

$$f_{reproj}(\mathbf{X}_i, \mathbf{D}_i) = \frac{1}{2}\|\mathbf{U}_i - \mathbf{\Phi}(\mathbf{P}_{persp_i}\begin{bmatrix}\mathbf{X}_i \\ \mathbf{1}^T\end{bmatrix})\|_F^2, \tag{3.32}$$

where $\mathbf{\Phi}(\mathbf{X})$ is the inhomogenous representation of homogenous 2D points $\mathbf{X}$. Let us define $\mathbf{P}_{persp_i}\begin{bmatrix}\mathbf{X}_i \\ \mathbf{1}^T\end{bmatrix} = \begin{bmatrix}\mathbf{M}_i \\ \mathbf{z}_i\end{bmatrix}$ where $\mathbf{M}_i$ is $2 \times n_p$ and $\mathbf{z}_i$ is $1 \times n_p$ matrix. Then, the derivative for f_{reproj} is

$$\begin{aligned}\frac{\partial f_{reproj}}{\partial \mathbf{X}_i} &= \langle\frac{\partial f_{reproj}}{\partial \mathbf{M}_i}, \frac{\partial \mathbf{M}_i}{\partial \mathbf{X}_i}\rangle + \langle\frac{\partial f_{reproj}}{\partial \mathbf{z}_i}, \frac{\partial \mathbf{z}_i}{\partial \mathbf{X}_i}\rangle \\ &= \mathbf{P}_{persp_i}^T\begin{bmatrix}(\hat{\mathbf{X}}_i - \mathbf{U}_i) \oslash (\mathbf{1}\mathbf{z}_i) \\ \mathbf{1}^T[(\hat{\mathbf{X}}_i - \mathbf{U}_i) \oslash (\mathbf{1}\mathbf{z}_i)] \odot \hat{\mathbf{X}}_i\end{bmatrix}\end{aligned} \tag{3.33}$$

where $\odot$ and $\oslash$ denotes element-wise multiplication and element-wise division respectively, and $\hat{\mathbf{X}}_i = \mathbf{\Phi}(\begin{bmatrix}\mathbf{M}_i \\ \mathbf{z}_i\end{bmatrix})$.

Sometimes, one can assume that there is no noise in measuring 2d points. In this case, the data term is removed from the cost function (3.3), and $\mathbf{X}_i$ becomes a function of $\mathbf{U}_i$ and $\mathbf{h}_i$. Under the noise-free orthographic projection, 3D points are function of depths, i.e., $\mathbf{h}_i \triangleq \mathbf{z}_i$. Therefore, optimization is applied to z components of the 3D points. $\mathbf{X}_i$ becomes

$$\mathbf{X}_{i,ortho}(\mathbf{z}_i) = \begin{bmatrix} \mathbf{U}_i \\ \\ \mathbf{z}_i \end{bmatrix}. \tag{3.34}$$

Hence, we minimize the following function $\mathcal{J}'$,

$$\mathcal{J}' = g(\widetilde{\mathbf{X}}, \overline{\mathbf{X}}), \tag{3.35}$$

and the gradients of $\mathcal{J}'$ w.r.t. $\overline{\mathbf{X}}$ and $\mathbf{h}_i$ is needed to optimize the cost function where the derivative w.r.t. $\mathbf{h}_i$ is given as $\frac{\partial \mathcal{J}'}{\partial \mathbf{h}_i} = \langle \frac{\partial \mathcal{J}'}{\partial \overline{\mathbf{X}}_i} \frac{\partial \mathbf{X}_i}{\partial \mathbf{h}_i} \rangle$. The derivative w.r.t. $\mathbf{z}_i$ is the last row of $\frac{\partial \mathcal{J}'}{\partial \overline{\mathbf{X}}_i}$.

For the noise-free case under perspective projection, $\mathbf{X}_i$ is the function of projective depths, i.e., $\mathbf{h}_i \triangleq \mathbf{D}_i$.

$$\mathbf{X}_{i,persp}(\mathbf{D}_i) = \mathbf{P}_{persp_i}^{-1} \hat{\mathbf{U}}_i \mathbf{D}_i. \tag{3.36}$$

The derivative of $\mathbf{X}_{i,persp}$ is

$$\begin{aligned} \frac{\partial \mathbf{X}_{i,persp}}{\partial \mathbf{d}_i} &= \langle \frac{\partial \mathcal{J}'}{\partial \mathbf{X}_i}, \frac{\partial \mathbf{X}_i}{\partial \mathbf{d}_i} \rangle \\ &= (\frac{\partial \mathcal{J}'}{\partial \mathbf{X}_i} \odot (\mathbf{P}_{persp_i}^{-1} \hat{\mathbf{U}}_i))^T \mathbf{1} \end{aligned} \tag{3.37}$$

where $\mathbf{d}_i$ is the diagonal elements of $\mathbf{D}_i$.

Regularization term

The first and intuitive example for the regularization term is the rank of aligned shapes. The concept that aligned shapes form the low-rank basis is similar to

that in [59]. This regularization assumes that the aligned shape matrix $\widetilde{\mathbf{X}}$ should have low rank since the aligned shapes consist of a few basis shapes. We use a log-determinant function [30] as a smooth surrogate for the rank function. The regularization term is

$$g_{logdet}(\widetilde{\mathbf{X}}) = \frac{1}{2}\log|\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T + \mu\mathbf{I}|, \tag{3.38}$$

where $|\mathbf{A}|$ is a determinant of matrix $\mathbf{A}$. The derivative of g w.r.t. $\widetilde{\mathbf{X}}$ is

$$\frac{\partial g_{logdet}}{\partial \widetilde{\mathbf{X}}} = (\widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^T + \mu\mathbf{I})^{-1}\widetilde{\mathbf{X}}. \tag{3.39}$$

One may seek the possibility of using a nuclear norm instead of a log-determinant function. While the subgradient of a nuclear norm can be calculated easily based on SVD, we found out empirically that using the log-determinant gives better solution. Moreover, minimizing log-determinant is closely related to maximizing the log-likelihood of a Gaussian prior [61]. Hence, we adopted a log-determinant function rather than a nuclear norm.

Another possible regularization is to consider the temporal dependency between the shapes. If we assume that each point has a momentum, in other words, velocity of a point of an aligned shape changes smoothly, then the acceleration of the point has a small value. Therefore, we can minimize the acceleration of a point, i.e.,

$$g_{acc}(\widetilde{\mathbf{X}}) = \frac{1}{2}\sum_{i}^{n_f}\sum_{j}^{n_p}\|\mathbf{a}_{ij}\|^2, \tag{3.40}$$

where $\mathbf{a}_{ij}$ represents acceleration of jth point at ith frame and defined as $\mathbf{a}_{ij} = (\widetilde{\mathbf{x}}_{i+1,j} - \widetilde{\mathbf{x}}_{i,j}) - (\widetilde{\mathbf{x}}_{i,j} - \widetilde{\mathbf{x}}_{i-1,j})$, and $\widetilde{\mathbf{x}}_{i,j}$ is the jth point of the aligned shape at ith frame. The derivative w.r.t. $\widetilde{\mathbf{X}}_i$ is

$$\frac{\partial g_{acc}(\widetilde{\mathbf{X}})}{\partial \widetilde{\mathbf{X}}_i} = \widetilde{\mathbf{X}}_{i-2} - 4\widetilde{\mathbf{X}}_{i-1} + 6\widetilde{\mathbf{X}}_i - 4\widetilde{\mathbf{X}}_{i+1} + \widetilde{\mathbf{X}}_{i+2}. \tag{3.41}$$

Note that the acceleration constraint is imposed on aligned shapes. Previous work which seeks for a smooth trajectory in a shape space showed promising results [38]. This smoothness assumption can also be incorporated as a regularization based on the above formulation. In cases when camera matrices are known, temporal smoothness can be applied on the 3D shapes in world coordinates. This assumption can be more reasonable since aligned shapes lack rigid motions. In this case, the acceleration function depends on the variable $\mathbf{X}_i$, so the cost function should be integrated into the data term to calculate the gradient w.r.t. $\mathbf{X}_i$.

3.2.4 Handling Missing Points

Missing points can be easily dealt with in the proposed framework by removing the error costs of unobserved points from the data term. For example, let $\mathbf{O}_i = [\mathbf{o}_{i1}\ \mathbf{o}_{i2}\ \cdots\ \mathbf{o}_{in_p}]$ be the $3 \times n_p$ matrix where

$$\mathbf{o}_{ij} = \begin{cases} [1\ 1\ 1]^T, & \text{if } j\text{th point is visible.} \\ [0\ 0\ 0]^T, & \text{otherwise.} \end{cases} \tag{3.42}$$

Then, the data term for algebraic error with missing points has the following form:

$$f_{alg,occ} = \frac{1}{2}\|\mathbf{O}_i \odot (\mathbf{U}_i\mathbf{D}_i - \mathbf{P}_i\mathbf{X}_i)\|_F^2. \tag{3.43}$$

The derivative of f_{occ} w.r.t. $\mathrm{vec}(\mathbf{X}_i)$ is

$$\frac{\partial f_{alg,occ}}{\partial \mathrm{vec}(\mathbf{X}_i)} = \mathrm{vec}([\mathbf{O}_i \odot (\mathbf{P}_i\mathbf{X}_i - \mathbf{U}_i\mathbf{D}_i)])^T\hat{\mathbf{O}}_i(\mathbf{I} \otimes \mathbf{P}_i), \tag{3.44}$$

where $\hat{\mathbf{O}}_i = \mathrm{diag}(\mathrm{vec}(\mathbf{O}_i))$. The other data terms proposed in Section 3.2.3 can be modified to deal with missing points in a similar manner.

3.2.5 Optimization

Putting the derivatives from Section 3.2.2 and Section 3.2.3 together, it is possible to calculate the gradients of the cost function w.r.t. $\mathbf{X}_i$ and $\overline{\mathbf{X}}$. Any gradient-based solver can be efficiently used to minimize the cost function, and in this dissertation, we choose L-BFGS algorithm [77] for its low computation cost and memory efficiency.

For perspective projection, the bound constraints for projective depth are incorporated into the optimization method. L-BFGS-B [15] can be utilized in this case to handle the bound constraints. There is no specific rule to determine the bounding value of projective depths. In this dissertation, we simply used $d_{ij} \geq 1$ for all i, j where d_{ij} is the projective depth of the jth point on the ith frame.

As a consequence, each iteration of the proposed method has time and space complexity of $O(n_p n_f)$, which is advantageous over EM-PND which has time and space complexity of $O(n_p^2 n_f)$.

3.2.6 Initialization

Since the proposed cost function is highly nonlinear, it is important to find a good initial point. For the orthographic case, we used the same initialization scheme as in [61]. First, initial rotations are determined following the initialization method used in [38]. Rotations are calculated repeatedly using the factorization method for several numbers of shape bases, and those that satisfy the orthogonality constraint most are selected. Then, initial shapes are determined so that the trace of the sample covariance of the aligned shapes is minimized [61]. The input points are also normalized before the initialization as in [61].

For the perspective case, orthographic initialization cannot be used since reflection ambiguities should not exist. Therefore, we directly minimize the rank of the shape matrix. The method is similar to [24], but we used log-determinant instead of a nuclear norm. In other words, we minimize the following cost function:

$$\min \frac{1}{2}\log|\mathbf{X}\mathbf{X}^T + \mu\mathbf{I}| \quad \text{s.t. } \mathbf{X}_i = \mathbf{U}_i\mathbf{D}_i \text{ for all } i, \tag{3.45}$$

where $\mathbf{X} = [\mathbf{X}_1^T\ \mathbf{X}_2^T\ \cdots\ \mathbf{X}_{n_f}^T]^T$ is a $3n_f \times n_p$ matrix as in [24]. This problem is solved via L-BFGS-B with the same bounded constraint explained in Section 3.2.5. $\mathbf{D}_i$ is initially set to an identity matrix for all i. When missing points exist, the constraint in (3.45) is changed to $\mathbf{O}_i \odot \mathbf{X}_i = \mathbf{O}_i \odot \mathbf{U}_i\mathbf{D}_i$. As observed 2D points may have different scale according to the distance between a camera and a shape in perspective projection, we roughly align the scale of shapes using the observed 2D points. The normalized points are calculated as

$$\mathbf{U}_i' = \frac{\hat{\mathbf{U}}_i}{\|\mathbf{U}_i\|_F}. \tag{3.46}$$

In the case that the whole camera matrices are known, one can apply this initialization method instead to the rank of the trajectory matrix. In this case, the matrix $\mathbf{X}$ in (3.45) is rearranged as a $3n_p \times n_f$ matrix. After $\mathbf{X}_i$ is initialized, $\overline{\mathbf{X}}$ and $\mathbf{R}_i$ are initialized via GPA.

3.3 Experimental Results

We measured reconstruction performance for both orthographic and perspective camera cases. The performance is measured in terms of the normalized reconstruction error, i.e.,

$$e_{ortho} = \frac{1}{n_f}\sum_{i=1}^{n_f}\frac{\|\hat{\mathbf{X}}_i - \mathbf{X}_i^*\|}{\|\mathbf{X}_i^*\|}, \tag{3.47}$$

where $\hat{\mathbf{X}}_i$ is the reconstructed 3D shape and $\mathbf{X}_i^*$ is the ground truth 3D shape on the ith frame. The reconstructed shape has a reflection ambiguity in the case of orthographic camera. Therefore, we also evaluate the error for the reflected shape over Z-axis for each frame, and the shape that has a smaller error is chosen for each frame individually.

For the perspective camera, the shape is reconstructed up to scale in the proposed method. Therefore, we scale the shape appropriately to minimize the normalized error as

$$\alpha_i = \frac{\mathrm{vec}(\mathbf{X}_i^*)^T \mathrm{vec}(\hat{\mathbf{X}}_i)}{\|\mathbf{X}_i^*\|^2}. \tag{3.48}$$

Before the normalized error is calculated, the mean of the n_f ground truth points is subtracted on each frame to remove translation component. The normalized error for perspective projection is calculated as

$$e_{persp} = \frac{1}{n_f} \sum_{i=1}^{n_f} \frac{\|\alpha_i \hat{\mathbf{X}}_i - \mathbf{X}_i^*\|}{\|\mathbf{X}_i^* - \bar{\mathbf{X}}_i^*\|}, \tag{3.49}$$

where $\bar{\mathbf{X}}_i^* = \frac{1}{n_p} \mathbf{X}_i^* \mathbf{1}\mathbf{1}^T$.

For all the methods used in the experiments, reflection ambiguities or scale ambiguities are solved in the same way. In the following, we conducted extensive experiments on synthetic data and real world data including the case with missing points.

3.3.1 Orthographic Projection

For the experiments of orthographic projection, we used motion capture datasets from Akhter et al. [8] and Torresani et al. [108]. The performance of the proposed algorithm is compared with various NRSfM algorithms under orthographic camera assumption: EM-PPCA [108], MP [79], CSF2 [38], SPM [24], EM-PMP [62], EM-PND [61], and CNR [60]. We evaluated our method using two

Table 3.1: Normalized reconstruction errors under orthographic projection

Data	EM-PPCA [108]	MP [79]	SPM [24]	CSF2 [38]	EM-PMP [62]
walking	0.1485	0.4231	0.0861	0.0709	0.0424
shark	0.0688	0.1254	0.1659	0.0551	0.0099
face	0.0208	0.0328	0.0233	0.0209	0.0166
yoga	0.6100	0.5924	0.0224	0.0225	0.0128
stretch	0.5392	0.5915	0.0288	0.0219	0.0156
pickup	0.5149	0.3465	0.0356	0.0607	0.0124
drink	0.1292	0.2650	0.0216	0.0123	0.0018
dance	0.2325	0.4062	0.1445	0.1362	0.1278
AVG	0.2830	0.3479	0.0660	0.0501	0.0299
Data	EM-PND [61]	CNR [60]	PR	$\mathrm{PR}_{\mathrm{acc}}$	
walking	0.0407	0.0395	0.0544	0.0520	
shark	0.0134	0.0832	0.0272	0.0104	
face	0.0150	0.0249	0.0164	0.0188	
yoga	0.0128	0.0382	0.0175	0.0175	
stretch	0.0150	0.0370	0.0156	0.0156	
pickup	0.0133	0.0574	0.0157	0.0158	
drink	0.0031	0.0152	0.0063	0.0063	
dance	0.1247	0.0734	0.1266	0.1242	
AVG	0.0298	0.0461	0.0350	0.0326	

Table 3.2: Running time (sec) of reconstruction under orthographic projection

Data	EM-PPCA [108]	MP [79]	SPM [24]	CSF2 [38]	EM-PMP [62]
walking	86.72	3.30	1088.33	24.61	153.03
shark	247.68	1.83	916.32	5.58	207.25
face	59.53	2.34	320.62	32.74	77.48
yoga	61.08	2.29	79.85	34.26	86.07
stretch	72.55	2.77	83.73	48.45	114.88
pickup	70.46	2.68	59.58	16.38	106.15
drink	204.83	7.76	149.69	100.54	240.41
dance	166.34	2.23	171.16	69.20	292.03
AVG	121.15	3.15	358.66	41.47	141.92

Data	EM-PND [61]	CNR [60]	PR	$\mathrm{PR}_{\mathrm{acc}}$
walking	208.44	92.49	22.77	22.77
shark	482.15	111.89	24.84	28.13
face	93.05	100.00	4.25	4.87
yoga	165.20	87.97	30.84	31.37
stretch	160.65	106.96	33.17	33.25
pickup	161.06	106.78	42.21	40.09
drink	409.59	311.81	34.88	36.17
dance	387.17	104.38	79.21	81.32
AVG	258.41	127.78	34.02	34.75

Table 3.3: Average camera rotation errors (degree)

Data	EM-PPCA [108]	MP [79]	SPM [24]	CSF2 [38]	EM-PMP [62]
walking	7.631	28.999	3.084	2.910	2.271
jaws	3.076	6.060	5.975	2.266	**0.392**
face	0.678	1.157	0.713	0.598	0.539
yoga	39.552	41.608	0.816	0.810	0.547
stretch	43.525	43.454	1.125	0.717	**0.499**
pickup	30.095	21.990	1.483	2.263	**0.432**
drink	7.097	15.957	1.132	0.449	**0.067**
dance	15.894	25.563	5.868	7.482	5.051
AVG	18.443	23.098	2.525	2.187	**1.225**

Data	EM-PND [61]	CNR [60]	PR	$\mathrm{PR}_{\mathrm{acc}}$
walking	2.095	**1.566**	3.707	3.346
jaws	1.272	4.027	0.542	0.426
face	0.586	1.094	**0.502**	0.556
yoga	**0.498**	1.252	0.629	0.624
stretch	0.593	1.432	0.682	0.679
pickup	0.449	1.965	0.631	0.628
drink	0.126	0.791	0.248	0.248
dance	4.184	**2.375**	5.025	5.063
AVG	**1.225**	1.813	1.496	1.446

regularization costs: one is using the log-determinant regularizer (PR) as in (3.14), and the other is using both the log-determinant in (3.14) and the acceleration regularization in (3.16) (PR_{acc}). The performances of reconstruction are illustrated in Table 3.1. The smallest reconstruction error and the second smallest error for each dataset are marked as (1) and (2) respectively in the table. We choose $\lambda = 5 \times 10^{-8} n_f, \mu = 10^{-7} n_f$ for the experiments of orthographic projection. The weight for the acceleration regularization is set to $\lambda_{acc} = 1.25 \times 10^{-3}$.

Overall, the performance of PR and PR_{acc} is superior to the methods that do not take shape alignment into account, but it is slightly inferior to the state-of-the-art methods, EM-PND and EM-PMP. The proposed framework shows slightly better performance when the acceleration regularization is combined. CNR [36] performs best in walking and dance sequence, but its performance generally is not as good as the state-of-the-art. CNR is a part-based algorithm designed especially to solve very complex deformations, and its performance can be a bit worse than the best result that a holistic approach can achieve if the input data is simple as in this experiment. Fig. 3.3 provides qualitative results on pickup and dance sequences. It is shown that PR shows competitive reconstruction performance with EM-PND and it reconstructs 3D shapes more precisely than CSF2.

We also measured the running time of the algorithms, and the results are shown in Table 3.2. It is shown that PR and PR_{acc} are much faster than EM-based methods, which verifies the advantage of the proposed regression framework in terms of time complexity. In summary, the proposed methods provide reconstruction results that are competitive with the state-of-art methods while requiring much less time for reconstruction.

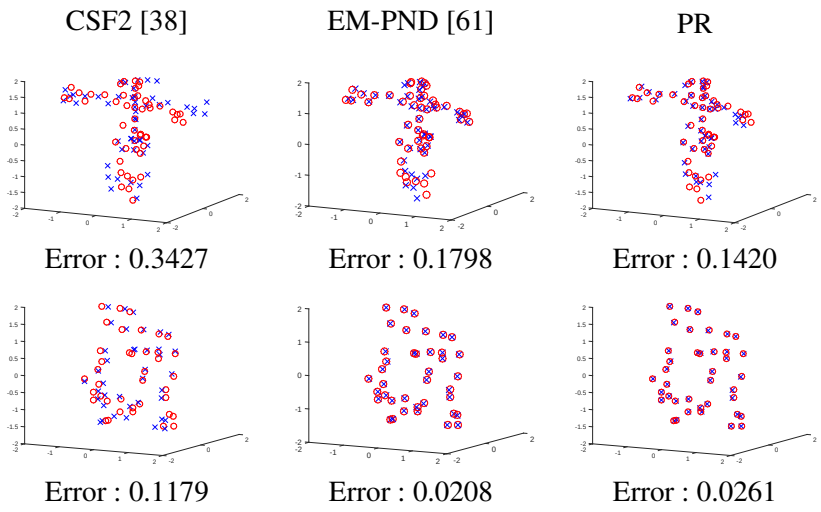


Figure 3.3: Qualitative results under orthographic projection. The results of CSF2, PND, and our method on dance sequence (top) and pickup sequence (bottom) are illustrated ($\circ$(red): ground truth points, $\times$(blue): reconstructed points).

In addition to the accuracy of the reconstructed shape, we also evaluated the accuracy of camera motion estimation. We found the rotation matrix, which aligns the reconstructed shape to the ground truth shape, via Procrustes analysis. Then, the angle of the rotation matrix is calculated for each frame in the axis-angle representation. The average of the angles is shown in Table 3.3. The proposed method accurately estimates the rotation of the camera matrix, and similar to the reconstruction performance, it outperforms the other methods except EM-PND and EM-PMP.

To evaluate the robustness to noisy data, experiments on data with gaussian noise are conducted. Following [61], the standard deviation of the gaussian noise is set to $\sigma_{noise} = 0.02 \max_{i,j,k} |u_{ijk}|$. The results are shown in Table 3.4. The proposed framework is more sensitive in parameter tuning under the effect of noise, compared to the other heavier methods. When the noise

Table 3.4: Normalized reconstruction errors on noisy data

Data	EM-PPCA [108]	MP [79]	SPM [24]	CSF2 [38]	EM-PMP [62]
walking	0.1357	0.3174	0.1047	0.0921	0.0925
shark	0.0501	0.1282	0.1846	0.1196	0.0631
face	0.0449	0.0517	0.0497	0.0527	0.0410
yoga	0.5253	0.6233	0.0841	0.0796	**0.0314**
stretch	0.5416	0.5785	0.0848	0.0542	**0.0348**
pickup	0.5023	0.3659	0.0981	0.0701	**0.0307**
drink	0.1768	0.2706	0.0406	0.0363	0.0244
dance	0.2233	0.4056	0.1668	0.1556	0.1419
AVG	0.2750	0.3427	0.1017	0.0825	0.0575
Data	EM-PND [61]	CNR [60]	PR	$\mathrm{PR}_{\mathrm{tune}}$	
walking	0.0760	**0.0395**	0.1516	0.0777	
shark	**0.0596**	0.0832	0.1320	0.1071	
face	0.0423	**0.0247**	0.0815	0.0444	
yoga	0.0407	0.0387	0.0754	0.0600	
stretch	0.0452	0.0366	0.0758	0.0558	
pickup	0.0414	0.0571	0.0736	0.0496	
drink	0.0339	**0.0151**	0.0681	0.0424	
dance	0.1372	**0.0761**	0.1846	0.1721	
AVG	0.0595	**0.0464**	0.1053	0.0761	

Table 3.5: Normalized errors on face sequence with structured missing points

Method	EM-PPCA [108]	MP [79]	SPM [24]	CSF2 [38]
Error	0.2918	0.0831	0.0398	0.0362
Method	EM-PMP [62]	EM-PND [61]	PR	
Error	**0.0202**	0.0277	**0.0202**	

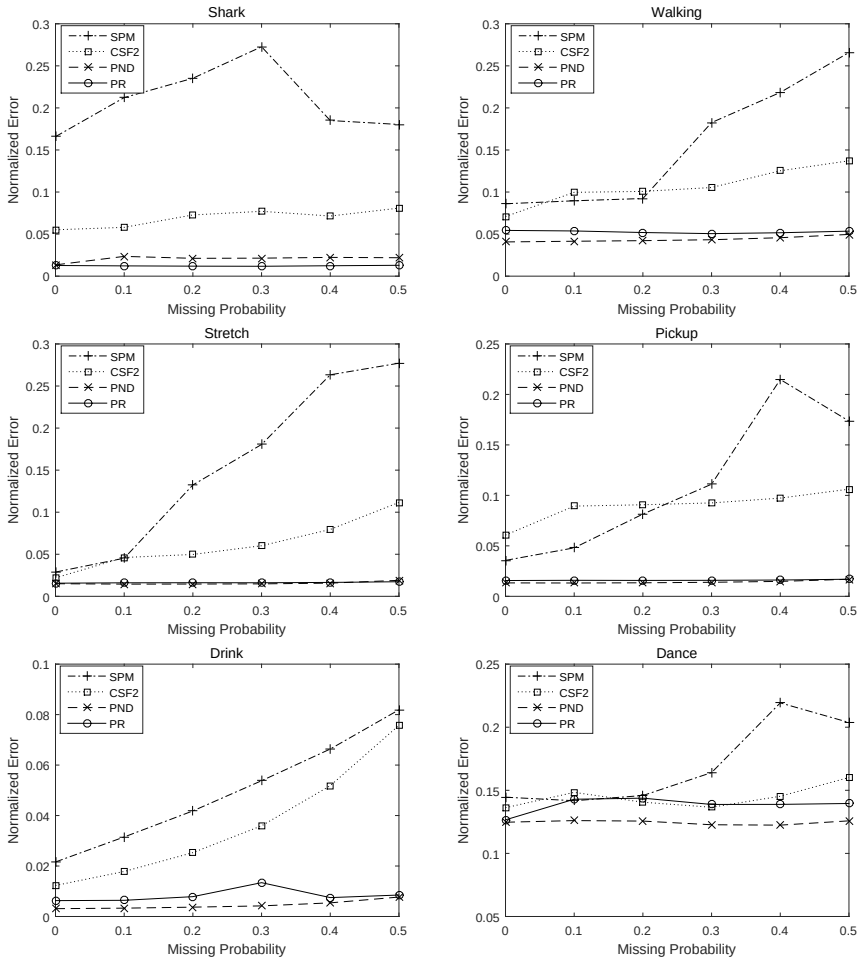
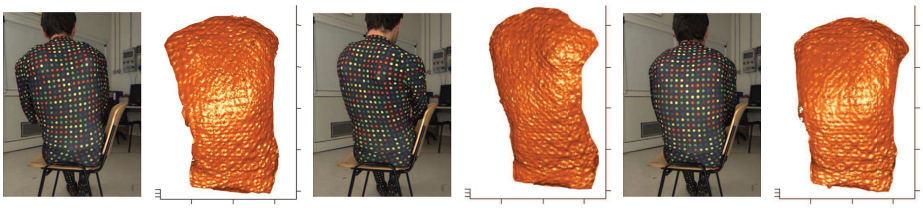


Figure 3.4: Normalized errors under orthographic projection with missing points. The results of SPM, CSF2, PND, and our method on 6 different sequences are compared.

Figure 3.5: The missing pattern for the structured missing data of *face* sequence.

is severe, the parameters λ and μ need to be increased. We set the values to $\lambda = 10^{-4}n_f, \mu = 10^{-5}n_f$ in order to raise the effect of the regularization term ($\mathrm{PR}_{\mathrm{tune}}$). We also provide the result without parameter tuning (PR). Before tuning the parameter, the performance is similar to SPM. With the new parameters, the proposed method outperforms the other algorithms except EM-PND, EM-PMP and CNR, which are much heavier than the proposed method. CNR builds a strong reconstruction from numerous part-based weak reconstructions, and inaccurate weak reconstructions are effectively removed during the process of obtaining a strong reconstruction. Therefore, it is essentially robust to the noisy inputs compared to the other algorithms.

Next, we evaluated the performance when missing points exist. For each dataset, missing points are randomly picked with varying the portion from 0% to 50%. The normalized error for each case is measured by averaging 10 trials. The results on six sequences are shown in Fig 3.4. It can be verified that both PR and EM-PND are robust to missing points since the performance does not get much worse even when 50% of the points are missing. Meanwhile, reconstruction errors of the other methods grow as the portion of missing points gets bigger. To simulate realistic occlusion, we generate structured missing data on *face* sequence. We followed [80] to infer the missing points due to self-occlusion. The missing rate for each frame is between 5% and 45%, and total 20.9% of the points are marked as missing points. The missing data pattern is visualized in Figure 3.5. In the figure, each row represents missing pattern of each point throughout the sequence. White points indicate missing points, while black ones are observed points. The normalized errors are shown in Table 3.5. EM-PMP and PR perform best, which verifies the robustness over structured occlusion.



frame 22 frame 90 frame 139

Figure 3.6: Dense reconstruction results on the back dataset [4].

Table 3.6: Normalized errors and running time on the pants sequences [117]

Dataset		CSF2 [38]	EM-PND [61]	PR
pace	Normalized error	0.1037	0.0747	0.0995
	Time (sec)	906	49179	1318
	Memory (MB)	717	31543	1483
jump	Normalized error	0.1507	0.1107	0.1153
	Time (sec)	1337	128784	3961
	Memory (MB)	755	86623	1830

We also evaluated our method on dense sequences. First, we evaluated our method on pants motion capture sequences provided in [117]. The dataset contains 1,453 points for each frame. Normalized error, running time, and memory consumption for CSF2, EM-PND, and PR are illustrated in Table 3.6. Due to large memory consumption of EM-PND, we used the modified implementation of [61] which is less efficient in terms of running time. There is a trade-off between the performance and the complexity for the three methods. Our method achieves reasonable running time and memory consumption while showing better performance than CSF2. The drawback of EM-PND is that the memory consumption and running time of the method increase quadratically with respect to

the number of points due to EM-algorithm. On the other hand, they increase linearly for our method. In table 3.6, our method is more than 30 times faster than EM-PND and consumes only 3% of the memory that used in EM-PND. Therefore, the experimental result indicates the efficiency of the proposed method in terms of memory and time consumption over EM-PND.

Lastly, the method is evaluated on another dense dataset, the back data [92], which contains movements of a human torso. We used the trajectories generated by [4]. The dataset consists of 20,561 dense point correspondences on 149 frames. Our method took 10,744 seconds and consumed 5,225 MB memory for the back data. The qualitative result of our method is illustrated in Fig. 3.6. We visualized 3D mesh of the reconstruction results on 3 selected frames. Our method correctly captures not only the motion variations of moving torso but also non-rigid motion variations such as expanding or shrinking of the back.

3.3.2 Perspective Projection

We evaluated the performance for the perspective projection case using the dataset provided in [112]. The dataset contains 100 different sequences from CMU MoCap database [1]. Each of the sequence is truncated to 100 frame, and each frame contains 31 points of human body parts. Following the convention of [112], the 3D points are projected via synthetic camera with a fixed focal length which orbits around 3D shapes. Four different sets of 2D input points are generated by varying the angular velocity of the synthetic camera. We compared our method to the recent state-of-the-art methods, which are trajectory-based algorithms including TB [8], TB-L_1 [134], and TF [112]. Note that the compared methods require that global camera matrices are known while the proposed

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

method only requires the intrinsic parameters of the cameras. The parameters for these algorithms were chosen as the ones that show the best performance for each angular speed. As mentioned in Section 2.3, shape-basis-based perspective NRSfM algorithms mostly work poorly in sequences containing complex shape variations [82]. Hence, we only compared the proposed method with recent, state-of-the-art trajectory-based algorithms for the perspective case.

We provided the result of our method using three different data costs covered in Section 3.2.3. The methods that use (3.30), (3.31), (3.32) as a data term are referred to as $\mathrm{PR}_{\mathrm{alg}}$, $\mathrm{PR}_{\mathrm{3D}}$, and $\mathrm{PR}_{\mathrm{reproj}}$, respectively. The log-determinant regularizer in (3.38) is used as the regularization term in all cases. The parameters are determined as $\lambda = 5\times10^{-2}, \mu = 5\times10^{-4}$ for $\mathrm{PR}_{\mathrm{alg}}$, $\lambda = 5\times10^{-4}, \mu = 5\times10^{-4}$ for $\mathrm{PR}_{\mathrm{3D}}$, and $\lambda = 5\times10^{-2}, \mu = 5\times10^{-5}$ for $\mathrm{PR}_{\mathrm{reproj}}$, respectively.

The normalized errors are illustrated in Table 3.7. Our methods show smaller reconstruction errors compared to other methods in various angular speeds that varies from 1° to 10° per frame. Especially, performance of the trajectory-based methods are significantly affected by the angular speed of the camera. In real world situations, fast camera movement that covers different sides of a 3D shape is impractical. Error of our method is much smaller than the other methods in the case of 1° per frame. Another advantage of our algorithm over the trajectory-based methods is that our method is able to reconstruct 3D shapes in the case of unknown extrinsic parameters of camera. Most trajectory-based algorithms need to know camera position and pose to make use of smoothness or basis information of point trajectories while our method is able to reconstruct 3D shapes as well as the relative position to the camera. Qualitative result on the 49th sequence of the dataset is illustrated in Fig 3.7. The result of the same frame with different orbiting speed is shown. Trajectory-based methods show

Table 3.7: Normalized errors of reconstruction under perspective projection

Deg/Frame	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{alg}}$	$\mathrm{PR}_{\mathrm{reproj}}$	$\mathrm{PR}_{\mathrm{3D}}$
1	0.5415	0.4469	0.8172	0.2290	0.2123	**0.2107**
2	0.4116	0.3856	0.3853	0.1770	0.1634	**0.1616**
5	0.3003	0.2473	0.1714	0.1193	0.1077	**0.1073**
10	0.2114	0.1490	0.0934	0.0920	**0.0809**	0.0832

Table 3.8: Normalized errors of perspective initialization

Deg/Frame	Shape-based	Trajectory-based
1° / frame	0.3205	0.3578
2° / frame	0.2875	0.3085
5° / frame	0.2335	0.2549
10° / frame	0.2127	0.2111

poor reconstruction performance when the orbiting speed is 1° / frame, and they show better result as the orbiting speed grows. However, our method shows consistent reconstruction results regardless of the orbiting speed of the camera. Since our method is basically a shape-based method, the coverage of viewpoints of the camera does not affect severely on the reconstruction performance unlike the trajectory-based methods.

We also evaluated performance of the initialization algorithm for perspective projection explained in Section 3.2.6. We suggested two initialization scheme, one of which is to minimize the rank of the shape matrix, and the other is to minimize the rank of the trajectory matrix. The normalized errors of both initialization algorithms are illustrated in Table 3.8. The result indicates that

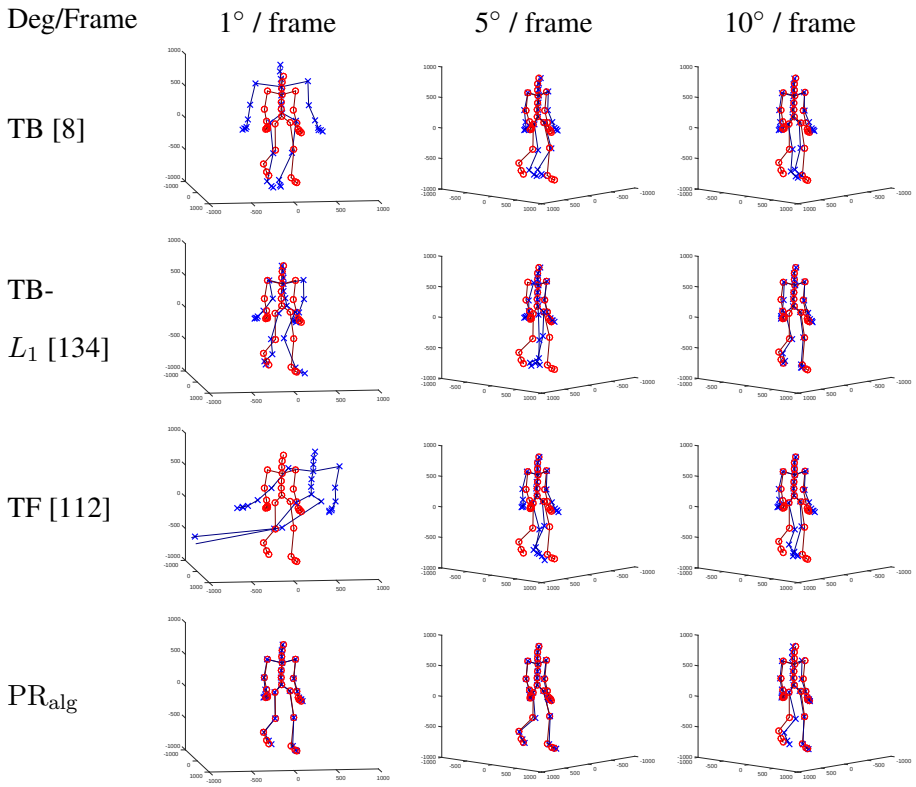


Figure 3.7: Qualitative results under perspective projection. The results of $\mathrm{PR}_{\mathrm{alg}}$ show consistent quality in various angular speeds while trajectory-based methods show poor reconstruction results in slower orbiting speed ($\circ$(red): ground truth points, $\times$(blue): reconstructed points).

both shape and trajectory-based methods show similar reconstruction results, but trajectory-based initialization is more sensitive to the orbiting speed. This result is analogous to the results of the trajectory-based methods which showed better performance in fast orbiting speed cases.

As in the orthographic experiments, robustness to the gaussian noise and occlusion are also measured under the perspective projection. First, we added gaussian noise to the dataset. The standard deviation of gaussian noise is the

Table 3.9: Normalized errors on noisy data under perspective projection

Deg/Frame	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{alg}}$	$\mathrm{PR}_{\mathrm{reproj}}$
1	0.5419	0.4420	0.7455	0.3403	0.3524
2	0.4123	0.3963	0.4033	0.2920	0.3182
5	0.3025	0.2537	**0.1960**	0.2786	0.3043
10	0.2133	0.1632	**0.1239**	0.2705	0.2935
Deg/Frame	$\mathrm{PR}_{\mathrm{3D}}$	$\mathrm{PR}_{\mathrm{alg\text{-}tune}}$	$\mathrm{PR}_{\mathrm{reproj\text{-}tune}}$	$\mathrm{PR}_{\mathrm{3D\text{-}tune}}$	
1	0.3388	0.2992	0.3327	**0.2869**	
2	0.2939	0.2386	0.2861	**0.2275**	
5	0.2778	0.2200	0.2701	0.2108	
10	0.2699	0.2202	0.2635	0.2056	

same as the orthographic case. Table 3.9 shows the average reconstruction error of 10 trials on noisy data. In this case, λ and μ for all methods of PR are increased by 20 times compared to the original. When the orbiting speed is slow, the proposed method outperforms trajectory-based methods, and the error is similar to the case without noise. In this case, however, our method does not improved much as the orbiting speed gets faster unlike the trajectory-based methods. Parameter tuning increases performance by about 15-20%.

Next, we examined occlusion robustness of the proposed scheme under perspective projection. As in the orthographic case, we measured normalized errors of the sequences with varying missing points ratio from 0% to 50%. Fig. 3.8 shows the average normalized errors over 100 sequences with various orbiting speed. As ratio of missing points grows, the errors of the proposed methods increase slightly. Three proposed cost function $\mathrm{PR}_{\mathrm{alg}}$, $\mathrm{PR}_{\mathrm{reproj}}$, $\mathrm{PR}_{\mathrm{3D}}$ show

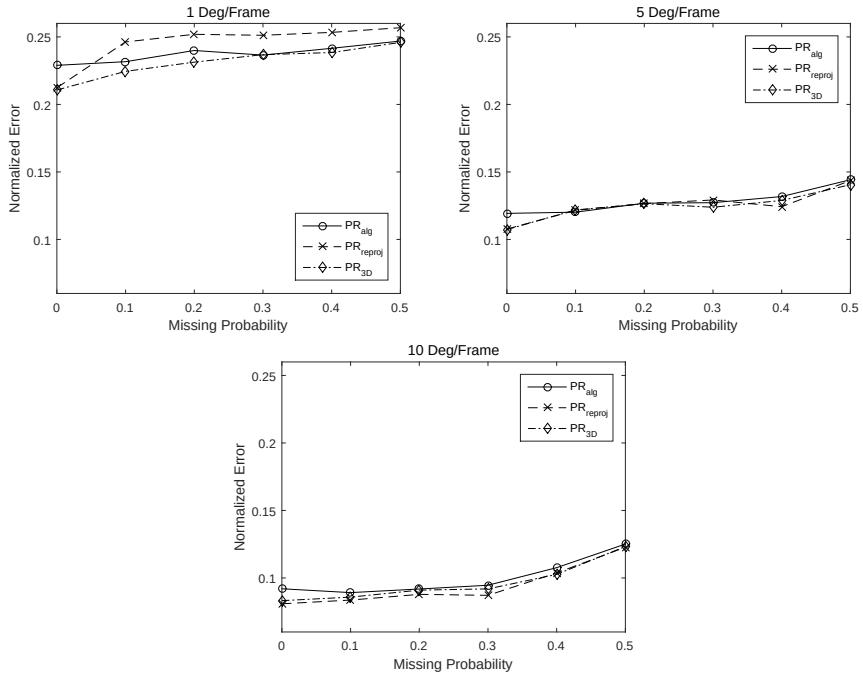


Figure 3.8: Normalized errors of $\mathrm{PR_{alg}}$, $\mathrm{PR_{reproj}}$, $\mathrm{PR_{3D}}$ with missing points.

similar robustness to missing points.

In addition to the data projected using synthetic cameras, we experimented the algorithm on the Human 3.6M dataset [49] to evaluate the performance of our method on real cameras. 2D input sequences generated from four calibrated cameras are provided with the corresponding 3D motion capture data. The dataset consists of 15 action sequences performed by 7 different subjects. We used 2 subjects, S9 and S11, for our experiments. We used the first 200 frames for each sequence with downsampling. Since there are four cameras whose positions are fixed, the experiments are conducted under three different scenarios: (1) using only a single camera, (2) alternating four cameras for each frame, (3) alternating four cameras for every 50 frames. The result of our method $\mathrm{PR_{3D}}$ as well as the trajectory-based methods are shown in Ta-

Table 3.10: Normalized errors on the Human 3.6m sequences under fixed camera setting (Setting 1)

Sequence	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{3D}}$
Directions	9.4328	**0.2990**	9.3631	0.4563
Discussion	9.3649	**0.2837**	9.3344	0.4137
Eating	11.3518	**0.4298**	10.9526	0.7637
Greeting	9.1384	0.3116	8.8320	**0.2755**
Phoning	9.0283	0.2614	8.3971	**0.1098**
Photo	9.4587	0.3419	8.7733	**0.2425**
Posing	8.7433	**0.2173**	8.8267	0.3468
Purchases	10.1387	**0.3852**	10.1265	0.4990
Sitting	11.8102	**0.5097**	11.7995	0.5782
SittingDown	11.7138	0.6323	11.8050	**0.6226**
Smoking	9.3101	0.2707	9.0095	**0.1631**
Waiting	10.1283	**0.2852**	9.6164	0.3347
WalkDog	9.5505	0.4139	9.1203	**0.2532**
Walking	9.5367	0.2922	8.9284	**0.0989**
WalkTogether	8.7306	0.3563	8.0695	**0.0907**
AVG	9.8291	0.3527	9.5303	**0.3499**

Table 3.11: Normalized errors on the Human 3.6m sequences under fast rotating camera setting (Setting 2)

Sequence	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{3D}}$
Directions	0.0308	0.3039	**0.0224**	0.0990
Discussion	0.0324	0.2848	**0.0199**	0.0742
Eating	0.0335	0.4356	**0.0190**	0.0884
Greeting	0.0415	0.3177	**0.0249**	0.1218
Phoning	0.0368	0.3807	**0.0286**	0.0668
Photo	0.0311	0.3652	**0.0234**	0.0867
Posing	0.0291	0.2156	**0.0191**	0.0733
Purchases	0.0314	0.4011	**0.0200**	0.1244
Sitting	0.0190	0.5187	**0.0121**	0.0649
SittingDown	0.0178	0.6353	**0.0117**	0.0700
Smoking	0.0309	0.2773	**0.0248**	0.0766
Waiting	0.0280	0.3155	**0.0204**	0.0785
WalkDog	0.0520	0.4715	**0.0321**	0.1493
Walking	0.0625	0.3986	**0.0360**	0.0696
WalkTogether	0.0408	0.4708	**0.0273**	0.0606
AVG	0.0345	0.3861	**0.0228**	0.0869

Table 3.12: Normalized errors on the Human 3.6m sequences under moderately rotating camera setting (Setting 3)

Sequence	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{3D}}$
Directions	0.3579	0.2986	0.3353	**0.1789**
Discussion	0.5084	0.2715	0.3828	**0.1384**
Eating	0.4078	0.3993	0.3266	**0.1668**
Greeting	0.5381	0.3289	0.4596	**0.1900**
Phoning	1.2841	0.2550	0.4507	**0.1120**
Photo	0.6677	0.4386	0.3282	**0.1859**
Posing	0.4695	**0.2496**	0.3610	0.2696
Purchases	0.4665	0.4151	0.3735	**0.2094**
Sitting	0.6235	0.5017	0.3037	**0.2287**
SittingDown	0.6869	0.7717	**0.2587**	0.2694
Smoking	0.9317	0.6563	0.3631	**0.1286**
Waiting	0.5910	0.3304	0.3993	**0.1743**
WalkDog	1.1870	0.4246	0.5409	**0.3210**
Walking	1.3644	0.4291	0.6334	**0.0976**
WalkTogether	1.4867	0.4718	0.6613	**0.1007**
AVG	0.7714	0.4161	0.4119	**0.1848**

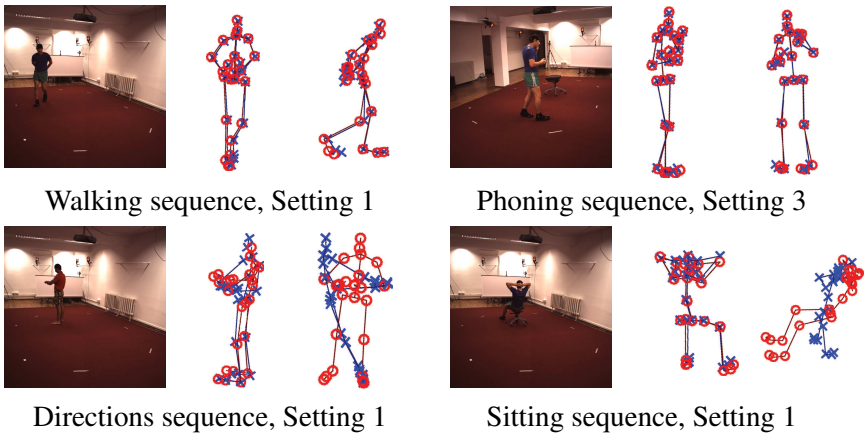


Figure 3.9: Qualitative results of the proposed method on the Human 3.6M dataset. The proposed method is able to reconstruct accurate 3D shapes in various camera settings. The second row shows the failure cases where the sequences contain very complex motion variations ($\circ$(red): ground truth points, $\times$(blue): reconstructed points).

ble 3.10, Table 3.11, and Table 3.12. Under the fixed-camera setting (Setting 1), our method outperforms the other methods. Some sequences with complex motion, such as *Eating* or *SittingDown*, all of the methods have poor reconstructions. However, $\mathrm{PR}_{\mathrm{3D}}$ still shows best or second-best performance. When there is fast rotation of the camera (which is simulated in Setting 2), trajectory-based methods show better performance, but the proposed method also gives very accurate reconstruction results. In Setting 3, although the sequence contains 2D observations from all cameras, the rotation of the camera is not fast. The proposed method beats the trajectory-based methods with a huge margin since the proposed framework does not assume temporal smoothness on 3D shapes. Qualitative results of our method are given in Fig. 3.9. Both successful reconstructions and failure cases are shown. All of the methods tend to output

Table 3.13: Normalized errors on the Human 3.6M Walking sequence with structured missing points

Settings	TB [8]	TB-L_1 [134]	TF [112]	$\mathrm{PR}_{\mathrm{alg}}$	$\mathrm{PR}_{\mathrm{reproj}}$	$\mathrm{PR}_{\mathrm{3D}}$
1	9.2598	0.3149	8.7913	0.1667	**0.0999**	0.1403
2	0.1660	0.4391	**0.0488**	0.2091	0.0736	0.1659
3	1.2976	0.5138	0.7116	0.1948	**0.1172**	0.1643

inaccurate results when the sequence contains very complex motion variations.

Lastly, we evaluate our method on the data with structured missing points. To simulate realistic occlusion of human body, we used a 2D human pose estimator [116] and detected the location of joints for each frame from the RGB images of the Human 3.6M dataset. Joints that have confidences lower than 0.6 are considered as occluded points. Since [116] detects 14 joints while the number of joints in the Human 3.6M dataset is 32, we categorized the 32 joints to 14 groups so that the nearby joints can also be treated as missing points. Around 9-10% of the points are marked as missing points as a consequence. The results on *Walking* sequence are shown in Table 3.13. All of the proposed methods give reasonable performance in the existence of occluded points. Compared to the case without missing points, performance of $\mathrm{PR}_{\mathrm{3D}}$ does not degrade much while errors of the trajectory-based methods drastically increase. This indicates the applicability of the proposed method in real-world situations.

3.4 Discussion

The proposed framework have a couple of limitations. Firstly, the performance of Procrustean Regression heavily depends on the initialization performance.

Since the cost function is non-linear due to the alignment constraint, providing good initialization is crucial to avoid bad local minima. When initialization is poorly given, the framework outputs undesirable shapes as shown in Figure 3.7. Developing better initialization algorithm whose solution can be found using convex optimization may increase the performance of overall algorithm especially for the perspective projection. Another drawback of Procrustean Regression compared to EM-PND [59] is that the low-rank regularization is imposed on the aligned shapes, not the pure non-rigid components. Although the alignment constraint eliminates the rigid changes in some extent, it is different from PND in which non-rigid components are obtained by subtracting mean shape from aligned shapes, and this results inferior performance compared to EM-PND.

The flexibility of Procrustean Regression provides the possibility to incorporate various assumptions and theories to the framework. For the data term, besides the reprojection error and the 3D error proposed in this dissertation, various kinds of distance measurements such as Mahalanobis distance [70] can be used considering data distribution of the observations. Additional constraints such as assigning different weights for the noisy data or ignoring the cost for the missing points can easily added as long as they are represented as differentiable functions. For the regularization term, smoothness constraint on feature space similar to [38] may be applied. Therefore, Procrustean Regression provides a general and flexible framework in order for researchers to easily build-up and experiment their own algorithms.

NRSfM is a field of theoretical research area rather than the practicality of it. In real-world scenarios, learning-based approaches or rigid 3D reconstruction algorithms with multiple cameras produces more accurate results. However, the

framework is still practically useful for the case that 3D ground truth dataset cannot be obtained or the case that synchronized multiple cameras cannot be used. The framework can be used to any deformable objects not only in the limited and controlled space, but in general environments.

3.5 Conclusion

We proposed a novel regression framework for NRSfM in this chapter. We argued the generality and flexibility of the proposed framework while maintaining the advantage of rigid/non-rigid separation in Procrustean-distribution approaches. Both orthographic and perspective camera cases can be efficiently handled by the framework, and various regularization strategies can be integrated. Experiment results show that the proposed method gives competitive performance with the state-of-the-art methods while requiring much less running time and memory consumption. The proposed framework also works robustly with the existence of missing points for both random and structured occlusion situations. Moreover, the flexibility of the proposed framework can provide a basis for designing a general NRSfM method for future researches.

Chapter 4

Weakly-Supervised Learning of 3D Human Pose via Procrustean Regression Networks

Although the 3D reconstruction algorithm proposed in Chapter 3 successfully retrieve 3D structures for given 2D observations, it is not a learning-based algorithm. Therefore, we need to run the reconstruction algorithm for each sequence one by one when there are multiple sequences of observations. Moreover, we cannot handle single image reconstruction with the reconstruction algorithm.

To address those problems, we extend Procrustean Regression proposed in chapter 3 so that it can be applied to train neural networks in this chapter. The proposed framework, *Procrustean Regression Network*, is a weakly-supervised learning method which learns to infer 3D structure from 2D inputs without requiring 3D ground truth at the training phase. Procrustean regression network (PRN) applies an NRSfM cost function directly to a loss function of deep neural networks. PRN is the first work that contains low-rank optimization in the loss function of the network. PRN takes consecutive image sequences or 2D point sequences as inputs during the training, and the network learns to estimate 3D

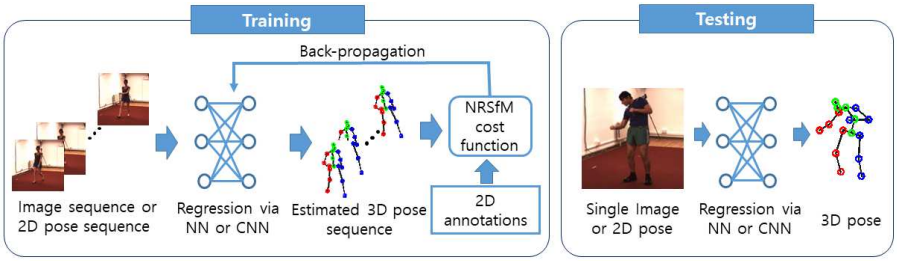


Figure 4.1: Overview of Procrustean Regression Network. During the training phase (left), a neural network takes consecutive image sequences as inputs. Gradients of NRSfM cost function which is based on 2D ground truth as described in detail in Section 4.1 are computed and back-propagated to the network. At test time (right), 3D pose of a single image is estimated via forward propagation.

shape of each input using only 2D ground truth annotations. The reconstruction result of a single image or 2D shape is generated during the test.

The overall procedure of PRN is illustrated in Fig. 4.1. The cost function of PRN and its gradients are explained in Section 4.1, data term and regularization term for PRN are proposed in Section 4.2, and the structure of the network and learning strategies are described in Section 4.3. Experimental results and conclusions are provided in Section 4.4 and Section 4.5 respectively.

4.1 The Cost Function for Procrustean Regression Network

In Section 3.2.1, the cost function of Procrustean regression (PR) is designed. We followed the notations and its meanings of Chapter 3. The cost function of PRN adopts that of PR with slight modifications. One may directly use the gradients of (3.24) to train the neural networks, but estimating the mean shape

for a sequence of multiple inputs can be problematic and impose additional burden to the networks. Hence, the reference shape is excluded from the cost function for PRN, and the reference shape is defined as the mean of the aligned shapes. In other words, the mean shape $\overline{\mathbf{X}}$ in (3.3) is replaced with $\sum_{j=1}^{n_f} \mathbf{R}_j \mathbf{X}_j$. The cost function of PRN can be written as follows:

$$\mathcal{J} = \sum_{i=1}^{n_f} f(\mathbf{X}_i) + \lambda g(\widetilde{\mathbf{X}}). \tag{4.1}$$

The alignment constraint is also changed to

$$\mathbf{R} = \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{i=1}^{n_f} \|\mathbf{R}_i \mathbf{X}_i \mathbf{T} - \sum_{j=1}^{n_f} \mathbf{R}_j \mathbf{X}_j\| \quad \text{s.t.} \quad \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}. \tag{4.2}$$

where $\mathbf{R}$ is concatenation of all rotation matrices, i.e., $\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_{n_f}]$. Let us define $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ as $\mathbf{X} \triangleq [\mathrm{vec}(\mathbf{X}_1), \mathrm{vec}(\mathbf{X}_2), \cdots, \mathrm{vec}(\mathbf{X}_{n_f})]$ and $\widetilde{\mathbf{X}} \triangleq [\mathrm{vec}(\widetilde{\mathbf{X}}_1), \mathrm{vec}(\widetilde{\mathbf{X}}_2), \cdots, \mathrm{vec}(\widetilde{\mathbf{X}}_{n_f})]$ respectively. The gradient of $\mathcal{J}$ with respect to $\mathbf{X}$ while satisfying the constraint (4.2) is

$$\frac{\partial \mathcal{J}}{\partial \mathbf{X}} = \frac{\partial f}{\partial \mathbf{X}} + \lambda \left\langle \frac{\partial g}{\partial \widetilde{\mathbf{X}}}, \frac{\partial \widetilde{\mathbf{X}}}{\partial \mathbf{X}} \right\rangle, \tag{4.3}$$

$\frac{\partial f}{\partial \mathbf{X}}$ and $\frac{\partial g}{\partial \widetilde{\mathbf{X}}}$ are derived once f and g are determined. Again, the most tricky part is calculating the derivative of $\frac{\partial \widetilde{\mathbf{X}}}{\partial \mathbf{X}}$. The derivation process is analogous to Section 3.2.2. As we did in Section 3.2.2, we introduce $\mathbf{Q}_i$ that satisfies $\mathbf{R}_i = \mathbf{Q}_i \hat{\mathbf{R}}_i$ and assume $\mathbf{Q}_i = \mathbf{I}$ at the time of gradient evaluation. Integrating the orthogonality constraint $\mathbf{Q}_i^T \mathbf{Q}_i = \mathbf{I}$ to (4.2) by introducing Lagrange multipliers $\mathbf{\Lambda}_i$ yields

$$\sum_{i=1}^{n_f} \|\mathbf{Q}_i \mathbf{X}_i' - \frac{1}{n_f} \sum_{j=1}^{n_f} \mathbf{Q}_j \mathbf{X}_j'\|^2 + \frac{1}{2} \sum_{i=1}^{n_f} \left\langle \mathbf{\Lambda}_i, \mathbf{Q}_i^T \mathbf{Q}_i - \mathbf{I}_3 \right\rangle. \tag{4.4}$$

Differentiating (4.4) with respect to $\mathbf{Q}_k$ and multiplying $\mathbf{Q}_k^T$ on both sides yields the following equation,

$$\mathbf{Q}_k(\frac{n-1}{n}\mathbf{X}'_k\mathbf{X}_k'^T + \mathbf{\Lambda}_k)\mathbf{Q}_k^T = \frac{1}{n}\sum_{i\neq k}\mathbf{Q}_i\mathbf{X}'_i\mathbf{X}_k'^T\mathbf{Q}_k^T. \tag{4.5}$$

Following the derivation process of (3.14) to (3.16), we get

$$\mathbf{L}^T\mathrm{vec}(\mathbf{Q}_k\mathbf{X}'_k\sum_{i\neq k}\mathbf{X}_i'^T\mathbf{Q}_i^T) = \mathbf{0}. \tag{4.6}$$

Differentiating (4.6) and substituting $\mathrm{vec}(\partial\mathbf{Q}_i) = \mathbf{L}\partial\mathbf{q}_i$ yields,

$$\begin{aligned}&\mathbf{L}^T(\sum_{i\neq k}\mathbf{X}'_i\mathbf{X}_k'^T\otimes\mathbf{I}_3)\mathbf{L}\partial\mathbf{q}_k + \mathbf{L}^T\sum_{i\neq k}(\mathbf{I}_3\otimes\mathbf{X}'_k\mathbf{X}_i'^T)\mathbf{E}\mathbf{L}\partial\mathbf{q}_i\\ &= -\mathbf{L}^T(\sum_{i\neq k}\mathbf{X}'_i\otimes\mathbf{I}_3)\mathrm{vec}(\partial\mathbf{X}'_k) - \mathbf{L}^T\sum_{i\neq k}(\mathbf{I}_3\otimes\mathbf{X}'_k)\mathbf{E}\mathrm{vec}(\partial\mathbf{X}'_i).\end{aligned} \tag{4.7}$$

The index k runs from 1 through n_f, so there are n_f equations made from (4.7). Let $\partial\mathbf{q}$ be a vector $\partial\mathbf{q} = [\partial\mathbf{q}_1^T, \partial\mathbf{q}_2^T, \cdots, \partial\mathbf{q}_{n_f}^T]^T$, and similarly we define $\mathrm{vec}(\partial\mathbf{X}') = [\mathrm{vec}(\partial\mathbf{X}'_1)^T, \mathrm{vec}(\partial\mathbf{X}'_2)^T, \cdots, \mathrm{vec}(\partial\mathbf{X}'_{n_f})^T]^T$. To formulate $\partial\mathbf{q}$ as a function of $\mathrm{vec}(\partial\mathbf{X}')$, we enumerate n_f equations and build a linear system that has the form of

$$\mathbf{A}\partial\mathbf{q} = \mathbf{B}\mathrm{vec}(\partial\mathbf{X}'). \tag{4.8}$$

$\mathbf{A}$ is a $3n_f \times 3n_f$ matrix whose block elements are

$$\mathbf{a}_{ij} = \begin{cases}\mathbf{L}^T(\sum_{k\neq i}\mathbf{X}_k'^T\mathbf{X}_i'^T\otimes\mathbf{I}_3)\mathbf{L} & \text{for} \quad i = j\\ \mathbf{L}^T(\mathbf{I}_3\otimes\mathbf{X}'_i\mathbf{X}_j'^T)\mathbf{E}\mathbf{L} & \text{for} \quad i\neq j\end{cases} \tag{4.9}$$

where $\mathbf{a}_{ij}$ means the submatrix whose rows are from $3i-2$ to $3i$ and columns are from $3j-2$ to $3j$ of $\mathbf{A}$, and i, j are integers range from 1 to n_f. $\mathbf{B}$ is a $3n_f \times 3n_f n_p$ matrix whose block elements are

$$\mathbf{b}_{ij} = \begin{cases}-\mathbf{L}^T(\sum_{k\neq i}\mathbf{X}'_k\otimes\mathbf{I}_3) & \text{for} \quad i = j\\ -\mathbf{L}^T(\mathbf{I}_3\otimes\mathbf{X}'_i)\mathbf{E} & \text{for} \quad i\neq j\end{cases} \tag{4.10}$$

where $\mathbf{b}_{ij}$ means the submatrix whose rows are from $3i-2$ to $3i$ and columns are from $3n_p(j-1)+1$ to $3n_pj$ of $\mathbf{B}$. Then, $\partial\mathbf{q}$ is expressed as

$$\partial\mathbf{q} = \mathbf{A}^{-1}\mathbf{B}\text{vec}(\partial\mathbf{X}'). \tag{4.11}$$

Next, we rewrite (3.20) which is also applicable for this case,

$$(\mathbf{X}_i'^T \otimes \mathbf{I}_3)\mathbf{L}\partial\mathbf{q}_i = \text{vec}(\partial\widetilde{\mathbf{X}}_i - \partial\mathbf{X}_i'). \tag{4.12}$$

Let $\text{vec}(\partial\widetilde{\mathbf{X}}) = [\text{vec}(\partial\widetilde{\mathbf{X}}_1)^T, \text{vec}(\partial\widetilde{\mathbf{X}}_2)^T, \cdots, \text{vec}(\partial\widetilde{\mathbf{X}})_{n_f}^T]^T$, and building linear equations by varying index i from 1 to n_f, we get

$$\mathbf{C}\partial\mathbf{q} = \text{vec}(\partial\widetilde{\mathbf{X}}) - \text{vec}(\partial\mathbf{X}'). \tag{4.13}$$

$\mathbf{C}$ is a $3n_pn_f \times 3n_f$ block diagonal matrix expressed as

$$\mathbf{C} = blkdiag((\mathbf{X}_1'^T \otimes \mathbf{I}_3)\mathbf{L}, (\mathbf{X}_2'^T \otimes \mathbf{I}_3)\mathbf{L}, \cdots, (\mathbf{X}_{n_f}'^T \otimes \mathbf{I}_3)\mathbf{L}) \tag{4.14}$$

where $blkdiag(\cdot)$ is a block-diagonal operator.

Substituting (4.11) to (4.13) yields

$$(\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{I}_{3n_pn_f})\text{vec}(\partial\mathbf{X}') = \text{vec}(\partial\widetilde{\mathbf{X}}). \tag{4.15}$$

Finally, dividing both sides of (4.15) by $\partial\text{vec}(\mathbf{X})$ gives the derivative we need,

$$\frac{\text{vec}(\partial\widetilde{\mathbf{X}})}{\partial\text{vec}(\mathbf{X})} = (\mathbf{C}\mathbf{A}^{-1}\mathbf{B} + \mathbf{I}_{3n_pn_f})\mathbf{D}, \tag{4.16}$$

where $\mathbf{D}$ is a $3n_fn_p \times 3n_fn_p$ block diagonal matrix expressed as

$$\mathbf{D} = blkdiag(\mathbf{T}\otimes\hat{\mathbf{R}}_1, \mathbf{T}\otimes\hat{\mathbf{R}}_2, \cdots, \mathbf{T}\otimes\hat{\mathbf{R}}_{n_f}). \tag{4.17}$$

$\frac{\partial\mathcal{J}}{\partial\widetilde{\mathbf{X}}_i}$ is a $3n_fn_p \times 3n_fn_p$ matrix. In the next section, we will discuss about the design of the functions f and g and their derivatives.

4.2 Choosing f and g for Procrustean Regression Network

In PRN, the network produces 3D position of each joint of the human body by regression. The network output is fed into the cost function, and the gradients are calculated to update the network. For the data term f, we use the reprojection error between the estimated 3D shapes and the ground truth 2D points. We only consider orthographic projection in this chapter, but the framework can be easily extended to perspective projection. In addition, we empirically found that the depth of the 3D shapes easily move towards zero because it usually lowers the rank of the aligned matrix $\tilde{\mathbf{X}}$. To prevent such cases, log barrier term is imposed on the depth of the 3D shapes. Overall, the function f has the following form.

$$f(\mathbf{X}) = \sum_{i=1}^{n_f} \frac{1}{2} \|\mathbf{U}_i - \mathbf{P}_o \mathbf{X}_i\|_F^2 - \eta \log \|\mathbf{P}_z \ddot{\mathbf{X}}\|_F^2, \tag{4.18}$$

where $\mathbf{P}_o = \mathrm{diag}([1, 1, 0])^T$ is a orthographic projection matrix, $\mathbf{U}_i$ is a $2 \times n_p$ 2D observation matrix (ground truth), $\mathbf{P}_z = [0, 0, 1]$ that extracts z-components from the 3D points, $\|\cdot\|_F$ is a Frobenius norm, $\ddot{\mathbf{X}} \triangleq [\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_{n_f}]$, and η is a weight parameter. The gradient of (4.18) is

$$\frac{\partial f}{\partial \mathbf{X}} = \sum_{i=1}^{n_f} \mathbf{P}_o(\mathbf{X}_i - \mathbf{U}_i) - \eta \frac{2}{\|\mathbf{P}_z \ddot{\mathbf{X}}\|_F^2} \mathbf{P}_z \ddot{\mathbf{X}}. \tag{4.19}$$

For a regularization term, we used nuclear norm to impose a low-rankness to the aligned shapes, i.e.,

$$g(\widetilde{\mathbf{X}}) = \|\widetilde{\mathbf{X}}\|_*, \tag{4.20}$$

where $\|\cdot\|_*$ stands for the nuclear norm of a matrix. The subgradient of a nuclear norm is calculated as

$$\frac{\partial g}{\partial \widetilde{\mathbf{X}}} = \mathbf{U}\mathrm{sign}(\mathbf{\Sigma})\mathbf{V}, \tag{4.21}$$

where $\mathbf{U}\mathbf{\Sigma}\mathbf{V}$ is a singular vector decomposition of $\widetilde{\mathbf{X}}$ and $\mathrm{sign}(\cdot)$ is a sign function. $\frac{\partial g}{\partial \widetilde{\mathbf{X}}_i}$ is easily obtained by reordering $\frac{\partial g}{\partial \widetilde{\mathbf{X}}}$.

4.3 Implementation Details

By integrating (4.19) and (4.21) into (4.16), the gradient of 3D shape $\mathbf{X}_i$ with respect to the cost function of NRSfM can be calculated. Then, the entire parameters in the network can also be calculated by back-propagation. We experimented two different structures of PRN in Section 4.4: neural networks and convolutional neural networks. For the neural network structure, inputs are multiple sequences of 2D points, and the network produces 3D position of the input sequences. We used two stacks of residual module [45] of 1024 units as the network structure. We split the prediction part to output the values of x,y coordinates and z coordinates separately as illustrated in Figure 4.2. Since we assume the orthographic camera model in this chapter, we used the ground truth values of x and y coordinates as 2D inputs, which is also used as a ground truth labels when calculating the cost function. Feeding the labels that are the same as the inputs may seem to be implausible, but the 2D inputs should be used as the labels to calculate and propagate the gradients of the cost function. For the solver of fully connected neural networks, we used Adam optimizer [57] with start learning rate of 0.0001.

For the CNNs, sequence of RGB images are fed into the network. For the network structure, single hourglass network module [76] followed by a half of hourglass module is used with the number of feature maps halved from the original implementation. Then, from the features of the final convolutional layers which consists of 256 feature maps of 4×4 size, 3D position of points in

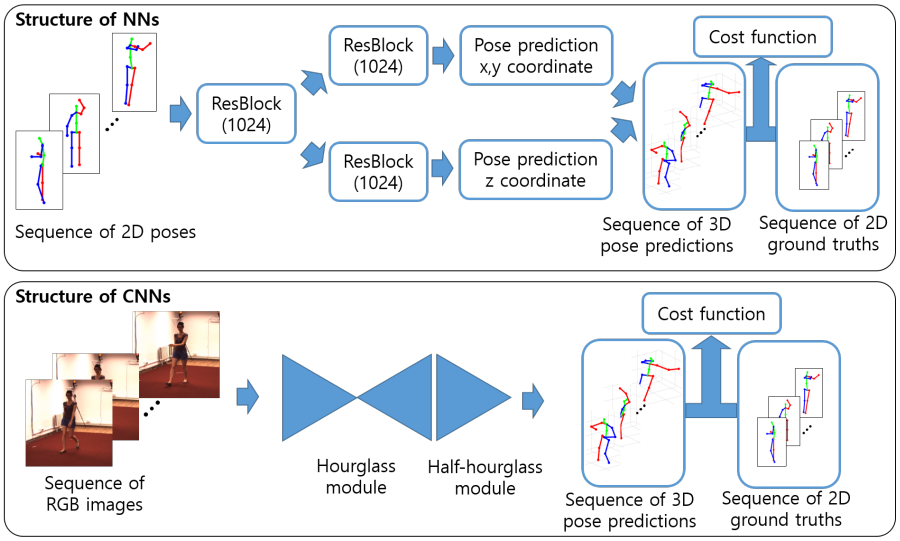


Figure 4.2: The structure of PRN for neural networks (top) and convolutional neural networks (bottom). Note that sequences are within the same batch, so testing can be done using a single pose or RGB image.

the input sequences are regressed, which is the output of the CNNs. The stacked hourglass module is pre-trained with Human 3.6M [49] dataset so that it outputs 2D joints heatmaps as in the original implementation for 2D pose estimation task [76]. This pre-training helps to learn image-related features efficiently and improves the overall performance. We set the start learning rate to 0.00005 for the case of CNNs.

The parameters are empirically set to $\lambda = 0.05$, $\eta = 10^{-7}$ for both neural net structures and CNN structures. The organization of mini-batch is an important task for PRN. A sequence of points or images within the same batch should contain moderate rotations to effectively reconstruct the 3D structure. However, the dataset used for the experiments of PRN, Human 3.6M [49], has images taken from 4 different cameras whose positions are fixed. Hence, we alternately

put the frames or 2D pose taken from different cameras for consecutive frames. This setting is the same as *Setting 2* in Table 3.11 except that the strides between the frames may not one. Large stride values simulate slower rotation than the case when the value of the stride is one. Details about the batch organization for each case in the experiment are explained in Section 4.4.

4.4 Experimental Results

We evaluated the reconstruction performance of human bodies using Human 3.6M [49] dataset. All sequences in the dataset are downsampled to 10fps in all experiments in this section. The performance is measured in normalized error as in the previous section. First, the performance of 3D pose estimation from 2D pose is evaluated. In this experiment, the batch size of the network is set to 128, and the batch consists of 4 different sequences, each of which contains 32 frames from 4 different cameras as explained in Section 4.3. To simulate moderate camera rotation, we set the stride between the previous and next frame to 5. Since all sequences are in 10fps, the interval between the previous and current frames in the same sequence is 0.5 second, which simulates realistic rotation speed for the experiment.

We experimented the effectiveness of the proposed framework by comparing the proposed method with baseline methods. In Table 4.1, the normalized error of the networks trained using only reprojection error term (Reproj), using both reprojection error and the regularization term (Reproj+Reg), and using reprojection error, regularization, log-barrier term (Reproj+Reg+Log) are compared. The performance of the network that trained with ground truth 3D data is also compared (GT). The proposed PRN effectively estimates 3D depth infor-

mation from 2D inputs since the normalized error dramatically decrease when the regularization term is added. The proposed cost function for PRN shows competitive performance even when compared with the model trained with ground truth 3D data. Qualitative results are shown in Figure 4.3. The 3D poses generated from PRN is very similar to the ground truth poses. In general, the estimated depth values from PRN is smaller than the ground truth depths. This is mainly due to the nuclear norm function which tries to make the 3D shapes low-rank and hence tends to make the depth values near zero. Log-barrier function alleviates this phenomenon and helps to generate more plausible results.

Next, we examined the effect of the minibatch organization on the reconstruction performance. First, we measured the reconstruction error varying the number of frames in a sequence. Batch size and the stride between the frames are fixed to 128 and 5 respectively. Hence, there are 128/(*length of a sequence*) different sequences exist in a minibatch. As the length of a sequence gets shorter, the aligned shape matrix, denoted as $\widetilde{\mathbf{X}}$ in Section 4.1, has narrower shape due to the small n_f value. This makes the rank of the aligned shape matrix smaller than the actual rank of the aligned shape matrix with ground truth shapes, and it leads to worsen the reconstruction performance. On the other hand, as the length of a sequence gets longer, i.e., number of sequences in a minibatch gets smaller, the procedure of training neural networks becomes unstable since samples in a minibatch only contains similar samples. To measure the trade-off between those two, we set the length of a sequence in a minibatch to 4,8,16,32, and 64, and we measured the reconstruction performance in each case. The results are shown as a graph in Figure 4.4. It is interesting to see that the error gets smaller with increasing length of a sequence until the length is 32, and the error drastically increases when the length is 64. It can be verified that the aligned

Table 4.1: Normalized errors on the Human 3.6M dataset using 2D pose inputs.

Sequence	Reproj only	Reproj+Reg	Reproj+Reg+Log	GT
Directions	0.3286	0.1660	**0.1620**	0.1220
Discussion	0.3523	0.1579	**0.1500**	0.1293
Eating	0.3847	0.1735	**0.1670**	0.1414
Greeting	0.3423	0.1863	**0.1828**	0.1434
Phoning	0.3849	0.1805	**0.1718**	0.1595
Photo	0.3088	0.1808	**0.1739**	0.1319
Posing	0.3879	0.1807	**0.1711**	0.1274
Purchases	0.4795	0.2377	**0.2322**	0.1934
Sitting	0.5774	0.2934	**0.2916**	0.2462
SittingDown	0.3794	0.1821	**0.1734**	0.1577
Smoking	0.3700	0.2103	**0.2074**	0.1723
Waiting	0.3374	0.1851	**0.1755**	0.1543
WalkDog	0.3229	0.1761	**0.1664**	0.1593
Walking	0.3800	0.1765	**0.1697**	0.1600
WalkTogether	0.3436	0.1877	**0.1821**	0.1573
AVG	0.3800	0.1898	**0.1831**	0.1568

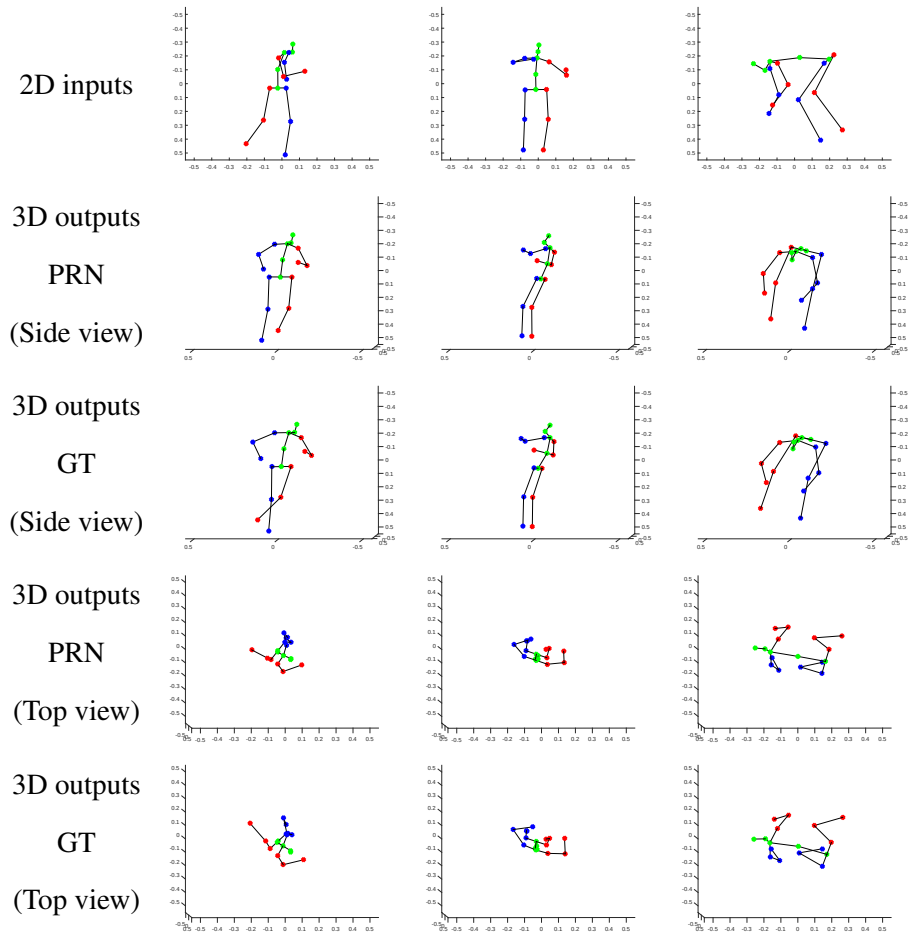


Figure 4.3: Qualitative results of PRN. 2D input points are shown in the first row. The outputs of PRN, and the ground truth (GT) 3D poses are shown in both side view and top view. The outputs of PRN shows very accurate 3D reconstruction results compared to the ground truth.

shape matrix, $\widetilde{\mathbf{X}}$, has advantageous in terms of performance when its shape is close to a square matrix rather than a narrow-shaped one. While there are only 4 different sequences in minibatch when the sequence length is 32, the minibatch contains enough diverse samples to stably train neural networks. However, when the sequence length is increased to 64, the minibatch only contains 2 different sequences, and this leads to provide instable gradients propagated to the neural networks because the gradients of different minibatches may have large variance due to the variations of the sequences.

Another characteristic of the minibatch setting that affects the performance is the interval, or stride in other words, between consecutive frames in a sequence. For instance, if the stride is 5, we pick the 2D inputs every 5 frames from the original sequence. We conducted this experiment to examine the practicability of the proposed method in real-world situations. As mentioned in Section 4.3, larger frame strides means longer interval between the frames in a sequence. Since there are 4 cameras and sequences are recorded in 10fps in the dataset, if we denote the stride as s, the effective speed of camera rotation simulated in the experiment is calculated as

$$(speed\_of\_rotation) = \frac{4 \times s}{10}(\mathrm{sec/rotation}). \tag{4.22}$$

We varied the frame strides to 1,2,5, and 10, and we measured the reconstruction errors, which is illustrated in Figure 4.5. Batch size and the length of a sequence in a minibatch are fixed to 128 and 32 respectively. As the stride gets wider, the inputs to the network contains longer duration of sequences and have larger pose variations within a sequence. This large amount of pose variation makes reconstruction task difficult, and it leads to decrease in performance. As it can be seen in Figure 4.5, the reconstruction error increases as the stride gets wider.

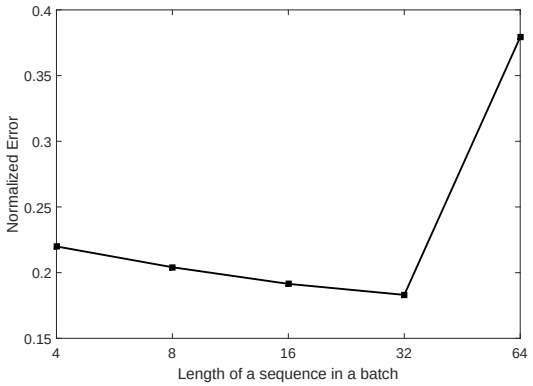


Figure 4.4: Normalized errors with varying length of a sequence in a minibatch.

Nevertheless, the network shows reasonable performance even when the frame stride is set to 10. Normalized error less than 0.21 is still acceptable compared to the errors presented in Figure 4.4. Although one complete rotation in 4 seconds may not seem to be slow enough, the reconstruction performance implies the proposed method can be applicable in real-world scenarios.

Lastly, we trained CNNs using the cost function of PRN which accepts RGB images as inputs. The normalized errors are shown in Table 4.2. As in the experiments using NNs, the performance shows large gap between the cases when the regularization term is used or not. The reconstruction performance of PRN is even close to the performance trained using 3D ground truth data. Therefore, it can be said that CNNs with the regularization term are also able to learn depth information from the images and 2D ground truth training data.

4.5 Conclusion

A novel weakly-supervised learning framework using neural networks and NRSfM cost function is proposed in this chapter. Different from the NRSfM methods,

Table 4.2: Normalized errors on the Human 3.6M dataset using RGB images.

Sequence	Reproj only	Reproj+Reg	GT
Directions	0.3515	0.2244	0.2020
Discussion	0.4003	0.2572	0.2557
Eating	0.4232	0.2583	0.2800
Greeting	0.3842	0.2562	0.2416
Phoning	0.4430	0.3099	0.3099
Photo	0.3560	0.2495	0.2266
Posing	0.4421	0.2938	0.2666
Purchases	0.5328	0.3722	0.3682
Sitting	0.6472	0.4947	0.5017
SittingDown	0.4324	0.2914	0.2883
Smoking	0.4445	0.3224	0.3239
Waiting	0.4015	0.2876	0.2529
WalkDog	0.3689	0.2589	0.2294
Walking	0.4325	0.3002	0.2967
WalkTogether	0.4394	0.3272	0.2860
AVG	0.4343	0.2994	0.2904

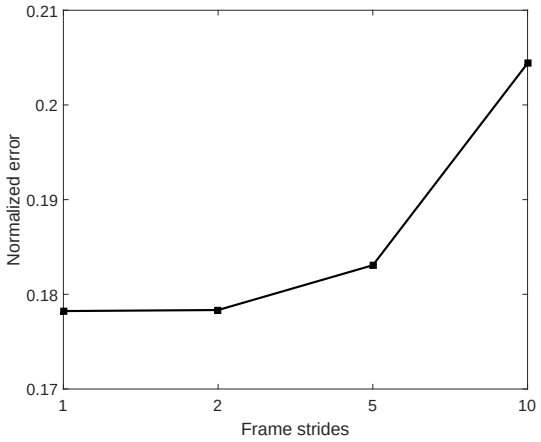


Figure 4.5: Normalized errors with varying length of a sequence in a batch.

PRN teaches how to reconstruct 3D shapes from 2D input for the specific class of objects to neural networks. Although our experiments in this dissertation is limited to 3D Human pose only, the algorithm is also applicable for any kind of non-rigid objects such as human faces or animals. The proposed framework inherits generality and flexibility of Procrustean regression in that various cost function can be designed and easily integrated to the cost function of the network. In addition to the generality of the framework, PRN is theoretically meaningful in that PRN is the first work that establish the connection between NRSfM and neural networks. Low-rank function is also firstly used in the cost function of the neural network in PRN.

More scrutinization to the framework should be studied. Various types of data term and regularization can be designed as in the case of Procrustean Regression. For example, missing 2D points can be handled by modifying the data term, and additional smoothness constraint can be imposed to the regularization term. Extension to the perspective projection is also a future work of PRN. In

addition, PRN can be combined with a semi-supervised learning when only little amount of 3D ground truth is given. Developing a semi-supervised learning method using the cost function of PRN would expand the applicability of PRN.

The limitation of PRN is that the performance heavily depends on how to organize the sequences in a batch during the training. We mostly used the camera setting that simulates fast rotation in the experiments, which may not the realistic situation in real-world. If we make sequences using a single camera only, PRN does not show promising results. This is originated from the difficulty of finding initial rotations. Unlike Procrustean regression, PRN does not take any initialization, so determining initial rotation is difficult for the sequences that contain slow or no rotations.

Chapter 5

Supervised Learning of 3D Human Pose via Relational Networks

In this chapter, we propose a supervised learning framework for 3D HPE. In other words, we are interested in the case when 3D ground truth data is available for training unlike Chapter 3 and Chapter 4. Specifically, we suggest an efficient neural network structure and training strategy that impose occlusion robustness.

Estimating 3D pose of human body joints from 2D joint locations is an under-constrained problem. However, since human joints are connected by rigid bodies, the search space of 3D pose is limited to the range of joints. Therefore, it is able to learn 3D structures from 2D positions, and numerous studies on 2D-to-3D mapping of human body have been conducted. Recently, Martinez et al. [71] proved that a simple fully connected neural network that accepts raw 2D positions as an input gives surprisingly accurate results. Inspired by this result, we designed a network that accepts 2D positions of joints as inputs and generates 3D positions similar to [71].

We designed the network so that it learns the relations among different body

parts. The relational modules for the neural networks proposed in [93] provided a way to learn relations between the components within a neural network architecture. We adopt this relational modules for 3D HPE with a little modification. Specifically, the body joints are divided into several groups, and the relations between them are learned via relational modules in the network. Then, the relational features from all pairs of groups are averaged to generate the final feature vectors that are used for 3D pose estimation. We demonstrate that this simple structure outperforms the fully connected baseline method. By extending the network structure, we also provide the framework that accepts multi-frame inputs to further improve 3D HPE performance.

In addition, we propose a method that can impose robustness to the missing points during the training. The proposed method, relational dropout, simulates the cases when certain groups of joints are missing. To capture the relations among joints within a group, we also designed a hierarchical relational network which further allows robustness to wrong 2D joint inputs. Lastly, we discovered that the proposed structure of the network modified from [71] and the finetuning schemes improve the performance of HPE.

The proposed method achieved state-of-the-art performance in 3D HPE on Human 3.6M dataset [49], and the network can robustly estimate 3D poses even when multiple joints are missing using the proposed relational dropout scheme. The remainder of this chapter is organized as follows. First, the relational networks [93] are reviewed in Section 5.1. Then, the structure of the relational network designed for 3D HPE is proposed in Section 5.2, and its extension to multi-frame inputs is proposed in Section 5.3. Next, relational dropout which impose occlusion robustness for 3D HPE is explained in Section 5.4. Section 5.5 shows experimental results, and Section 5.6 concludes this chapter.

5.1 Relational Networks

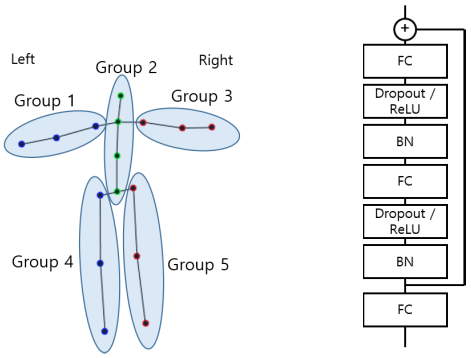
Relation networks (RN) proposed in [93] consists of two parts, the one that does relational reasoning and the other that performs a task-specific inference. The output of the RN is formulated as follows:

$$RN(O) = f(\sum_{(i,j)} g(o_i, o_j)), \tag{5.1}$$

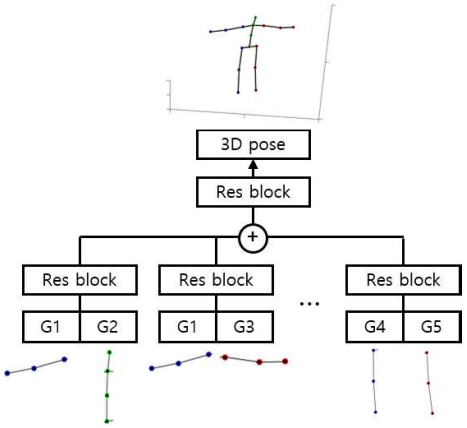
where f and g are functions that are represented as corresponding neural networks, and $O = \{o_1, \cdots o_n\}$ is the set of objects. Pairs of different objects o_i, o_j are fed to the network g, and the relation of all pairs are summed together to generate features that capture relational information. In [93], relational network is applied to visual question answering problems, and features from convolutional neural networks are treated as objects from which relations are implicitly learned. We will treat groups of joints in human bodies as objects to use the relational network for 3D HPE.

5.2 Relational Networks for 3D HPE

We adopt the concept and the structure of the RN to 3D human pose estimation. The network proposed in this chapter takes $2n_{2D}$-dimensional vectors as inputs and outputs $3(n_{3D} - 1)$-dimensional vectors where n_{2D} and n_{3D} are the number of 2D and 3D joints respectively. For 2D inputs, we used (x, y) coordinates of detected joints in RGB images whereas relative positions of (x, y, z) coordinates from the root joint are estimated for 3D pose estimation. In the original RN [93], a neural network module that generates a pairwise relation, $g(\cdot)$, shares weights across all pairs of objects. This weight sharing makes the network learn order-invariant relations. However, this scheme is not applied to our 2D-to-3D



(a) (b)



(c)

Figure 5.1: Overview of the framework. (a) Group configurations used in this chapter. We divided 16 2D input joints to non-overlapping 5 groups each of which corresponds to left/right arms, left/right legs and a torso. (b) The residual module used in this chapter. We adopted the structure suggested in [46]. (c) The structure of the RN for 3D HPE. Features extracted from all pairs of groups are averaged to produce features for pose estimation. Each Resblock in the figure has the same structure shown in (b).

regression of human pose as the following reasons. While original RN tries to capture the holistic relations that does not depend on the position of the objects or order of pairs, the groups on human body represent different parts where order of pairs matters. For instance, if the 2D positions of the left arm and the right arm are switched, the 3D pose should also be changed accordingly. However, the relational features generated will be the same for both cases if the order of pair is not considered. For these reasons, we did not use weight sharing for relational models. The 3D HPE algorithm proposed in this chapter is formulated as

$$S_{3D}(S_{2D}) = f(\frac{1}{n_p}\sum_{(i,j)} g_{i,j}(G_i, G_j)), \tag{5.2}$$

where n_p is the number of pairs, S_{3D}, S_{2D} represents 3D and 2D shape of human body joints respectively, and G_i corresponds to the subsets of 2D input joints belonging to group i. We divide the input 2D joints to non-overlapping five groups as illustrated in Fig. 5.1(a). Total 16 joints are given as an input to the proposed network. Each joint group contains 3 or 4 joints, which we designed so that each group has a small range of variations. Each group represents a different part of a human body in this configuration. In other words, the groups contain joints from left/right arms, left/right legs, or the rest (a head and a torso). Thus, the relational network captures how different body parts are related with each other. All pairs of (i, j) such that $i < j$ are fed to the network and generates features of the same dimension. The mean of the relational features is passed to the next network module that is denoted as $f(\cdot)$ in Eq. 5.2. We empirically found that using the mean of the relational features instead of the sum stabilizes training.

We used ResNet structures proposed in [46] for neural network modules that

are used for relation extraction and 3D pose estimation. The structure of a single module is illustrated in Fig. 5.1(b). A fully connected layer is firstly applied to increase the input dimension to that of a feature vector. Then, a residual network consisting of two sets of batch normalization [47], dropout [101], a ReLU activation function, and a fully connected layer is applied. The overall structure of the proposed network for 3D HPE is illustrated in Fig. 5.1(c).

It can be advantageous if we are able to capture the relations of pairs of individual joints. However, in this case, there are total $\frac{n_{2D}(n_{2D}-1)}{2}$ pairs which makes the network quite large. Instead, we designed a hierarchical relational network in which relations between two joints in a group are extracted within the group. The feature of each group G_k is generated as

$$G_k = \frac{1}{n_{p_k}} \sum_{(i,j)} g^k_{i,j}(P_i, P_j), \tag{5.3}$$

where n_{p_k} is the number of pairs in group k, and P_i, P_j correspond to 2D joints that belong to group k. The generated features are used as an input to the next relational network which is formulated as Eq. 5.2. Empirically, we observe that the hierarchical representation does not outperform a single level relational network, but the structure is advantageous if the relational dropout is applied as described in Section 5.4.

5.3 Extensions to Multi-Frame Inputs

In the previous section, only a single frame is used to predict 3D human pose of the input. If multiple frames including not only the current frame but previous and next frames are used as an input, we can expect better performance by virtue of additional information obtained from multiple frames. We propose

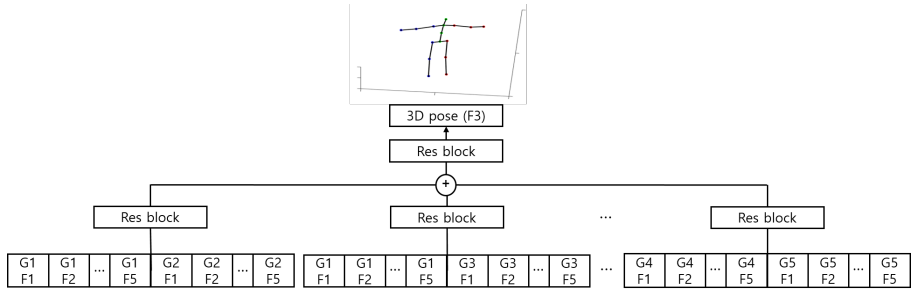


Figure 5.2: The structure of the RN for multi-frame inputs. The joints belong to the same group in all input frames are concatenated to form the new groups. We used five consecutive frames as an input, and 3D pose of the middle input frame (the third frame) is produced as the network output.

a simple extension that accepts 2D joint frames of multiple frames as an input in this section. We used 5 consecutive frames for the multi-frame extension. In other words, 2D pose estimation results of 5 consecutive frames are fed into the neural network. As an output, we predict 3D pose of the middle frame, the third one. Therefore, the network accepts two more frames before and after the target frame.

The structure of the multi-frame relational network is illustrated in Figure 5.2. The number of relational modules in the network remains unchanged. Only the input to each relational module is different from the single-frame relational network. Specifically, instead of forming a group of joints using a single frame, we concatenate the joints in multiple frames that belong to the same group. If we denote five consecutive input frames as F_1, F_2, F_3, F_4, F_5 in a chronological order and G_{ij} as the joints belong to ith group in F_jth frame,

then the 3D HPE task is formulated as

$$S_{3D\_F3}(S_{2D\_F1}, S_{2D\_F2}, S_{2D\_F3}, S_{2D\_F4}, S_{2D\_F5}) \\ = f(\frac{1}{n_p}\sum_{(i,j)} g_{i,j}(G_{i1}, G_{i2}, \cdots, G_{i5}, G_{j1}, G_{j2}, \cdots, G_{j5})). \tag{5.4}$$

When the result of 2D pose estimation in the current frame is inaccurate, the pose estimation results of previous and next frames may provide more reliable results, which is the main advantage of using multi-frame inputs. We verify that this simple extension improves the accuracy of 3D HPE in Section 5.6.

5.4 Relational Dropout

In this section, we propose a regularization method, which we call *'relational dropout'*, that can be applied to relational networks. Similar to dropout [101], we randomly drop the relational feature vectors that contain information on a certain group. In this chapter, we restrict the number of dropping element to be at most 1. Thus, when the number of groups is n_G, among the $\frac{n_G(n_G-1)}{2}$ pairs, $n_G - 1$ relational feature vectors are dropped and replaced with zero vectors when relational dropout is applied. After the mean of the feature vectors are calculated, it is divided by the portion of non-dropping vectors to maintain the scale of the feature vector as in the general dropout method. Concretely, when group k is selected to be dropped, the formulation becomes

$$S_{3D}(S_{2D}|\text{drop} = k) = f(\frac{1}{n_p - n_G + 1}\sum_{(i\neq k, j\neq k)} g_{i,j}(G_i, G_j)). \tag{5.5}$$

Dropping features of a certain group simulates the case that the 2D points belonging to the dropping group are missing. Hence, the network learns to estimate the 3D pose not only when all the 2D joints are visible but also when some

of them are invisible. The relational dropout is applied with the probability of p_{drop} during the training. Since at most one group is dropped, the combinational variability of missing joints is limited. To alleviate the problem, we applied the proposed relational dropout to hierarchical relational networks. In this case, we are able to simulate the case when a certain joint in a group is missing and to simulate various combinations of missing joints. At test time, we simply apply relational dropout to the groups that contain missing points.

5.5 Implementation Details

For the networks used in the experiments, the pose estimator $f(\cdot)$ in the relational networks has fully connected layers of 2,048 dimensions with a dropout probability of 0.5. For the modules $g_{i,j}(\cdot)$ that generates relational feature vector of the pairs G_i and G_j, 1,024 dimensional fully connected layers with a dropout probability of 0.25 are used. Lastly, for the hierarchical relational networks, the modules that generate relations from the pairs of 2D joints consist of 256 dimensional fully connected layers with a dropout probability of 0.1. When the relational dropout is applied during the training, p_{drop} is set to 0.2 for the case that one of the groups of joints is dropped, and it is set to 0.1 when the relational dropout is applied to the hierarchical relational units to drop a single joint.

We used stacked hourglass network [76] to infer 2D joint positions from training and test images. We finetuned the network pre-trained on MPII human pose dataset [9] using the frames of Human3.6M dataset. Mean subtraction is the only pre-processing applied to both 2D and 3D joint positions.

The proposed network is trained using ADAM optimizer [57] with a starting learning rate of 0.001. The batch size is set to 128, and the learning rate is halved

for every 20,000 iterations. The network is trained for 100,000 iterations.

As a final note, we found that finetuning the trained model to each sequence of Human 3.6M dataset improves the estimation performance. During the finetuning, batch normalization statistics are fixed and the dropout probability is set to 0.5 in all modules.

5.6 Experimental Results

We used Human 3.6M dataset [49] to validate the proposed algorithm. The dataset is the largest dataset for 3D HPE, and it consists of 15 action sequences which were performed by 7 different persons. Following the previous works, we used 5 subjects (S1, S5, S6, S7, S8) for training and 2 subjects (S9, S11) for testing. Mean per-joint position error (MPJPE), which is the average of the distances between the ground truths and the predictions of all joints in 3D space, is used as an evaluation metric. We reported MPJPE for two types of alignments: aligning the root joints of the estimated pose and the ground truth pose denoted as *Protocol 1*, and aligning via Procrustes analysis including scaling, rotation, and translation denoted as *Protocol 2*. The proposed method is compared to the recently proposed methods that estimates 3D pose from a single image [85, 71, 29, 18, 126, 74, 130, 105].

To compare the performance of the proposed algorithm to the network that does not use relational networks, we designed a baseline network containing only fully connected layers. The baseline network consists of two consecutive ResBlocks of 2,048 dimensions. Dropout with probability of 0.5 is applied.

The MPJPE of various algorithms using Protocol 1 is provided in Table 5.1. It can be seen that the baseline network already outperforms most of the existing

Method	Direct	Discuss	Eat	Greet	Phone	Photo	Pose	Purchase
Pavlakos et al. [85]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3
Tekin et al. [105]	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6
Zhou et al. [130]	54.8	60.7	58.2	71.4	62.0	65.5	53.8	55.6
Martinez et al. [71]	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1
Fang et al. [29]	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7
Cha et al. [18]	48.4	52.9	55.2	**53.8**	62.8	73.3	52.3	52.2
Yang et al. [126]	51.5	58.9	50.4	57.0	62.1	**65.4**	**49.8**	52.7
FC baseline	50.5	54.5	52.4	56.7	62.2	74.0	55.2	52.0
RN-hier	49.9	53.9	52.8	56.6	60.8	76.1	54.3	51.3
RN	49.7	54.0	52.0	56.4	60.9	74.1	53.4	51.1
RN-FT	49.4	54.3	51.6	55.0	61.0	73.3	53.7	50.0
RN-multiframe	**48.0**	**52.1**	**49.8**	**53.8**	**59.4**	73.3	52.1	**49.6**

Method	Sit	SitDown	Smoke	Wait	WalkD	Walk	WalkT	**Avg**
Pavlakos et al. [85]	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Tekin et al. [105]	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
Zhou et al. [130]	75.2	111.6	64.2	66.1	**51.4**	63.2	55.3	64.9
Martinez et al. [71]	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Fang et al. [29]	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Cha et al. [18]	71.0	89.9	58.2	**53.6**	61.0	43.2	50.0	58.8
Yang et al. [126]	69.2	**85.2**	57.4	58.4	60.1	43.6	**47.7**	58.6
FC baseline	70.0	90.8	58.7	56.8	60.4	46.3	52.2	59.7
RN-hier	68.5	90.9	58.5	56.4	59.3	45.5	50.0	59.2
RN	69.3	90.4	58.1	56.4	59.5	45.6	50.6	59.0
RN-FT	68.5	88.7	58.6	56.8	57.8	46.2	48.6	58.6
RN-multiframe	**67.7**	89.4	**56.6**	54.5	56.7	**42.9**	48.4	**57.2**

Table 5.1: MPJPE (in mm) on Human 3.6M dataset under Protocol 1.

methods, which validates the superiority of the proposed residual modules. The relational networks are trained without applying relational dropouts. The proposed relational network (*RN*) gains 0.7 mm improvements over the baseline on average, and it is further improved when the network is finetuned on each sequence (*RN-FT*), which achieves state-of-the-art performance. Therefore, it is verified that capturing relations between different groups of joints improves the pose estimation performance despite its simpler structure and training procedures than the compared methods. Hierarchical relational networks (*RN-hier*) does not outperform *RN* although it has bigger number of parameters than *RN*. We conjecture the reason to be that it is hard to capture the useful relations in a small number of joints which leads to output poorer features than the ones using the raw 2D positions. Lastly, multi-frame relational network (RN-multiframe) outperforms all single-frame based methods.

The MPJPE using the alignment Protocol 2 is provided in Table 5.2. When shape aligning via Procrustes analysis is applied, our method RN-FT showed superior performance to the existing methods except [126]. Multi-frame relational networks also improves the accuracy of 3D HPE under Protocol 2.

Next, we discuss the effectiveness of the relational dropout for the case of missing joints. The experiments about the relational dropout is conducted on single-frame based methods only. MPJPE for all sequences with various types of missing joints are measured and provided in Table 5.3 and Table 5.4. We simulated 3 types of missing joints following [74], which are 2 random joints (Rand 2), left arm (L Arm), and right leg (R Leg). We consider 3 missing joints for the latter 2 cases including shoulder or hip joints. Note that [74] used different training schemes for experiments on missing joints where six subjects were used for training. For the baseline method that can be applied to the fully connected net-

Method	Direct	Discuss	Eat	Greet	Phone	Photo	Pose	Purchase
Moreno-Noguer [74]	66.1	61.7	84.5	73.7	65.2	67.2	60.9	67.3
Martinez et al. [71]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6
Fang et al. [29]	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2
Cha et al. [18]	39.6	41.7	45.2	45.0	46.3	55.8	39.1	38.9
Yang et al. [126]	26.9	30.9	36.3	39.9	43.9	47.4	28.8	29.4
FC baseline	43.3	45.7	44.2	48.0	51.0	56.8	44.3	41.1
RN-hier	42.5	44.9	44.2	47.4	49.1	57.4	43.9	40.5
RN	42.4	45.2	44.2	47.5	49.5	56.4	43.0	40.5
RN-FT	38.3	42.5	41.5	43.3	47.5	53.0	39.3	37.1
RN-multiframe	37.3	40.8	40.1	42.2	45.9	52.1	38.7	36.7

Method	Sit	SitDown	Smoke	Wait	WalkD	Walk	WalkT	**Avg**
Moreno-Noguer [74]	103.5	74.6	92.6	69.6	71.5	78.0	73.2	74.0
Martinez et al. [71]	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang et al. [29]	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Cha et al. [18]	55.0	67.2	45.9	42.0	47.0	33.1	40.5	45.7
Yang et al. [126]	36.9	58.4	41.5	30.5	29.5	42.5	32.2	37.7
FC baseline	57.0	68.8	49.2	45.3	50.5	38.2	45.0	48.9
RN-hier	56.7	68.5	48.5	44.7	49.4	37.0	43.1	48.1
RN	56.8	68.4	48.4	44.7	49.8	37.6	44.1	48.2
RN-FT	54.1	64.3	46.0	42.0	44.8	34.7	38.7	45.0
RN-multiframe	52.9	64.5	44.6	40.7	44.4	31.8	38.6	43.8

Table 5.2: MPJPE on Human 3.6M dataset under Protocol 2.

Method	None	Rand 2	L Arm	R Leg
FC baseline	59.7	256.1	213.9	222.7
FC-drop	68.6	241.6	98.1	90.6
RN	**59.0**	540.2	314.1	332.8
RN-drop	59.3	218.7	**73.8**	70.6
RN-hier-drop	59.7	**65.9**	74.5	**70.4**

Table 5.3: MPJPE on Human 3.6M dataset with various types of missing joints under Protocol 1.

Method	None	Rand 2	L Arm	R Leg
Moreno-Noguer [74]	74.0	106.8	109.4	100.2
FC baseline	48.9	192.3	153.8	155.7
FC-drop	52.3	159.7	82.0	70.2
RN	48.2	280.7	225.8	214.1
RN-drop	**45.5**	145.3	**62.7**	**55.0**
RN-hier-drop	45.6	**51.4**	63.0	55.2

Table 5.4: MPJPE on Human 3.6M dataset with various types of missing joints under Protocol 2.

work, we assign zero to the value of input 2D joints with the probability of 0.1, which is denoted as *FC-drop*. It imposes robustness to the missing joints compared to the *FC baseline* in which random drop is not applied. When relational dropout is applied to the relational network (*RN-drop*), the model outperforms *FC-drop* in all cases. The model successfully estimates 3D pose when one of the groups in the relational network is missing. Therefore, it shows smaller MPJPE when the left arm or the right leg is not visible. However, when two joints belonging to different groups are missing, the two groups are dropped at the same time, which is not simulated during the training. Thus, *RN-drop* shows poor performance for the case that random two joints are missing. This problem can be handled when relational dropout is applied to the hierarchical relational network. When one joint is missing in a group, relational dropout is applied to hierarchical relational unit within the group. In the case that two or more joints are missing in a group, relational dropout is applied to the group. This model (*RN-hier-drop*) showed impressive performance in all types of missing joints. Another advantage of the relational dropout is that it does not degrade the performance of the case of all-visible joints. It can be inferred that the robustness on missing joints increases as various combinations of missing joints are simulated during the training.

Qualitative results on Human 3.6M dataset are provided in Figure 5.3. Each row simulates different cases of missing joints, none, right leg, left arm, and random 2 joints. The results of *RN*, *FC-drop*, *RN-drop*, *RN-hier-drop* is displayed with ground truth poses. When all joints are visible, all models generate similar poses that are close to the ground truth. On the other hand, *RN* generates inaccurate poses when 2D inputs contain missing points. *RN-drop* provides more accurate results than *FC-drop*, but the model fails when joints of two different

groups are missing. It can be seen that *RN-hier-drop* outputs 3D poses that are similar to the ground truth poses in all cases.

Lastly, we displayed qualitative results on real world images. We used MPII human pose dataset [9] which is designed for 2D human pose estimation. 3D pose estimation results for the relational network (*RN*) and the hierarchical relational network with relational dropouts (*RN-hier-drop*) are provided in Figure 5.4. We first generate 2D pose results for the images and the joints whose maximum heatmap value is less than 0.4 are treated as missing joints for *RN-hier-drop*. As it can be seen in the second and third rows of Figure 5.4, *RN-hier-drop* generates more plausible poses than *RN* when some 2D joints are wrongly detected. The last row shows failure cases which contain noisy 2D inputs or an unfamiliar 3D pose that is not provided during the training.

5.7 Conclusion

In this chapter, we propose a novel method for 3D human pose estimation. The relational network designed for 3D pose estimation showed state-of-the-art performance despite its simple structure. We also proposed the relational dropout which is fitted for the relational network. The relational dropout successfully impose the robustness to the missing points while maintaining the performance of the original network.

The proposed network is flexible in that it allows lots of variations in terms of its structure and group organization. Though the configurations of the groups proposed in this chapter is intuitive and anatomically plausible, analyzing and measuring the effectiveness between different group configurations are remained to future works. One may investigate ways to improve the performance by

changing the structure of the feature generating networks to deeper networks.

The policy of the relational dropout is also flexible and allows a lot of variations. It is able to drop more than one component during training and testing. In this case, occlusion robustness can improve for the case when large portion of missing points exist, but it becomes harder to learn the case of no missing points. The relational dropout can also be applied to other tasks that use relational networks.

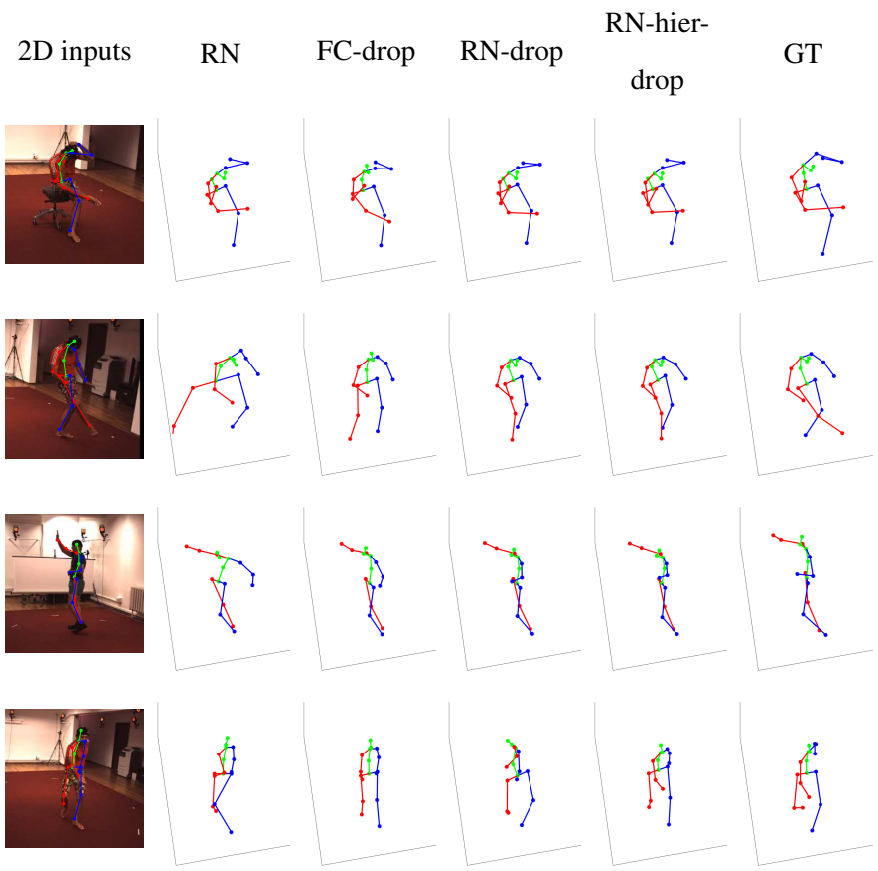


Figure 5.3: Qualitative results on Human 3.6M dataset in various cases of missing joints. For the 2D pose detection results, visible joints are marked as •, and missing joints are marked as ×. Five groups are denoted as green (torso), red (right arm/leg) and blue (left arm/leg).

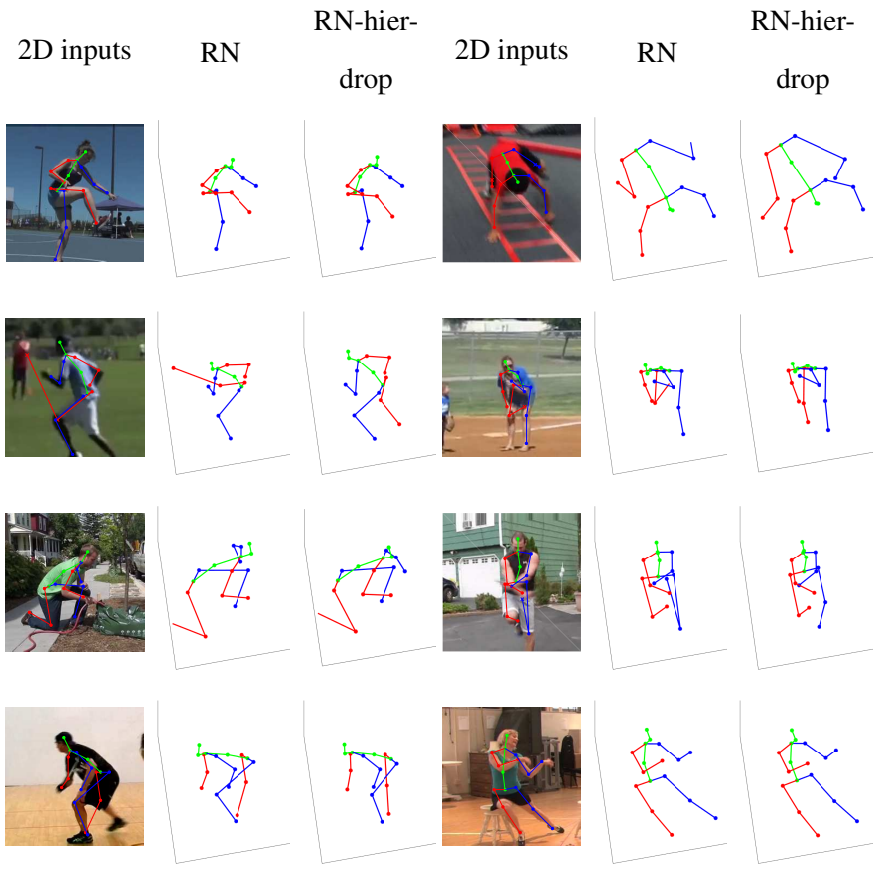


Figure 5.4: Qualitative results on MPII pose dataset.

Chapter 6

Concluding Remarks

In this dissertation, we proposed various methods for 3D HPE which have different settings to solve different problems. All of the methods for 3D HPE are important yet challenging task mainly due to the underconstrainedness of the problem. The proposed methods is applicable to many applications that are related to human-computer interaction, gesture recognition, or virtual reality systems. In this chapter, we give a brief summary of the methods proposed in this dissertation. Then, we discuss limitations and future directions of our research.

6.1 Summary

In this dissertation, methods for retrieving 3D human pose have been proposed in three different viewpoints. First, we tackled 3D reconstruction of human bodies from a sequence of 2D observations. To this end, we proposed a novel regression framework for NRSfM, which is able to reconstruct 3D shapes of human bodies or any non-rigid objects from 2D points correspondences. We insisted the generality and flexibility of the proposed framework while maintaining the ad-

vantage of rigid/non-rigid separation used in previous approaches [59, 62, 61]. Meanwhile, different from the previous methods, both orthographic and perspective camera cases are handled by the proposed framework, and various regularization models can be easily integrated. In addition, Procrustean Regression is a simple and light-weighted algorithm which has advantages in terms of memory and time complexity to the previous works.

Second, we developed a weakly-supervised framework by integrating the cost function modified from Procrustean Regression into the neural networks. The proposed Procrustean Regression Network learns reconstructing 3D shapes of the objects from specific categories using only 2D ground truth of the training data. The network is able to infer 3D structures of the test images or 2D inputs without supervision of 3D ground truth points at the training time. While sequences of training samples are given within a minibatch during the training, testing can be done in a single input using simple feed-forward operations. This is the first research that explicitly connects NRSfM cost function and neural nets, so it provides a new direction for future research of NRSfM.

Third, we proposed a supervised method for 3D human pose estimation. The relational network designed for 3D pose estimation , which learns relational features between different body parts, achieved state-of-the-art performance while maintaining the network structure small and simple. The regularization method designed to be fitted for the relational networks, relational dropout, impose the robustness to the missing points during the training. The model trained with relational dropout showed only little amount of performance degradation when 15%-20% of the inputs are missing.

6.2 Limitations

All of the methods proposed in this dissertation are focused to infer 3D structures of human skeletons, rather than dense 3D mesh of human bodies. The methods that reconstructs 3D mesh of human bodies from RGB images [12, 120, 53] gives richer representation to understand than the methods that infers 3D position for small number of body joints. Although the 3D reconstruction method can process dense datasets that contains thousands of point correspondences, running time of the algorithm is much longer than the learning-based methods. It is also not an easy task to obtain dense correspondences of human bodies. The weakly-supervised and supervised learning methods can be extended to predict more body joints if appropriate datasets are given, but it is still far from reconstructing 3D mesh that are composed of thousands of 3D points.

Another limitation of the proposed methods is that the methods are targeted to predict 3D pose of a single person. Single-person pose prediction methods can be easily extended to multi-person cases by combining the framework with state-of-the-art object detectors [89, 67]. Nevertheless, considering running time and robustness of algorithms, an algorithm that estimates 3D poses of multiple people in a single shot is preferable.

Lack of datasets that provides 3D ground truth of human pose is also a limitation that weakens the generality of the proposed algorithms. Publicly available 3D human pose datasets [72, 49, 97, 51] obtained 3D ground truth using motion capture system or using multiple cameras. Thus, the images provided in the dataset has consistent backgrounds, and only a small number of people are appeared in the datasets. A model learned from these data may work poorly for

the images that contains arbitrary backgrounds and poses that are not appeared in the datasets.

Now, we summarize the limitations of each algorithm proposed in this dissertation. The 3D reconstruction method proposed in Chapter 3 provides flexible optimization framework. However, due to the non-linearity of the cost function, initialization algorithm heavily affects the performance of the proposed method. Inferiority to the performance of state-of-the-art algorithm under orthographic projection is also a weakness of the proposed method. The weakly-supervised learning method proposed in Chapter 4 suggested a learning algorithm for 3D HPE using only 2D ground truth data, but the input sequences for learning should contain moderate rotations to successfully learn the 3D structures. More analysis and understanding about the cost function and the effect of minibatch configuration should be needed to further improve the performance and the practicality of the algorithm. Lastly, the supervised learning method proposed in Chapter 5 effectively learns 2D-to-3D mapping of human poses using 3D ground truth data. However, the proposed method tends to be overfitted on the training data, which lowers the generality of the proposed method. Small amount of the improvements in terms of performance over the baseline methods is also a limitation of the algorithm.

6.3 Future Directions

As the last remark of this dissertation, we discuss the future directions of the research in terms of fully automatic accurate human body reconstruction from RGB images, which is the ultimate goal we want to achieve through the research done in this dissertation.

First, while only small number of joints in human bodies are used in most of the experiments in this dissertation, dense reconstruction of human bodies is essential to precise human body modeling. There are publicly available parameterized human body models [69, 52] which can be used to convert human skeleton to 3D mesh models. There are a few recently proposed approaches that learns parameters of the models using CNNs [12, 120, 53]. Following these approaches and combining the regularization from Procrustean Regression may further improve the quality of human mesh reconstruction.

Second, a semi-supervised learning scheme combining supervised and weakly-supervised learning proposed in chapter 4 can improve 3D HPE performance. 2D human pose datasets contain large variations in terms of the pose and the backgrounds. Using both 2D and 3D datasets may take advantage of generalization and scalability. The weakly-supervised approach in chapter 4 only targets for the case that have only 2D point ground truths. The method may be useful for semi-supervised settings if it can be applied to the model that trained using 3D ground truth training sets.

Lastly, for the sake of generating dense correspondences of human bodies in RGB images, automatic point correspondence algorithm is needed. Shape alignments may also applicable to finding 2D correspondences by using it as a regularizer in the optical-flow based algorithms.

Bibliography

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2006.

[2] A. Agudo, L. Agapito, B. Calvo, and J. Montiel. Good vibrations: A modal analysis approach for sequential non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1558–1565, 2014.

[3] A. Agudo and F. Moreno-Noguer. Learning shape, motion and elastic models in force space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 756–764, 2015.

[4] A. Agudo and F. Moreno-Noguer. Simultaneous pose and non-rigid shape with particle dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2179–2187, 2015.

[5] A. Agudo and F. Moreno-Noguer. Recovering pose and 3d deformable shape from multi-instance image ensembles. In *Asian Conference on Computer Vision*, 2016.

[6] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Sequential non-rigid structure from motion using physical priors. *IEEE transac-*

tions on pattern analysis and machine intelligence, 38(5):979–994, 2016.

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

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

[9] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[10] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1014–1021. IEEE, 2009.

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

[12] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[13] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Computer Vision, 2009 IEEE 12th Interna-*

tional Conference on, pages 1365–1372. IEEE, 2009.

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

[15] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[17] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016.

[18] G. Cha, M. Lee, J. Cho, and S. Oh. Deep pose consensus networks. *arXiv preprint arXiv:1803.08190*, 2018.

[19] C.-H. Chen and D. Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *CVPR*, volume 2, page 6, 2017.

[20] 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*, 117(11):1549–1559, 2013.

[21] 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*, 117(3):226–246, 2016.

[22] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision*, pages 628–644. Springer, 2016.

[23] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4715–4723, 2016.

[24] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

[25] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3041–3048, 2013.

[26] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[27] X. Fan, K. Zheng, Y. Lin, and S. Wang. Combining local appearance and holistic view: Dual-source deep neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1347–1355, 2015.

[28] H. Fang, S. Xie, Y.-W. Tai, and C. Lu. Rmpe: Regional multi-person pose estimation. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, 2017.

[29] H. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu. Learning knowledge-guided pose grammar machine for 3d human pose estimation. *arXiv*

preprint arXiv:1710.06513, 2017.

[30] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003.

[31] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International journal of computer vision*, 61(1):55–79, 2005.

[32] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[33] J. Flynn, I. Neulander, J. Philbin, and N. Snavely. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5515–5524, 2016.

[34] K. Fragkiadaki, M. Salas, P. Arbelaez, and J. Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2014.

[35] M. Gadelha, S. Maji, and R. Wang. 3d shape induction from 2d views of multiple objects. *arXiv preprint arXiv:1612.05872*, 2016.

[36] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016.

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

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

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

[40] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[41] A. Grinciunaite, A. Gudi, E. Tasli, and M. den Uyl. Human pose estimation in space and time using 3d cnn. In *European Conference on Computer Vision*, pages 32–39. Springer, 2016.

[42] B. C. Hall. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer, 2015.

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

[44] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[45] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[46] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[47] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[48] C. Ionescu, F. Li, and C. Sminchisescu. Latent structured models for human pose estimation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2220–2227. IEEE, 2011.

[49] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2014.

[50] S. Johnson and M. Everingham. Clustered pose and nonlinear appearance models for human pose estimation. pages 12–1.

[51] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):190–204, Jan 2019.

[52] H. Joo, T. Simon, and Y. Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8320–8329, 2018.

[53] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[54] A. Kanazawa, D. W. Jacobs, and M. Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[55] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[56] A. Kar, S. Tulsiani, J. Carreira, and J. Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1966–1974, 2015.

[57] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[58] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 3, 2017.

[59] 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*, pages 1280–1287, 2013.

[60] M. Lee, J. Cho, and S. Oh. Consensus of non-rigid reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4670–4678, 2016.

[61] M. Lee, J. Cho, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

[62] 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*, pages 1550–1557, 2014.

[63] K. Li, J. Yang, and J. Jiang. Nonrigid structure from motion via sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1401–1413, Aug 2015.

[64] S. Li and A. B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[65] S. Li, W. Zhang, and A. B. Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2848–2856, 2015.

[66] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[67] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

1353, 2010.

[69] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015.

[70] P. C. Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[71] J. Martinez, R. Hossain, J. Romero, and J. J. Little. A simple yet effective baseline for 3d human pose estimation. In *International Conference on Computer Vision*, volume 1, page 5, 2017.

[72] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*, 2017.

[73] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.

[74] F. Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1561–1570. IEEE, 2017.

[75] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.

[76] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[77] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[78] K. Onishi, T. Takiguchi, and Y. Ariki. 3d human posture estimation using the hog features from monocular image. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE, 2008.

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

[80] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stošić, and M. Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *International Journal of Computer Vision*, 96(2):252–276, 2012.

[81] H. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *Computer Vision–ECCV 2010*, pages 158–171. Springer, 2010.

[82] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3d trajectory reconstruction under perspective projection. *International Journal of Computer Vision*, 115(2):115–135, 2015.

[83] S. Park, J. Hwang, and N. Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *European Conference on Computer Vision*, pages 156–169. Springer, 2016.

[84] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3d human pose estimation. pages 7307–7316, 2018.

[85] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1263–1272. IEEE, 2017.

[86] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. *arXiv preprint arXiv:1704.04793*, 2017.

[87] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[88] M. Rayat Imtiaz Hossain and J. J. Little. Exploiting temporal information for 3d human pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[89] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[90] G. Rogez, J. Rihan, S. Ramalingam, C. Orrite, and P. H. Torr. Randomized trees for human pose detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[91] G. Rogez, P. Weinzaepfel, and C. Schmid. Lcr-net: Localization-classification-regression for human pose. In *CVPR 2017-IEEE Confer-*

ence on Computer Vision & Pattern Recognition, 2017.

[92] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3009–3016. IEEE, 2011.

[93] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[94] M. Sanzari, V. Ntouskos, and F. Pirri. Bayesian image based 3d pose estimation. In *European Conference on Computer Vision*, pages 566–582. Springer, 2016.

[95] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *European conference on computer vision*, pages 406–420. Springer, 2010.

[96] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[97] 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*, 87(1):4, Aug 2009.

[98] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Computer Vision and Pattern*

Recognition, 2006 IEEE Computer Society Conference on, volume 2, pages 2041–2048. IEEE, 2006.

[99] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh. Separable spatiotemporal priors for convex reconstruction of time-varying 3d point clouds. In *European Conference on Computer Vision*, pages 204–219. Springer, 2014.

[100] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh. Kronecker-markov prior for dynamic 3d reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2017.

[101] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[102] X. Sun, J. Shang, S. Liang, and Y. Wei. Compositional human pose regression. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 2, page 7, 2017.

[103] M. Tatarchenko, A. Dosovitskiy, and T. Brox. *Multi-view 3D Models from Single Images with a Convolutional Network*, pages 322–337. 2016.

[104] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[105] B. Tekin, P. Marquez Neila, M. Salzmann, and P. Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *International Conference on Computer Vision (ICCV)*, number EPFL-CONF-230311, 2017.

[106] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[107] D. Tome, C. Russell, and L. Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. *CVPR 2017 Proceedings*, pages 2500–2509, 2017.

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

[109] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[110] S. Tulsiani, A. Kar, J. Carreira, and J. Malik. Learning category-specific deformable 3d models for object reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):719–731, 2017.

[111] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, volume 1, page 3, 2017.

[112] J. Valmadre and S. Lucey. General trajectory prior for non-rigid reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1394–1401. IEEE, 2012.

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

[114] S. Vicente, J. Carreira, L. Agapito, and J. Batista. Reconstructing pascal voc. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 41–48. IEEE, 2014.

[115] G. Wang, H.-T. Tsui, and Z. Hu. Structure and motion of nonrigid object under perspective projection. *Pattern recognition letters*, 28(4):507–515, 2007.

[116] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[117] R. White, K. Crane, and D. A. Forsyth. Capturing and animating occluded cloth. In *ACM Transactions on Graphics (TOG)*, volume 26, page 34. ACM, 2007.

[118] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances In Neural Information Processing Systems*, pages 540–550, 2017.

[119] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. *Single Image 3D Interpreter Network*, pages 365–382. 2016.

[120] D. Xiang, H. Joo, and Y. Sheikh. Monocular total capture: Posing face, body, and hands in the wild. *arXiv preprint arXiv:1812.01598*, 2018.

[121] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *European Conference on Computer Vision (ECCV)*, 2018.

[122] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, 2006.

[123] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1075–1082. IEEE, 2005.

[124] J. Xie, R. Girshick, and A. Farhadi. *Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks*, pages 842–857. 2016.

[125] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[126] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2018.

[127] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[128] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[129] D. Zhang, J. Han, Y. Yang, and D. Huang. Learning category-specific 3d shape models from weakly labeled 2d images. In *Proc. CVPR*, pages 4573–4581, 2017.

[130] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *IEEE International Conference on Computer Vision*, 2017.

[131] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016.

[132] X. Zhou, M. Zhu, G. Pavlakos, S. Leonardos, K. G. Derpanis, and K. Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[133] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

[134] Y. Zhu, M. Cox, and S. Lucey. 3d motion reconstruction for real-world camera motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1–8. IEEE, 2011.

초 록

RGB 영상에서의 사람 자세 추정 방법은 컴퓨터 비전 분야에서 중요하며 여러 어플리케이션의 기본이 되는 기술이다. 사람 자세 추정은 동작 인식, 인간-컴퓨터 상호작용, 가상 현실, 증강 현실 등 광범위한 분야에서 기반 기술로 사용될 수 있다. 특히, 2차원 입력으로부터 3차원 사람 자세를 추정하는 문제는 무수히 많은 해를 가질 수 있는 문제이기 때문에 풀기 어려운 문제로 알려져 있다. 또한, 3차원 실제 데이터의 습득은 모션캡처 스튜디오 등 제한된 환경하에서만 가능하기 때문에 얻을 수 있는 데이터의 양이 한정적이다. 본 논문에서는, 얻을 수 있는 학습 데이터의 종류에 따라 여러 방면으로 3차원 사람 자세를 추정하는 방법을 연구하였다. 구체적으로, 2차원 관측값 또는 RGB 영상을 바탕으로 3차원 사람 자세를 추정, 복원하는 세 가지 방법–3차원 복원, 약지도학습, 지도학습–을 제시하였다.

첫 번째로, 사람의 신체와 같이 비정형 객체의 2차원 관측값으로부터 3차원 구조를 복원하는 비정형 움직임 기반 구조 (Non-rigid structure from motion) 알고리즘을 제안하였다. 프로크루스테스 회귀 (Procrustean regression)으로 명명한 제안된 프레임워크에서, 3차원 형태들은 그들의 정렬된 형태에 대한 함수로 정규화된다. 제안된 프로크루스테스 회귀의 비용 함수는 3차원 형태 정렬과 관련된 제약을 비용 함수에 포함시켜 경사 하강법을 이용한 최적화가 가능하다. 제안된 방법은 다양한 모델과 가정을 포함시킬 수 있어 실용

적이고 유연한 프레임워크이다. 다양한 실험을 통해 제안된 방법은 세계 최고 수준의 방법들과 비교해 유사한 성능을 보이면서, 동시에 시간, 공간 복잡도 면에서 기존 방법에 비해 우수함을 보였다.

두 번째로 제안된 방법은, 2차원 학습 데이터만 주어졌을 때 2차원 입력에서 3차원 구조를 복원하는 약지도학습 방법이다. 프로크루스테스 회귀 신경망 (Procrustean regression network)로 명명한 제안된 학습 방법은 신경망 또는 컨볼루션 신경망을 통해 사람의 2차원 자세로부터 3차원 자세를 추정하는 방법을 학습한다. 프로크루스테스 회귀에 사용된 비용 함수를 수정하여 신경망을 학습시키는 본 방법은, 비정형 움직임 기반 구조에 사용된 비용 함수를 신경망 학습에 적용한 최초의 시도이다. 또한 비용함수에 사용된 저계수 함수 (low-rank function)를 신경망 학습에 처음으로 사용하였다. 테스트 데이터에 대해서 3차원 사람 자세는 신경망의 전방전달(feed forward)연산에 의해 얻어지므로, 3차원 복원 방법에 비해 훨씬 빠른 3차원 자세 추정이 가능하다.

마지막으로, 신경망을 이용해 2차원 입력으로부터 3차원 사람 자세를 추정하는 지도학습 방법을 제시하였다. 본 방법은 관계 신경망 모듈(relational modules)을 활용해 신체의 다른 부위간의 관계를 학습한다. 서로 다른 부위의 쌍마다 관계 특징을 추출해 모든 관계 특징의 평균을 최종 3차원 자세 추정에 사용한다. 또한 관계형 드랍아웃(relational dropout)이라는 새로운 학습 방법을 제시해 가려짐에 의해 나타나지 않은 2차원 관측값이 있는 상황에서, 강인하게 동작할 수 있는 3차원 자세 추정 방법을 제시하였다. 실험을 통해 해당 방법이 2차원 관측값이 일부만 주어진 상황에서도 큰 성능 하락이 없이 효과적으로 3차원 자세를 추정함을 증명하였다.

주요어: 3차원 사람 자세 인식, 비정형 움직임 기반 구조, 관계 신경망, 프로크루스테스 회귀, 3차원 복원, 딥러닝, 약지도학습

학번: 2014-30814