2014

MONOCULAR POSE ESTIMATION AND SHAPE RECONSTRUCTION OF QUASI-ARTICULATED OBJECTS WITH CONSUMER DEPTH CAMERA

Mao Ye
University of Kentucky, maoye.ustc@gmail.com

Recommended Citation

Ye, Mao, "MONOCULAR POSE ESTIMATION AND SHAPE RECONSTRUCTION OF QUASI-ARTICULATED OBJECTS WITH CONSUMER DEPTH CAMERA" (2014). *Theses and Dissertations--Computer Science*. 25.
https://uknowledge.uky.edu/cs_etds/25

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Mao Ye, Student

Dr. Ruigang Yang, Major Professor

Dr. Miroslaw Truszczynski, Director of Graduate Studies

MONOCULAR POSE ESTIMATION AND SHAPE RECONSTRUCTION OF QUASI-ARTICULATED OBJECTS WITH CONSUMER DEPTH CAMERA

DISSERTATION

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in the College of Engineering
at the University of Kentucky

By
Mao Ye
Lexington, Kentucky

Director: Dr. Ruigang Yang, Professor of Computer Science
Lexington, Kentucky 2014

ABSTRACT OF DISSERTATION

MONOCULAR POSE ESTIMATION AND SHAPE RECONSTRUCTION OF QUASI-ARTICULATED OBJECTS WITH CONSUMER DEPTH CAMERA

Quasi-articulated objects, such as human beings, are among the most commonly seen objects in our daily lives. Extensive research have been dedicated to 3D shape reconstruction and motion analysis for this type of objects for decades. A major motivation is their wide applications, such as in entertainment, surveillance and health care. Most of existing studies relied on one or more regular video cameras. In recent years, commodity depth sensors have become more and more widely available. The geometric measurements delivered by the depth sensors provide significantly valuable information for these tasks. In this dissertation, we propose three algorithms for monocular pose estimation and shape reconstruction of quasi-articulated objects using a single commodity depth sensor. These three algorithms achieve shape reconstruction with increasing levels of granularity and personalization. We then further develop a method for highly detailed shape reconstruction based on our pose estimation techniques.

Our first algorithm takes advantage of a motion database acquired with an active marker-based motion capture system. This method combines pose detection through nearest neighbor search with pose refinement via non-rigid point cloud registration. It is capable of accommodating different body sizes and achieves more than twice higher accuracy compared to a previous state of the art on a publicly available dataset.

The above algorithm performs frame by frame estimation and therefore is less prone to tracking failure. Nonetheless, it does not guarantee temporal consistent of the both the skeletal structure and the shape and could be problematic for some applications. To address this problem, we develop a real-time model-based approach for quasi-articulated pose and 3D shape estimation based on Iterative Closest Point (ICP) principal with several novel constraints that are critical for monocular scenario. In this algorithm, we further propose a novel method for automatic body size estimation that enables its capability to accommodate different subjects.

Due to the local search nature, the ICP-based method could be trapped to local minima in the case of some complex and fast motions. To address this issue, we explore the potential of using statistical model for soft point correspondences association. Towards this end, we propose a unified framework based on Gaussian Mixture Model for joint pose and shape estimation of quasi-articulated objects. This method achieves state-of-the-art performance on various publicly available datasets.

Based on our pose estimation techniques, we then develop a novel framework that achieves highly detailed shape reconstruction by only requiring the user to move naturally in front of a single depth sensor. Our experiments demonstrate reconstructed shapes with rich geometric details for various subjects with different apparels.

Last but not the least, we explore the applicability of our method on two real-world applications. First of all, we combine our ICP-base method with cloth simulation techniques for Virtual Try-on. Our system delivers the first promising 3D-based virtual clothing system. Secondly, we explore the possibility to extend our pose estimation algorithms to assist physical therapist to identify their patients movement dysfunctions that are related to injuries. Our preliminary experiments have demonstrated promising results by comparison with the gold standard active marker-based commercial system. Throughout the dissertation, we develop various state-of-the-art algorithms for pose estimation and shape reconstruction of quasi-articulated objects by leveraging the geometric information from depth sensors. We also demonstrate their great potentials for different real-world applications.

KEYWORDS: Pose Estimation, Shape Reconstruction, Articulated Objects, Depth Sensor, Non-rigid Registration

Author's signature: Mao Ye

Date: December 17, 2014

MONOCULAR POSE ESTIMATION AND SHAPE RECONSTRUCTION OF QUASI-ARTICULATED OBJECTS WITH CONSUMER DEPTH CAMERA

By
Mao Ye

Director of Dissertation: Ruigang Yang

Director of Graduate Studies: Miroslaw Truszczynski

Date: December 17, 2014

TO MY BELOVED FAMILIES

My grandparents Zaisheng Ye and Yuhua Huang

My parents Yongming Ye and Huiping Hou

My brother Liang Ye

My wife Ziyu Jia

ACKNOWLEDGMENTS

For these years, I have received a tremendous amount of help and support from my advisor, my colleagues, my friends and my families. Without them, I will not have been able to finish this dissertation and earn my Ph.D. degree.

I would like to express my first sincere appreciation to my advisor, Dr. Ruigang Yang, for introducing me into the exciting area of computer vision and for years of support and guidance throughout my Ph.D. journey. I am so grateful that I have been able to conduct research on topics that I am really interested in under Dr. Yang's supervision. The inspirations and encouragement I have received from Dr. Yang have had significantly impacts on my ways of thinking, exploring and doing. I am more than fortunate to have been able to walk through my Ph.D. journey with him.

I also want to thank all my committee members, Dr. Fuhua Cheng, Dr. Sen-ching Cheung, Dr. Nathan Jacobs and Dr. Qiang Ye. I really appreciate their efforts on guiding my study and on my dissertations.

I have also been so lucky to share my Ph.D. life with a group of lovely people. They are all so creative, inspiring to work with and so nice to get along with. We worked and played together for all these memorable years. They have all contributed to my work and my life. I want to express my thankfulness to all of them, in particular Liang Wang, Xianwang Wang, Xinyu Huang, Miao Liao, Qing Zhang, Jizhou Gao, Chenxi Zhang, Bo Fu, Hui Lin, Changpeng Ti, Yajie Zhao, Chao Du, Yongwook Song, Yin Hu, Yan Huang, Xinan Liu and Yigong Zhang.

Most importantly, my most sincere gratitudes are for all my family members who have offered me unconditional love throughout my life, in particular my grandparents, my parents, my brother and my wife. They have always been the most significant part of my life, my greatest and strongest support whenever I was down. Everything good in my life, I owe to them.

TABLE OF CONTENTS

Acknowledgments . . . iii

Table of Contents . . . v

List of Tables . . . viii

List of Figures . . . ix

Chapter 1 Introduction . . . 1
- 1.1 Historic Review . . . 1
- 1.2 Motivation . . . 2
- 1.3 Contributions . . . 5
- 1.4 Structure . . . 7

Chapter 2 Related Work . . . 9
- 2.1 Marker-less Motion Capture . . . 9
  - 2.1.1 Color-based Methods . . . 10
  - 2.1.2 Depth-based Methods . . . 10
  - 2.1.3 Template Adaptation . . . 14
- 2.2 Shape Reconstruction . . . 15
  - 2.2.1 Model-based Approaches . . . 15
  - 2.2.2 Model-free Approaches . . . 16

Chapter 3 Preliminaries: Shape and Pose Representations . . . 18
- 3.1 Articulated Deformation Model . . . 18
- 3.2 Linear Blend Skinning . . . 21
- 3.3 Twist-Based Representation . . . 22
- 3.4 Incremental Pose Update . . . 23

Chapter 4 Data Driven Human Pose Estimation . . . 26
- 4.1 Overview . . . 27
- 4.2 Point Cloud Segmentation and Denoising . . . 28
- 4.3 Model Based Motion Estimation . . . 31
  - 4.3.1 Point Cloud Alignment . . . 32
  - 4.3.2 Nearest-Neighbor Search in Low-Dimensional Subspace . . . 33
- 4.4 Experimental Results . . . 34
  - 4.4.1 The Effectiveness of Denoising . . . 36
  - 4.4.2 The Effectiveness of Pose Refinement . . . 36
  - 4.4.3 Viewpoint Independency . . . 37
  - 4.4.4 Failure Detection . . . 38
  - 4.4.5 Database Dependency . . . 39

4.4.6 Qualitative Evaluation using Kinect 39
4.4.7 Body-size Invariance 40
4.5 Conclusion 41

Chapter 5 Real-Time Model-based Pose and Shape Estimation with Closest Point Strategy 42
5.1 Pose Tracking via Closest Points 43
5.2 Surface Geometry Adaptation 49
5.3 Body Size Adaptation 52
5.4 The Tracking Pipeline 55
5.5 Experimental Results 55
5.6 Limitations and Conclusion 58

Chapter 6 Real-Time Model-based Pose and Shape Estimation with Probabilistic Correspondences Association 60
6.1 The General Probabilistic Model 60
6.2 The Tracking Algorithm 62
6.3 Monocular Scenario 64
6.4 Template Shape Adaptation and Initialization 65
6.4.1 Limb Lengths Estimation 66
6.4.2 Geometric Shape Adaptation 67
6.4.3 Template Initialization 69
6.5 Experiments 69
6.5.1 Tracking Accuracy Evaluations 72
6.5.2 Shape Adaptation Accuracy 74
6.5.3 Application in Non-rigid Shape Registration 75
6.6 Conclusion and Future Work 77

Chapter 7 Dynamic Human Avatar Reconstruction 78
7.1 Confidence-based Frame Selection 81
7.1.1 Confidence Evaluation 81
7.1.2 Frame Selection Strategy 84
7.2 Template Guided Key Frame Merging 86
7.2.1 Embedded Deformation Model 87
7.2.2 Key Frame Upsampling 90
7.2.3 Template Guided Initial Alignment 91
7.2.4 Global Non-rigid Alignment 93
7.3 Results 97
7.3.1 Quantitative Evaluations 97
7.3.2 Human Subject Examples 98
7.3.3 Comparisons 101
7.3.4 Computational Performance 102
7.3.5 Post Reconstruction Applications 103
7.4 Limitations and Conclusion 104

Chapter 8 Applications: Virtual Try-On and Smart Health 106
8.1 Virtual Try-On with a Single Commodity Depth Camera 106
8.1.1 Related Work 108
8.1.2 Final Image Composition 113
8.1.3 Results and Discussions 114
8.1.4 Conclusion 116
8.2 Pose Estimation for Movement Dysfunction Identification in Smart Health . 117
8.2.1 Data Collection and Processing 120
8.2.2 Preliminary Experimental Results 121
8.2.3 Conclusion 125

Chapter 9 Conclusion and Future Work 126
9.1 Contributions 126
9.2 Future Work 128

Bibliography 132

Vita 149

LIST OF TABLES

5.1 The constraints and parameters used for point correspondence construction. . . . 49

6.1 The three datasets we use for quantitative evaluations. 71
6.2 The parameter settings for our experiments. 72

8.1 Comparison of flexion-extension angle measurement for a jig configuration of $-45.6^\circ \pm 0.1^\circ$. 123
8.2 Reliability and agreement of systems for calculating peak joint angles during a squat. 124
8.3 95% limits of agreement (LOA) and the bias of the motion capture systems. . . . 125

LIST OF FIGURES

3.1 Illustration of articulated motion of an articulated object with two segments. Image adapted from [87]. See the text for explanation. 19

3.2 A typical hierarchical representation of human body [14]. 20

3.3 Our template model consisting of the mesh and the underlying skeleton (kinematic structure). A rigid body transformation is represented via a twist ξ. The vector ω is the orientation of the rotation axis corresponding to ξ. 21

4.1 The outline of our processing pipeline. The leftmost image is a typical depth map, we define distance thresholds to cut the subject out. It goes through a number of processing stages, generating the estimated skeleton embedded in the input point cloud. 28

4.2 Comparison of original LOP and our modified LOP applied on a point cloud from depth sensor. from left to right (a) the input; (b) LOP with support radius of $0.2m$; (c) LOP with support radius of $0.1m$; (d) with our modified LOP. . . . 29

4.3 Comparison of point cloud smoothing using bilateral filtering and our modified LOP algorithm. From left to right: input; after Bilateral Filtering; after our modified LOP. 31

4.4 Visualization of the three principal axes (color-coded) of different point clouds. They are quite stable across different viewpoints. Two different poses are shown. (The right-most one is rendered from back-view with the entire right leg occluded.) 32

4.5 The mean estimation error of our approach on Stanford's public test dataset that consists of 28 sequences of motions,compared with the results reported in [52] using Hill Climbing combined with Evidence Propagation (HC+EP) . . 34

4.6 Comparison of estimation errors of three methods: using our entire pipeline, without smoothing and directly using neighbor pose without refinement. The results demonstrate the necessity of such processing components. 36

4.7 Viewpoint independency test. First row: examples of synthetic inputs (see text for explanation). Second row shows quantitative evaluations. 37

4.8 An example of our failure case (left), for which the most similar pose in our database (right) exhibits a very large difference in pose. 38

4.9 Estimation errors, with and without failure detection and recovery, of the test frames where failure cases happen and our failure detection and recovery method takes effects. 38

4.10 Estimation errors of our approach with a set of motion samples which is subsampled from our original database, with different ratios. 39

4.11 Examples of estimation results using pose tracking algorithms in [97]((a) and (c)) and our method ((b) and (d)), from depth images captured by Kinect. . . . 40

4.12 Examples of our experiments on two subjects with very different body sizes. The two images on the left are results on a 1.9m adult, while the other two on the right are results on a 1.6m child. 40

5.1 The diagram of our pose tracking system. During initialization, our system takes in one single frame of depth map and adjusts the limb lengths of the template. For each frame of the live data, we perform pose and shape adaptation to adjust the template for better personalization. 42

5.2 2D examples illustrating the effectiveness of two of our constraints. The arrows connect a point to its closest point on the other surface. In (a), the dash lines representing occluded regions. The visible points (black circles) attempt to move to the right as desired; while the invisible points (the gray circles) tend to move the surface upward and could lead to incorrect local optima if not excluded. In (b), since p_1 and p_2 belong to two different segments, our relaxed bijective consistency constraint will reject $< p_1, v_1 >$, so will the original version. However, the pair $< p_3, v_2 >$ will also be rejected by the original version. By contrast, our relaxed version will accept it, and therefore, could guide the tracker more effectively. 45

5.3 Edge point correspondences. (a) The false edge points due to self-occlusion are not used as they do not correspond to true shape boundary. (b) The dash lines represents projection, and the red dash circle indicates the range of candidate points to match v based on 2D measurement. With closest point in 3D, v is matched to a noise p_1. With closest point in 2D, it is matched to p_2. With our two stage approach, it can be mapped to the point p_3 on the correct target. However, it cannot handle the situation that projection of p_1 or p_2 lies in the red circle. In most cases, it is better than direct 3D or 2D closest point strategy. 47

5.4 A male (left) and a female (right) example showing the surface estimation results. The first row shows the template without displacements overlayed on the input, while the second row shows the one with displacements. The highlighted regions demonstrate its effectiveness. 51

5.5 An example showing results of our body size adaptation algorithm. In the first row, the input mesh, template mesh and template skeletons are shown together. (a) the initial step of the algorithm. (b) results with only pose estimation. (c) the adapted results. Notice the improvement for the joints indicated by the arrows. The second row compares our adapted skeleton, with the skeleton estimated by Kinect SDK [4]. (d) our skeleton (green) and the Kinect skeleton side by side. (e) and (f) comparisons of these two skeletons with close-up views for the legs and the feet, respectively. Discussions are provided in the text. 57

5.6 A failure case for our pose tracking. The torso of the template (right) it not properly rotated to fit to the input surface (middle). In addition, the right arm is problematic as well due to lack of constraint. 58

6.1 Our novel algorithm effectively estimates the pose of articulated objects using one single depth camera, such as human and dogs, even with challenging cases. 70

6.2 Quantitative evaluation of our tracker on the SMMC-10 dataset [52] with two error metrics. Notice that in (b), although the method by Ye et al. [133] achieves comparative accuracy, their reported computation time is much higher. 71

6.3 Quantitative evaluations of our tracker's accuracy with comparisons with the state of the arts. 73

6.4 Visualization of example tracking results on the publicly available datasets. . . 73

6.5 Visual comparison of our results (second row) with KinectSDK [4] (first row) on some relatively complex poses. 74

6.6 Example results of our algorithm applied to dog data. 75

6.7 Visual results of our shape adaptation algorithm. (a) and (c) are results of only pose estimation (male and female respectively), while (b) and (d) are results with shape adaptation during tracking. Input meshes are overlaid on each result. Notice the adjustment of limb lengths, in particular the arms and feet of the male subject. 76

6.8 Results of applying our algorithm for shape registration. The first row shows the initial state (input) for our algorithm. The second row shows the overlaid shapes after registration. Our algorithm automatically handles substantial shape and pose variations. 77

7.1 Taking a dynamic sequence (moving subject + potentially moving camera) as input , our method automatically selects a sparse subset of frames (left) that contains relatively rich and reliable geometric information and reconstructs a quality watertight model (middle left). The reconstructed model is animatable and can be used for various interesting applications, including pose manipulation (middle right) and space-time resolution enhancement for 3D videos (right). 79

7.2 A single mesh from the SCAPE database is rigged and skinned and used as the template for our pose estimation. The mesh is segmented into sub-regions for coverage estimation. Each of the right two images shows sub regions without overlap. We use a combination of these two segmentations. 80

7.3 Frames with large motion blur (right image) might be selected if $\{c_m^t(k)\}$ in Equation 7.6 is excluded. 85

7.4 The original input meshes and our upsampled meshes. The upsampled meshes are smoother, contain more reliable geometric information, have small holes filled both through fusion with nearby frames. 89

7.5 A typical example of the improvement through the global alignment of discriminative body parts. 92

7.6 An upsampled key frame aligned to the reference pose with our approach is shown on the top left. Close-up views of the two circled regions for various alignment approaches are shown to demonstrate the effectiveness of our template guided initial alignment. Some problematic regions produced by the alternative approaches are highlighted by green rectangles. 93

7.7 An example of our proxy model and two models reconstructed with and without our adaptive part selection scheme. The over-smoothed proxy model is only used to assist our global alignment. The close-up views of the two reconstructed models show that some geometric details are corrupted when all key frames are used for reconstruction. By contrast, the adaptive part selection scheme reduces the chance of loosing details. 94
7.8 Quantitative evaluation of our method on a mannequin under different poses. . . 97
7.9 Our results on four different subject data. For each example, the set of automatically selected key frames are shown on the top (Section 7.1). The second row shows the key frames with our template guided initial alignment (Section 7.2.3). The globally aligned key frames are in the third row (Section 7.2.4). The final water-tight models without and with coloring are shown in the last two rows. 100
7.10 A visual comparison of our result dynamic input with models scanned using KinectFusion and Shapify.me with the subject holds static. 101
7.11 The computational time is calculated over our four subject data and provided here as per frame measurement, except those marked as *once*. 102
7.12 Several examples of adapting the reconstructed models for various tasks. (a) The model is skinned into a novel pose. (b) The model is made higher and slimmer. (c) The model is used for space-time resolution enhancement of 3D video. 103

8.1 Our Virtual Try-On system takes the RGBD data in (a) as input and captures the pose and body shape of the user accurately as shown in (b). It then realistically simulates virtual clothing on the user, so the user can examine the appearance of personalized virtual clothing, as illustrated in (c) and (d). Our system robustly handles a wide range of human motions and shape variations. . 107
8.2 The diagram of our virtual try-on system. The left part is the same as in Figure 5.1 and is provided here for completeness. After performing pose and shape adaptation, the adjusted template is used by our cloth simulator to drive the virtual apparel. Then a composite image of the virtual apparel and the input are delivered to the user to provide virtual try-on experience. 108
8.3 Sample final results on both male and female subjects. The results shown in the last column are generated using our near real-time simulation engine while the others are generated with higher quality offline simulation engine. 115
8.4 The angles of interests for three different medical conditions in lower extremity. From left to right: (a) hip adduction angle for patellofemoral pain (PFP); (b) hip flexion angle for Anterior cruciate ligament reconstruction (ACL); and (c) knee adduction for knee osteoarthritis (knee OA). 118
8.5 The diagram of our marker-less motion capture system for lower extremity joint angle assessment. 120
8.6 Our testing jig in sagittal (left) and frontal (right) view. 121
8.7 The flexion-extension angle measured by the marker-based system and our marker-less system. Positive angle denotes extension. 122

8.8 Ensemble curves (± standard deviation) of joint angles calculated by the marker-based and marker-less systems for a slow squat motion. The patterns of motion are similar for both the markerless and marker-based system. 123

Chapter 1 Introduction

Qusi-articulated objects are one of the most general categories of objects in our daily lives. One particular example would be human beings. Reconstruction and analysis of the motion and shape of this category of objects have many applications in real life, such as in entertainment, health care and surveillance. The research has achieved extraordinary advances in recent years. However, the study, in particular on movements, has been going on for more than 2000 years.

1.1 Historic Review

Human beings has a long history of interests in understanding and analyzing the movements of animals and humans, dating back to the studies of gait of animals by the ancient Greek philosopher Aristotle (384-322 B.C.E) [12]. For centuries, art has also been a significantly important driving force that advanced the studies, as evidenced by the detailed analysis of human anatomy by Leonardo da Vinci (1452-1519) [99]. The concept of *Kinamtic trees* that has been and is still serving as a fundamental model to describe movements of humans and animals could be found in his sketchbooks. Later on, the scientist Giovanni Alfonso Borelli (1608-1679), often called "the father of biomechanics", applied the analytical and geometrical methods, developed by Galileo Galilei (1564-1642) in the field of mechanics to provide quantitative measurements [22]. The invention of chronophotography at the later half of 19th century brought a new way to analyze motions. A major contribution was the collaborated research on physics, human anatomy and locomotion by brothers Ernst Heinrich Weber (1795-1878), Wilhelm Eduard Weber (1804-1891), and Eduard Friedrich Weber (1806-1871) [127] . Other famous work built upon chronophotography include those by the French astronomer Pierre Janssen (1824-1907), the French scientist Etienne-Jules Marey (1830-1904), the British-born Eadweard Muybridge (1830-1904) and Albert Londe (1858-1917). With the development of computer technologies in 20th century, the research on motion analysis and biomechanis stepped into a new era, in particular in marker-based pose tracking with its origin in the work by G. Johannsson [67]. Since then, human movement has been actively studied in Biomechanis, Computer Graphics and Computer Vision. A majority of recent research focuses on marker-less motion capture and extraordinary achievements have been made. A more detailed historic review can be found in [99]. With the computational power dramatically boosted in the past two decades, digitized modeling of human shapes has also attracted great attentions of the scientists and extensive research has also been conducted [30].

1.2 Motivation

Traditionally, most computer vision techniques operate on 2D images or videos for motion and shape capture [30, 84]. Great successes have been achieved with multi-camera systems [15, 16, 38, 51, 81, 105, 109]. Most of these techniques rely on prior models that are either hand designed or created with 3D scanner, for exmple based on Structured Light [74]. Most of the time, the 3D scanners can only capture a partial view of the subject at a time. Therefore, extensive research in the areas of Computer Graphics and Geometric Modeling has been conducted on building a complete model out of a set of partial scans [20,28,36,78,91,123]. Despite of the high level of details provided, the 3D scanners can only capture static scenes and is not able to cope with motions. The emergence of real-time range sensors, such as SwissRanger [3], PMD [6] and Microsoft Kinect [4], provide a new opportunity. Specifically, depth cues provided by these sensors overcome several major difficulty in 2D measurements, such as ambiguity due to perspective projection and sensitivity to lighting conditions. Although with lower resolution, it has the advantage of real-time dense geometric measurement over traditional 3D scanners. On the other hand, it provides the possibility to combine knowledge in these two areas to develop new algorithms taylored for depth sensors. Besides, the availability of such devices is increasing rapidly in our daily lives, as they are becoming more and more affordable, such as the Microsoft Kinect sensor [4].

The research presented in this dissertation embraces the new opportunity and explores the potential to use commodity depth sensors for motion and shape estimation of quasi-articulated objects, which is one of the most commonly seen object categories in our daily lives. More specifically, our research is motivated mainly by the following thoughts.

1. Although the commercial marker-bases systems, such as Vicon [8], has delivered highly accurate measurements, the attachment of markers suffers from several limitations. First of all, markers and the specific suits required can significantly alter the pattern of locomotions [43]. Secondly, it is time consuming to setup the markers and to process the recorded data that is sometimes necessary. My research thus focuses on marker-less motion capture that allows nature movements.

2. Existing techniques, especially those with surrounding cameras, are mostly applicable only in laboratory environments [38, 51, 121]. Techniques that can operate in

people's home environments are desired. Therefore, my research is focused on the scenario of a monocular setup.

3. Algorithms that directly operate on 3D data have been well studied in related fields, such as Computer Graphics and Geometry Processing. However they usually deal with data of substantially higher quality than those provided by current depth sensors. It has been observed that directly applying these techniques could be problematic [34]. Consequently, our techniques are taylored for relatively low quality data provided by commodity depth sensors.

4. Techniques for scanning general static objects are available and are still being actively studied. One could use a hand-held camera [91] or body scanner (e.g., Cyberware body scanner [2]). However our experience shows that it is tedious and error-prone to fully scan an object with a hand-held camera using the state-of-art technique [91]. And a body scanner can only scan static objects. More critically, it is unnatural and usually impractical for human or animals to remain completely static during the acquisition process. Driven by these thoughts, my research aims mainly at dynamic scenes.

Despite all the benefits brought by the depth sensors, there are several challenges that need to be addressed. Similar to video cameras, a depth sensor captures only a partial view at a time. Consequently, there is at least 50% data missing in each observation. The occlusion still introduces substantial ambiguity for motion and shape estimation. The huge high dimensional space of human poses further increases the difficulty. Besides, existing depth cameras generally suffer from various degrees of noises, as well as low resolution, compared to high quality body scanners. It poses significant challenges for shape reconstruction, especially if rich geometric details are desired.

1.3 Contributions

In this dissertation, we propose three algorithms for monocular pose estimation and shape reconstruction of articulated objects using a single commodity depth sensor. These three algorithms achieve shape reconstruction with increasing levels of granularity and personalization. We then further develop a method for highly detailed shape reconstruction based on our pose estimation techniques.

The first algorithm we develop overcomes the challenges caused by occlusions and high dimensionality by utilizing a motion database and statistical dimensionality reduction techniques [133]. The motion data is acquired with an active marker-based motion capture system, namely Vicon [8]. Dimensionality reduction techniques is then applied to built a subspace where nearest neighbor search is performed to identify a most similar exemplar in our data for an input depth map. Through non-rigid point cloud registration, we reconstruct the missing data due to occlusions from which the pose can be reconstructed. The algorithm is capable of accommodating different body sizes and achieves more than twice higher accuracy compared to a previous state of the art on a publicly available dataset [52].

Performed in a frame by frame fashion, the above algorithm is more robust then temporal tracking. Nonetheless, it does not guarantee temporal consistency of the both the skeletal structure and the shape. It could be problematic in scenarios where consistency is desired or even critical, such as in health care applications. To address this problem, we develop a real-time model-based approach for pose and 3D shape estimation based on

Iterative Closest Point (ICP) principal with several novel constraints that are critical for monocular setup [132]. To achieve personalization, we propose a novel method for automatic body size estimation that enables its capability to accommodate different subjects.

Despite the good performance of the ICP-based technique, its local search strategy could lead to a local solution in the case of some complex and fast motions. To address this issue, we use statistical model for soft point correspondences association and develop a probabilistic framework for real-time simultaneous pose and shape estimation [134]. This method achieves state-of-the-art performance on various publicly available datasets [52,53, 62]. Experiments demonstrate that the proposed method can better accommodate complex and fast motions compared to previous works. Moreover, we show that this technique can be used to register across a set of shape collections where no correspondences are known beforehand. It will then enable statistical shape analysis or automatic rigging for animation purpose.

All the above algorithms focus more on the motion estimation and have limited capability for shape estimation. An overall shape without rich geometric details can be estimated by fitting a database exemplar [133] or by adapting a generic template [132, 134]. In order to deliver personalized detailed models, we then develop a novel framework on top of our pose estimation techniques [134] that achieves highly detailed shape reconstruction. It allows the user to move naturally in front of the cameras, as opposed to the static settings in other previous work [78]. Our experiments demonstrate reconstructed shapes with rich geometric details for various subjects with different apparels.

Last but not the least, we demonstrate the effectiveness of our techniques in two different applications: Virtual Try-On and Smart Health. For this first application, we combine our ICP-based pose and shape estimation techniques with cloth simulation to deliver the first promising 3D-based virtual clothing system [132]. In the Smarth Health application, we explore the possibility to extend our pose estimation algorithms to assist physical therapist to identify their patients movement dysfunctions that are related to injuries. Our preliminary experiments have demonstrated promising results by comparison with the gold standard active marker-based commercial system [103].

1.4 Structure

The reminder of this dissertation is structured as follows. Chapter 2 provides discussion of existing works mainly on pose and shape estimation using depth information. Some of those techniques are developed concurrently with ours presented in this dissertation. Our data-driven approach is detailed in Chapter 4. Since it utilized a motion database captured with Vicon [8], we first discuss the procedure to build this database. The algorithm that combines pose detection and pose refinement is then presented. The performance is quantitatively evaluated on a publicly available dataset [52]. In Chapter 5, we move to the model-based approach under the Iterative Closest Point (ICP) framework. The effectiveness of this technique is demonstrated through the Virtual Try-On application discussed in Section 8.1 in Chapter 8. Our development of the model-base algorithm with probabilistic correspondence association is presented in Chapter 6. Extensive experiments are conducted to demonstrate the performance both quantitatively and qualitatively. The details of our novel system for detailed shape reconstruction is provided in Chapter 7. Experimental results on various subjects are presented to show the quality of our reconstruction. The two applications on Virtual Try-On and Smart Health are described in Section 8.1 and

Section 8.2 of Chapter 8 respectively. Chapter 9 concludes with outlook of several future research possibilities.

Chapter 2 Related Work

2.1 Marker-less Motion Capture

Human motion capture has been a highly active research area in computer vision and graphics, due in part to its many applications. Such applications span fields as diverse as security surveillance, medical diagnostics, games, movies and sports. Traditional marker-based motion capture systems provide a feasible solution, however such systems carry the burden of specially designed equipment or suits, which is inconvenient for many practical applications. Therefore, non-invasive marker-less approaches are the main focus of research in recent years. There has been several detailed surveys in this topic using color inputs [46,83,84,96,99] or depth inputs [63]. Generally, methods for marker-less motion capture can be categorized into three groups, namely generative approaches, discriminative approaches and hybrid approaches. Generative approaches normally fit a template, either parametric or non-parametric, to the observed data through optimization or filtering [49]. By contrast, discriminative approaches attempt to learn a mapping between the observations, e.g. the image or features extracted, and the target pose. Then the pose of the given image(s) is estimated based on the learned mapping function. Hybrid approaches combines the complementary characteristics of both categories of approaches. A common way is to use discriminative component for initialization and then use the generative component to refine the pose through model fitting. In the following, based on this categorization, methods based on color inputs are briefly reviewed and depth-based approaches are discussed in more details.

2.1.1 Color-based Methods

From color inputs, popular cues used for motion capture include color, texture, silhouettes, edges and motions [49]. In general, multiple cues are considered and integrated together to achieve good performance. Techniques have been develop to estimate human pose from monocular image or video data in generative [23,107,119], discriminative [106,120] or hybrid fashio [47, 101]. “However, they are not always robust, in part because image data is inherently noisy and in part because it is inherently ambiguous” [85,95]. Using surrounding camera setup, the issue can be dramatically mitigated and impressive results have been achieved [16,38,39,110,121]. Most of these approaches rely on a generic [110] or personalized [16, 38, 39, 121] model as a prior and perform both pose estimation in a generative fashion. Normally the shape of the prior model is adapted to the subject in the process, in particular in the setting of performance capture [38, 121]. Although substantial improvements have been made on the accuracy and robustness of techniques using color inputs, the inherent pose ambiguity and appearance diversity still pose significant challenges. In recent years, the emergence of commodity range sensors drives the community to develop methodologies that can effectively utilize the depth information.

2.1.2 Depth-based Methods

Discriminative Approaches

Existing discriminative approaches based on depth information either perform body part detection by identifying salient points of the human body [94, 138], or rely on classifiers or regression machines trained offline to infer the joint locations [55, 104, 116]. Plagemann et al. [94] detects geodesic extrema on surface meshes generated from depth images. The assumption is that these extrema normally coincide with salient points on the human body. These extrema are then classified into different body parts using local shape descriptors centered from normalized de;th images at these extrema. Zhu et al. [138] detects a set of key features points from depth images that represent positions of anatomical landmarks, and track them over time based on a probabilistic inferencing algorithm. These key-points are then kinematically self retargeted to estimate the pose of the person.

Shotton et al. [104] uses an extensive amount of synthetic data ($\geq$ 300K) to train a Random Forest classifier for body part segmentation from a single depth image using simple pair-wise depth features.The joint locations are subsequently estimated using mean shift algorithm. This algorithm serves as the foundation of the great commercial success of the Microsoft Kinect [4] and is considered as a breakthrough. Also based on Random Forest, Hough forests to be specific), Girhick et al. [55] by voting from each raw depth pixel on multiple joint locations. Outliers are then removed and density estimation is applied to infer the locations of both the visible and occluded joints. Both of these two algorithms are very efficient during prediction and runs at real time. Taylor et al. [116] directly infers correspondences between depth images and a parametrized human model. Offline optimization is then performed for pose estimation.

Discriminative approaches show impressive performance both in terms of efficiency and robustness to the diversity of poses. However, a major limitation is the lack of temporal consistency and stability in the estimation results as normally these approaches operates in a frame-by-frame fashion. As mentioned in Section 1.3, these characteristics are desired in some scenarios. Moreover, discriminative approaches are mostly learning based and its accuracy is generally lower than generative approaches and might not be sufficient for applications such as in medical area. Last but not the least, normally a large amount of training data is required which might not be available for objects other than human, such as animals.

Generative Approaches

The generative approaches fit a template, either parametric or non-parametric, to the observed data, mostly with variants of ICP. Pekelny et al. [92] presents one of the first approaches for pose and surface estimation of articulated rigid objects from ToF depth images. They assume an initial segmentation on the first frame is given, along with the knowledge of underlying kinematic structure. Hierarchical ICP is used to align an accumulated model, which originally is the first frame, to a new depth frame to aggregate more observations. The approach runs at around 0.5 frames per second. Knoop et al. [72] propose a fusion framework that combines 2D and 3D input to construct correspondences that used in ICP for pose estimation. The approach is specifically designed for human body and cannot well accommodate occlusions. Bleiweiss et al. [21] also relies on hierarchical ICP to fit a model to the observation and then use the estimated pose to drive a virtual avatar. Fribog et al. [48] move the computationally expensive local optimization to the GPU for speedup. A major limitation of the ICP-based methods are the sensitivity to local minima. To remedy the problem, Ganapathi et al. [53] introduces free space constraint into their Dynamic Bayesian Network (DBN) that models the motion states. Good performance has been demonstrated with this approach. Yet their over-simplified cylindrical template cannot capture the subject's shape and only allow a limited capability for shape adaptation. Ye et al. [131] reduces the ambiguity caused by occlusion by using multiple depth sensors. High resolution pre-scanned body models are fitted to the mult-view observations for pose estimation.

Although prior models are generally used in generative approaches, the exponential combinations of joint configurations lead to an extremely large pose space where global search is generally computationally prohibited. Therefore, most existing approaches rely on local optimization which, however, is sensitive to local minima. In our first model-based algorithm [132] (Chapter 5), we propose several novel constraints that are critical for monocular data to guide the local optimization. A more advance solution is our algorithm based on probabilistic correspondence association [134] (Chapter 6) that is more capable of avoiding local minima.

Hybrid Approaches

The complementary characteristics of these two sets of approaches have been combined to achieve higher accuracy. Similar to our data-driven approach [133], Baak et al. [13] also relies on a database to identify a exemplar with similar pose and subsequently refine the pose. Instead of using PCA, they adopt the idea similar to [94] that extracts features corresponding to salient points for database lookup. Specifically, they stack the first five geodesic extrema as the index that is used to locate the most similar pose in a database consisting of 50K poses. ICP is then used to align the database exemplar to the input to refine the pose. Lately, Helten et al. [62] extended this algorithm to obtain personalized tracker that can handle larger body shape variations and achieves real-time performance. However in order to deal with the missing data in monocular setup, they rely on visibility test to select candidate points for correspondences construction, and therefore could not effectively handle body rotations. Ganapathi et al. [52] adopts the body part detector proposed by Plagemann et al. [94] to guide their generative tracker within a Dynamic Baysian Network (DBN). By combining this DBN framework with the discriminative classifier developed by Shotton et al [104], Wei et al. [128] present a system that achieves high accuracy and real-time performance. Specifically, they use the generative component for tracking and trigger the discriminative component when tracking failure is detected for reinitialization.

2.1.3 Template Adaptation

During pose tracking, especially in the case of generative tracker, the consistency body shapes between the template and the subject plays an important role. Therefore, in most of the approaches that involve a mesh model, surface estimation couples with or follows the pose estimation, in order to capture the surface geometry of the subject. One option is to use pre-scanned personalized models [16, 38, 51, 121]. The scans can be captured with one [129] or multiple depth sensors [117]. The requirement of personalized mesh models is normally fulfilled with high precision 3D scanners and consequently limits the methods' applicability. Another options is to adapt the template either using dedicated frames [62,128] or along the tracking process [53,110]. Our approaches [132,134] fall into the second category and are developed concurrently with the alternatives [53,62,110,128]. Our approach requires no parametric shape models needed in [62], no training database for discriminative pose estimation as in [128] and allows larger flexibility for shape adaptation compared to [53,110]. A shared limitation of the second class of methods compared to the performance capture settings [38, 121] is the lack of rich geometric details in the adapted template. Normally detailed personalized shape model are constructed with another line of techniques mostly developed in Computer Graphics and Geometry Processing.

2.2 Shape Reconstruction

The topic of shape reconstruction has also been extensively studied, however mainly for rigid bodies [26, 78]. Some methods can deal with small degree of motions for articulated objects [34, 129, 136]. Only a few methods allow the subject to perform free movements. Using multi-view setup, detailed geometry can be recovered [38,76,110,121]. High quality monocular scans of strictly articulated objects under different poses can also be accurately merged to obtain a complete shape [28]. In the following, closely related methods are classified into model-based and model-free approaches and briefly discussed.

2.2.1 Model-based Approaches

Statistical shape models have been widely used as the prior information for shape reconstruction. In particular, SCAPE model [11] has been successfully used for human body modeling both from color images [57] and from range maps [62, 129]. Several extensions of the SCAPE model have also been proposed [29, 65]. Weiss et al. [129] captures depth maps of a subject from four different views using a Microsoft Kinect and adapts the SCAPE model according to both depth and silhouette information. The subject is required to remain almost static or perform known poses across four scans. A common limitation of parametric model is the limited capability of extrapolation. For example, the SCAPE model does not capture the geometric information of the subject's apparel and therefore is not appropriate for general purpose scanning. To overcome the limitation, Tong and colleagues [117] adopts the statistical shape model developed in [58] to only estimate the shape of the subject and build a common deformation graph. The model is then built through pair-wise registration using the deformation graph based on Embedded Deformation Model [113], followed by global alignment with loop closure. They use three Kinect sensors to scan the subject holding still on a turntable and rely on color feature tracking to localize control points for pair-wise registration.

Non-parametric models generally come in the form of a pre-scanned personalized model or a generic mesh model. When a personalized model is used, the purpose is normally to capture the deformation details from high quality input, such as the performance capture [38,75,121], rather than reconstructing a 3D model. Straka et al [110] adapts a generic human template model to the visual hull of multi-view images where the subject's body poses are known. Our detailed shape reconstruction approach (Chapter 7) also uses a generic non-parametric template model, however requires only a single depth sensor.

2.2.2 Model-free Approaches

Extensive research have been conducted for general 3D reconstruction and various techniques have been developed, such as stereo, structure from model, structured light scanning and shape from X [114]. Some recent studies use same input as ours, namely range scans, to reconstruct complete 3D models of static scenes as well as deformable objects.

For static scenes, KinectFusion [91] can be used for effective real-time scanning using a single Kinect sensor. The technique developed by Cui et al. [35] can operate on low resolution Time-of-Flight sensor. However, neither method could easily generalize to scenes even with small movements, which is normally the case when scanning a live subject. To solve this issue, several authors [78,117,136] propose to perform non-rigid alignment based on the Embedded Deformation Model [113] to accommodate potential small movements. Zeng et al. [136] place two sensors facing each other to capture both the frontal and the back of the subject and incrementally integrate the scanned depth maps with implicit loop closure. Li and colleagues [78] take advantage of the built-in motor of Microsoft Kinect for close range yet full body scanning while the subject orientCui et al. [34] relax the assumption to limited articulated motions. Chang and colleagues [28] develop a method that works with full articulated motions. They rely on SpinImages [68] for pair-wise registration [27] and then perform a global alignment to merge all meshes. Although their results are promising on high quality (partial) scans, this method is sensitive to the noises from commodity depth sensors as reported in [34].

Chapter 3 Preliminaries: Shape and Pose Representations

Throughout our work, we represent the geometry of a 3D shape as a 3D mesh $\mathcal{M} = \{\mathcal{V}, \mathcal{F}\}$. It consists of the surface vertices $\mathcal{V} = \{\boldsymbol{v}_m | m = 1, 2, \cdots, M\}$ and the surface topology (connectivities) $\mathcal{F} = \{(i, j, k) \,|\text{for some } 1 \leq i, j, k \leq M\}$ with each element being a triplet (i, j, k) that defines a triangular patch in our case. How the geometry evolves as the object performs certain movements is dependent on the deformation models adopted to approximate the nature of the object. If an object is completely rigid, it moves according to some rigid body transformation as one piece. In our settings, the targets are the quasi-articualted objects which undergo articulated motions globally and small non-rigid motions locally. In the remainder of this section, we describe the models we use in this dissertations to model quasi-articulated objects.

3.1 Articulated Deformation Model

The assumption of articulated structure implies certain hierarchy across different components of the object. An articulated motion is normally described with a kinematic tree and computed via forward kinematics.

Figure. 3.1 illustrates a simple scenario of two segments. In this case, there are two joints, one root joint with six degrees of freedoms and another with one degree of freedom (called revolute joints [87]). A local coordinate system can be defined on each of the joints. Without loss of generality, we assume the local coordinate system of the root joint coincides with the global (world) coordinate system (an additional rigid transformation

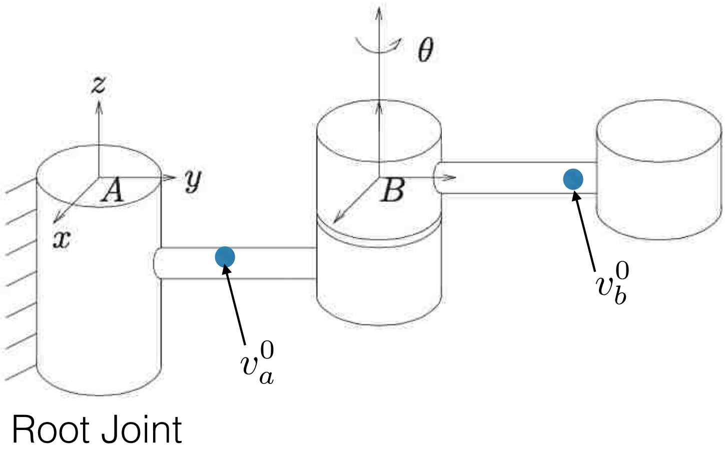


Figure 3.1: Illustration of articulated motion of an articulated object with two segments. Image adapted from [87]. See the text for explanation.

can be applied at the end if this does not hold). Assume the root undergoes a rigid body transformation G_g and the other joint rotates around the local axis for a certain degree that results in a transformation matrix G_b. The transformation G_b actually involves three parts, namely transformation from the world coordinate system to the local coordinate system, the rotation matrix around local coordinates system and then transformation from the local coordinate system to the world coordinate system. With these motions, the surface vertex $\boldsymbol{v}_a$ on segment A and $\boldsymbol{v}_b$ on segment B will move in the following way:

$$\boldsymbol{v}_a = G_g \cdot \boldsymbol{v}_a^0 \tag{3.1}$$

$$\boldsymbol{v}_b = G_g \cdot G_b \cdot \boldsymbol{v}_b^0 \tag{3.2}$$

where $\boldsymbol{v}_a^0$ and $\boldsymbol{v}_b^0$ are the positions in some default configuration. If the object contains more than two segments, the hierarchy is normally in a tree shape. Similar to Equation 3.2, the transformations of a node is the product of its own transformation with all its parents and

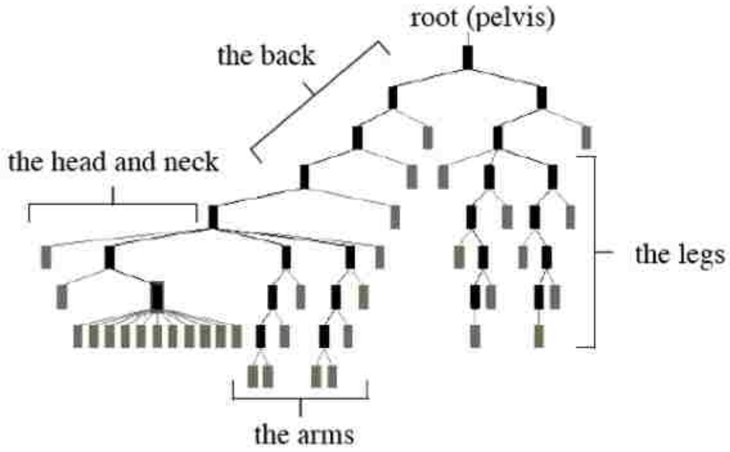


Figure 3.2: A typical hierarchical representation of human body [14].

can be represented as:

$$T_i = T_g \prod_{k=1}^{K} (G_k)^{\delta_{ki}} \tag{3.3}$$

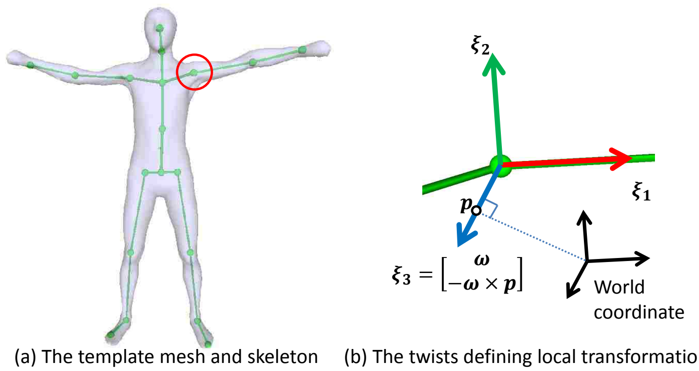
$$\text{where} \quad \delta_{ki} = \begin{cases} 1 & \text{joint } k \text{ is an ancestor of joint } i \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

where T_g is the global transformation, i.e. transformation of the root, G_k is the local transformation and T_i is the accumulated transformation.

For human body, a typical hierarchical representation is illustrated in Figure 3.2. The forward kinematic follows Equation 3.3. Therefore a surface vertex deforms according to the following equation:

$$\boldsymbol{v}_m = T_{b(m)} \cdot \boldsymbol{v}_m^0 \qquad b(m) \in [1, K] \tag{3.5}$$

where $\boldsymbol{v}_m^0$ is the vertex position in some reference pose and K is the number of bones. The function $b(\cdot)$ maps a vertex index to the index of the bone directly controlling this vertex. Equation 3.5 represents the shape deformation with a pose under the articulated deforma-



ξ_2
ξ_1
p
$\xi_3 = \begin{bmatrix} \boldsymbol{\omega} \\ -\boldsymbol{\omega} \times \boldsymbol{p} \end{bmatrix}$
World coordinate

(a) The template mesh and skeleton

(b) The twists defining local transformations for the joint circled in (a)

Figure 3.3: Our template model consisting of the mesh and the underlying skeleton (kinematic structure). A rigid body transformation is represented via a twist $\boldsymbol{\xi}$. The vector $\boldsymbol{\omega}$ is the orientation of the rotation axis corresponding to $\boldsymbol{\xi}$.

tion model. The underlying kinematic tree essentially describes the skeleton structure. The template we use throughout this dissertation is shown in Figure 3.3.

3.2 Linear Blend Skinning

Due to the local non-articulated deformation of human body, the Linear Blend Skinning (LBS) deformation model [93] is widely used to better encode this behavior. This model is used throughout this dissertation due to its simplicity and effectiveness. With LBS, a surface vertex is influenced not only by a single bone, but potentially multiple bones. The degree of influences are encoded in the so-called skinning weights, which we denote as $\mathcal{A} = \left\{\alpha_{m,k} \middle| \sum_{k=1}^{K} \alpha_{m,k} = 1; m = 1, \cdots, M; k = 1, \cdots, K\right\}$. A vertex is then transformed by a linear combination of the transformations of all the bones weighted by the skinning weights.

$$\boldsymbol{v}_m = \Big(\sum_{i=1}^{K} \alpha_{m,i} T_i\Big) \boldsymbol{v}_m^0 \tag{3.6}$$

Substitute T_i in the above equation with Equation 3.3, we get

$$\boldsymbol{v}_m = \left(T_g \sum_{i=1}^{K} \alpha_{m,i} \prod_{k=1}^{K} (G_k)^{\delta_{ki}}\right) \boldsymbol{v}_m^0 \tag{3.7}$$

3.3 Twist-Based Representation

A rigid body transformation T can be represented in various ways, e.g. through euler angles, quaternions, etc. In this dissertation, we represent a 3D transformation T via an exponential map, similar to [23, 51]:

$$G = e^{\hat{\xi}\theta} = \sum_{k=0}^{\infty} \frac{\hat{\xi}^k}{k!} \tag{3.8}$$

Here $\hat{\xi}$ is the 4×4 matrix form of the twist $\boldsymbol{\xi}$ (a six dimensional vector representing the location and orientation of the rotation axis), while θ is the angle of rotation around $\boldsymbol{\xi}$. An example is shown in Figure 3.3. The advantage of this representation is its simplicity in linearizing the transformation with respect to the rotation angle θ, which leads to linear optimization for pose tracking. Details about this representation can be found in the excellent book by Murray et al. [87] or in [23].

Given the template model (Figure 3.3) in a reference pose, the set of rotation axes are pre-defined, represented as the set of twists $\Xi = \{\boldsymbol{\xi}_k | k = 1, \cdots, K\}$. Therefore, a pose ($\Theta$) of the model is defined by the set of joint angles $\{\theta_k | k = 1, \cdots, K\}$ corresponding to each of the twists, as well as the global transformation $\boldsymbol{\xi}_g$. In practice, one joint might have more than one degree of freedoms, therefore is related to more than one joint twists and angles. For simplicity, we can treat each $< \boldsymbol{\xi}_i, \theta_i >$ as a joint. Without lost of generality, we assume the indexes of parent nodes in the kinematic tree are smaller than those of their children.

With this representation, we can represent the global transformation T_g and the transformations of all bones $\{T_i\}$ as:

$$T_g = e^{\hat{\xi}_g} \tag{3.9}$$

$$G_k = e^{\hat{\xi}_k \theta_k} \tag{3.10}$$

Consequently, the transformation in Equation 3.3 can be written as

$$T_i = e^{\hat{\xi}_g} \prod_{k=1}^{K} e^{\delta_{ki} \hat{\xi}_k \theta_k} \tag{3.11}$$

And the deformation according to LBS defined in Equation 3.7 can be expressed as:

$$\boldsymbol{v}_m(\Theta) = e^{\hat{\xi}_g} \sum_{i=1}^{K} \Big(\alpha_{m,i} \prod_{k=1}^{K} e^{\delta_{ki} \hat{\xi}_k \theta_k} \Big) \boldsymbol{v}_m^0 \tag{3.12}$$

Where $\Theta = [\boldsymbol{\xi}_g, \theta_1, \cdots, \theta_K]^T$ defines the global and local motion and is the body pose which is then the target for pose estimation. Note that in this dissertation, for notation simplicity, the vector $\boldsymbol{v}$ could represent a homogeneous or an inhomogeneous coordinate, whichever is appropriate depending on the context.

3.4 Incremental Pose Update

For continuous movements, we can assume the pose change is small and continuous between two time instances. With this assumption, we can derive how the surface vertices evolve with respect to previous states and the incremental pose update. At this point, we will take advantage of the linearization of exponential map as mentioned earlier.

Denote the pose of time t as $\Theta^t = [\boldsymbol{\xi}_g^t, \theta_1^t, \cdots, \theta_K^t]^T$ and the corresponding surface vertices as $\{\boldsymbol{v}_m(\Theta^t) | m = 1, \cdots, M\}$. The small pose changes between time t and $t+1$, denoted as

$\Delta\Theta = [\Delta\boldsymbol{\xi}_g^t, \Delta\theta_1, \cdots, \Delta\theta_K]^T$, can be defined as the follows:

$$e^{\hat{\xi}_g^{t+1}} = e^{\Delta\hat{\xi}_g^t} e^{\hat{\xi}_g^t} \tag{3.13}$$

$$\theta^{t+1} = \theta^t + \Delta\theta^t. \tag{3.14}$$

We can then use the first-order approximation to linearize the infinite sum in Equation 3.8 to obtain:

$$e^{\Delta\hat{\xi}_g^t} \approx (I + \Delta\hat{\xi}_g^t) \tag{3.15}$$

$$e^{\hat{\xi}_k \theta_k^{t+1}} \approx e^{\hat{\xi}_k \theta_k^t}(I + \Delta\theta_k^t \hat{\xi}_k) \tag{3.16}$$

where I is the 4×4 identity matrix. Substitute these two equations into Equation 3.11, the transformations of a bone i at time $t+1$ can be expressed as.

$$T_i^{t+1} \approx (I + \Delta\hat{\xi}_g^t) e^{\hat{\xi}_g^t} \prod_{k=1}^{K} e^{\delta_{ki}\hat{\xi}_k \theta_k^t}(I + \delta_{ki}\Delta\theta_k^t \hat{\xi}_k) \tag{3.17}$$

Expand this product and ignore terms with product of at least two angular changes, i.e. $\Delta\theta_j^t \Delta\theta_k^t$, we then get

$$T_i^{t+1} \approx (I + \Delta\hat{\xi}_g^t) e^{\hat{\xi}_g^t} \left[\prod_{k=1}^{K} e^{\delta_{ki}\hat{\xi}_k \theta_k^t} + \sum_{k=1}^{K} \delta_{ki} \Big(\prod_{j=1}^{k} e^{\delta_{jk}\hat{\xi}_j \theta_j^t} \hat{\xi}_k \Delta\theta_k^t \prod_{j=k+1}^{K} e^{\delta_{ji}\hat{\xi}_j \theta_j^t} \Big) \right] \tag{3.18}$$

With our assumption that the index of parent joints is smaller than their children for notation, we can define two transformation matrix M_b^t and T_b^t as

$$M_i^t = \prod_{k=1}^{K} e^{\delta_{ki}\hat{\xi}_k \theta_k^t} = \prod_{k=1}^{i} e^{\hat{\xi}_k \theta_k^t} \tag{3.19}$$

$$T_i^t = e^{\delta_{ki}\hat{\xi}_g^t} M_i^t \tag{3.20}$$

Therefore Equation 3.18 becomes

$$\begin{aligned}
T_i^{t+1} &\approx (I + \Delta\hat{\xi}_g^t)e^{\hat{\xi}_g^t}\Big[M_i^t + \sum_{k=1}^{K}\delta_{ki}M_k^t\hat{\xi}_k(M_k^t)^{-1}M_i^t\Delta\theta_k^t\Big] \\
&= (I + \Delta\hat{\xi}_g^t)\Big[T_i^t + \sum_{k=1}^{K}\Big(\delta_{ki}T_k^t\hat{\xi}_k(M_k^t)^{-1}(e^{\hat{\xi}_g^t})^{-1}e^{\hat{\xi}_g^t}M_i^t\Delta\theta_k^t\Big)\Big] \\
&= (I + \Delta\hat{\xi}_g^t)\Big[T_i^t + \sum_{k=1}^{K}\Big(\delta_{ki}T_k^t\hat{\xi}_k(T_k^t)^{-1}T_i^t\Delta\theta_k^t\Big)\Big] \qquad (3.21)
\end{aligned}$$

Substitute the above equation into Equation 3.6, we can derive the relationship between surface vertices update and the pose update under LBS as follows.

$$\boldsymbol{v}_m(\Theta^{t+1}) = \sum_{i=1}^{K}\alpha_{m,i}(I + \Delta\hat{\xi}_g^t)\Big[T_i^t\boldsymbol{v}_m^0 + \sum_{k=1}^{K}\Big(\delta_{ki}T_k^t\hat{\xi}_k(T_k^t)^{-1}T_i^t\boldsymbol{v}_m^0\Delta\theta_k^t\Big)\Big] \qquad (3.22)$$

Define

$$\hat{\xi'}_k^t = T_k^t\hat{\xi}_k(T_k^t)^{-1} \qquad (3.23)$$

Further expand Equation 3.22 and ignore the higher order term, we get

$$\begin{aligned}
\boldsymbol{v}_m(\Theta^{t+1}) &\approx \boldsymbol{v}_m(\Theta^t) + \Delta\hat{\xi}_g^t\boldsymbol{v}_m(\Theta^t) + \sum_{i=1}^{K}\alpha_{mi}\sum_{k=1}^{K}\delta_{ki}\hat{\xi'}_k^t\boldsymbol{v}_m(\Theta^t)\Delta\theta_k^t \\
&= \boldsymbol{v}_m(\Theta^t) + I_m^t\boldsymbol{\Delta\xi}_g^t + \sum_{i=1}^{K}\beta_{k,i}\hat{\xi'}_k^t\boldsymbol{v}_m(\Theta^t)\Delta\theta_k^t \qquad (3.24)
\end{aligned}$$

where

$$I_m^t = \Big[I_{3\times3} \quad [\boldsymbol{v}_m(\Theta^t)]_\times\Big] \qquad (3.25)$$

$$\beta_{m,i} = \sum_{k=1}^{K}\delta_{ik}\alpha_{m,k} \qquad (3.26)$$

and the operator $[\cdot]_\times$ converts a vector to a skew-symmetric matrix. Here the weight $\beta_{i,k} \in [0,1]$ reflects the influence of the joint k on the vertex i, by accumulating skinning weights from all the children of joint k in the kinematic tree.

Equation 3.24 defines the relationship between incremental pose update and surface update and will be used later for pose estimation.

Chapter 4 Data Driven Human Pose Estimation

In this chapter, we present a data-driven approach for accurate human pose estimation from depth images. In this work, we use the SwissRanger ToF sensor [3]. Our algorithm use a pre-captured motion database to constrain the possible body configuration space. The motion database contains both body surface models and corresponding skeleton body configurations. When an input depth map is matched with a body surface model, it obtains not only the semantic labeling from the surface model, but also the underlying body configuration. We then further optimize the body configuration using a non-rigid point registration process. It serves two purposes: first to account for the difference between the sample surface model; and second to fill in the missing regions that are occluded in the input. In this two-stage process, we avoid the typical problem of a huge motion database associated with directly mapping input data to body configuration, while we also provide good initial estimation so that optimization-based refinement is unlikely to be trapped in local minima.

An important technical contribution of this work is a new *view-independent* matching algorithm between a 3D full-body surface mesh and a depth map. A new challenge in any single-view setup is that the input is *view-dependent* and incomplete. Matching the depth map directly with a complete surface mesh, *with non-rigid deformation*, can lead to inaccurate or even wrong initial body configuration estimation and eventually reduce the final accuracy. Toward this end, we apply PCA to both the input depth map and the motion database; first aligning them in the three principal axes, then searching in a reduced space to both accelerate the computation and remove ambiguity caused by small variations in postures. Our method can effectively handle body-size variations across subjects, benefiting from the matching approach and the non-rigid point registration. In addition, we extend a point denoising scheme to significantly reduce the noise and outliers in the input depth map – a problem that is detrimental in practice.

4.1 Overview

In our method, a motion database is utilized, which is generated by driving a *generic* human mesh model with movements captured from an eight-camera optical motion capture system [8] operating at 120Hz. Around 19300 poses are recorded, including walking, running, bending, etc. The human model includes both a surface mesh and an embedded skeleton that contains 19 joints, as shown in Figure 3.3. The mesh model is animated with linear blending technique according to the recorded motions [93]. We denote a deformed mesh under a certain pose as $\mathcal{M}_l$. Four *synthesized* depth images, denoted as $\{\mathcal{P}_l^i\}_{i=1}^4$, are also rendered from four different perspectives. These depth images will be used for view-independent shape matching explained in Section 4.3.1.

The input to our approach is one or more depth images $\{\mathcal{X}_j\}_{j=1}^N$, or equivalently point clouds, from a single depth sensor (we will use these two terms interchangeably through this chapter). Our goal is to estimate a configuration $\hat{\Upsilon}_j$ given a depth image $\mathcal{X}_j$ based on our motion database $\{\mathcal{M}_l, \mathcal{P}_l^1, \mathcal{P}_l^2, \mathcal{P}_l^3, \mathcal{P}_l^4, \Upsilon_l\}$.

Figure 4.1 shows the outline of our processing pipeline. Given a point cloud, we first remove irrelevant objects based on distance information, for which we use two fixed distance thresholds representing the interested distance range throughout our test. A modified surface reconstruction algorithm is applied to remove noise. Then the cleaned point cloud

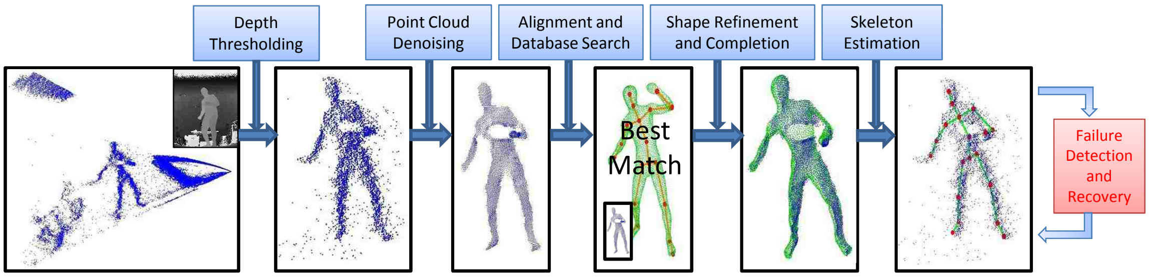

Depth Thresholding
Point Cloud Denoising
Alignment and Database Search
Shape Refinement and Completion
Skeleton Estimation
Best Match
Failure Detection and Recovery

Figure 4.1: The outline of our processing pipeline. The leftmost image is a typical depth map, we define distance thresholds to cut the subject out. It goes through a number of processing stages, generating the estimated skeleton embedded in the input point cloud.

is transformed into a canonical coordinate frame in order to remove viewpoint dependency, and a similar pose is identified in our motion database. Then a refined pose configuration is estimated through non-rigid registration between the input and the rendered depth map for the corresponding pose. We rely on database exemplars and a shape completion method to deal with large occlusions, i.e., missing body parts. Finally a failure detection and recovery mechanism is adopted to handle occasional failures from previous steps, using the temporal information.

4.2 Point Cloud Segmentation and Denoising

The input depth map first needs to be processed to remove background and noise. Given the depth information, background objects can be easily removed by defining a bounding box or through background subtraction. However, the noise level in the depth map from typical ToF sensor is significant as shown in Figure 4.2(a). This is most likely due to the long range (for full body capture) between the subject and the camera. Since our subsequent processing requires finding point correspondences between the input point cloud and database exemplar, we need to overcome this obstacle of noisy input. Here we modify a

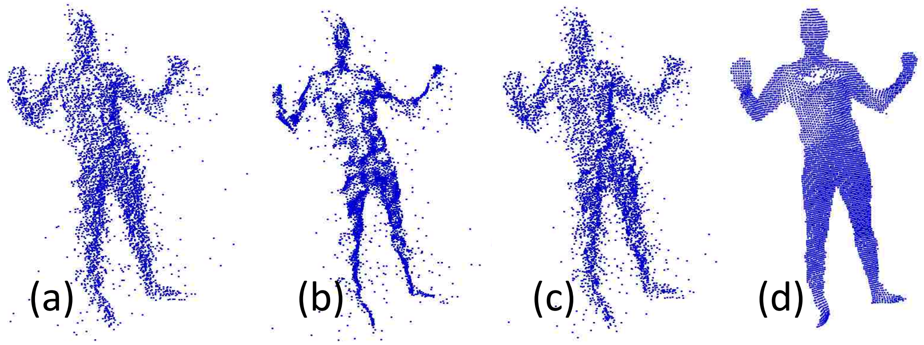


Figure 4.2: Comparison of original LOP and our modified LOP applied on a point cloud from depth sensor. from left to right (a) the input; (b) LOP with support radius of 0.2m; (c) LOP with support radius of 0.1m; (d) with our modified LOP.

surface reconstruction algorithm–*Locally Optimal Projection* (LOP) [80] for denoising.

LOP is a parameterization-free operator that, given a target point-set $\mathcal{P} = \{p_i\}_{i \in I}$, projects an arbitrary point-set $\mathcal{X} = \{x_j\}_{j \in J}$ onto the data $\mathcal{P}$, to reconstruct the underlying geometry structure in the data $\mathcal{P}$. The criteria is to minimize the sum of weighted distances between the projected point-set and $\mathcal{P}$, meanwhile preventing points in the projected point-set being too close to each other. Applying LOP directly to our point cloud is problematic though. The amount of smoothness is controlled by a radius value h, which determines how big a neighborhood a projected point can contribute to the objective distance function. As shown in Figure 4.2, if h is too large, it leads to obvious shrinkage of the point cloud; if h is too small, there is little effect of denoising.

We thereby seek a solution that offers both smoothness and the preservation of geometric structure. With a closer look into applying this method to the depth map, we realize that shrinkage could be avoided by projecting only z coordinates of the point-set, which contains the most important depth information; while x and y can then be calculated through re-projection.

Therefore, given a point-set $\mathcal{P} = \{p_i\}_{i \in I}$, where $p_i = [x_i, y_i, z_i] \in R^3$ are the 3D coordinates, following the original derivation in [80], our modified LOP algorithm is initialized as follows:

$$z_i^{(1)} = \frac{\sum_{s \in I} z_s * \theta(||p_s - p_i||)}{\sum_{s \in I} \theta(||p_s - p_i||)} \tag{4.1}$$

$$x_i^{(1)} = z_i^{(1)} * x_i / z_i \tag{4.2}$$

$$y_i^{(1)} = z_i^{(1)} * y_i / z_i \tag{4.3}$$

where $\theta(\cdot)$ is a fast decreasing function controlled by h, $\theta(r) = e^{-16r^2/h^2}$. Then for each iteration $k = 1, 2, \ldots K$, the point is updated as

$$z_i^{(k+1)} = \frac{\sum_{s \in I} \alpha_s^i z_s}{\sum_{s \in I} \alpha_s^i} + \mu \frac{\sum_{s \in I \setminus \{i\}} ||p_i^{(k)} - p_s^{(k)}|| \beta_s^i}{\sum_{s \in I \setminus \{i\}} \beta_s^i} \tag{4.4}$$

$$x_i^{(k+1)} = z_i^{(k+1)} * x_i / z_i \tag{4.5}$$

$$y_i^{(k+1)} = z_i^{(k+1)} * y_i / z_i \tag{4.6}$$

where

$$\alpha_s^i = \frac{\theta(||p_s^{(k)} - p_i||)}{||p_s^{(k)} - p_i||} \tag{4.7}$$

$$\beta_s^i = \frac{\theta(||p_s^{(k)} - p_i^{(k)}||)}{||p_s^{(k)} - p_i^{(k)}||} \left| \frac{\partial \eta}{\partial r} (||p_s^{(k)} - p_i^{(k)}||) \right| \tag{4.8}$$

Here $\eta(r) = 1/3r^3$ and $\mu \in [0, 1/2)$ is a repulsion parameter that leverages between smoothness and surface geometry accuracy. In practice we found that setting $h = 0.5$ and $\mu = 0.35$ usually obtains the best results within $K = 5$ iterations. After applying LOP, we also remove isolated outliers since these points do not have effective supporting neighborhood and remain unchanged after projection. These points are identified by thresholding distance to their nearest point. The effectiveness of our denoising scheme is shown in Figure 4.3, in

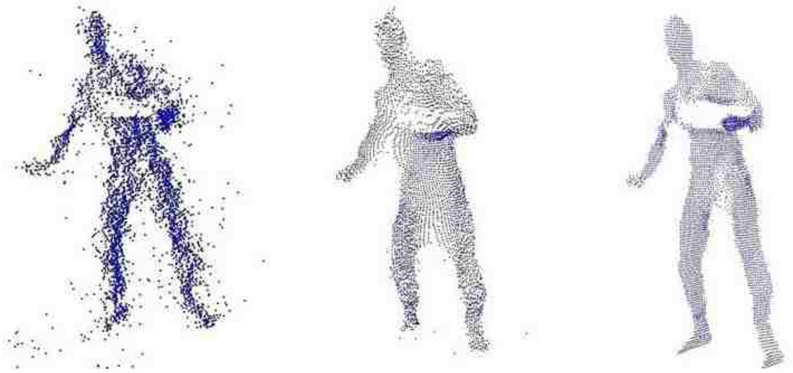


Figure 4.3: Comparison of point cloud smoothing using bilateral filtering and our modified LOP algorithm. From left to right: input; after Bilateral Filtering; after our modified LOP.

which we also compare it with bilateral filtering (BF) on the depth map. BF generates undesirable points connecting disjoint parts, probably due to the low-resolution of the depth map and too many stray points on occlusion boundaries.

4.3 Model Based Motion Estimation

After the depth map has been segmented and cleaned, our next step is to search for a similar pose in the motion database. Directly measuring similarity between the complete mesh model and a depth map is difficult, since a depth map is incomplete (at least 50% of a subject's information is missing) and there is no prior knowledge about from what viewpoint a depth map is captured.

Our solution to this search problem involves two steps. First we generate several synthesized depth maps from representative viewing directions, and align the input point cloud to these representative views, in this way removing the **viewpoint dependency** of the input point cloud. Then we apply dimension reduction techniques to find the most similar depth map (and body configuration) efficiently.

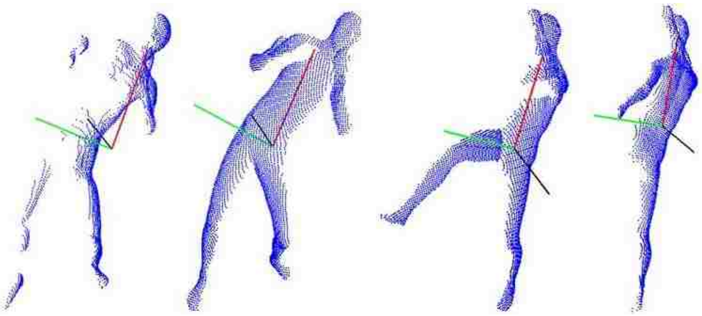


Figure 4.4: Visualization of the three principal axes (color-coded) of different point clouds. They are quite stable across different viewpoints. Two different poses are shown. (The right-most one is rendered from back-view with the entire right leg occluded.)

4.3.1 Point Cloud Alignment

We address this viewpoint dependency problem with the observation that principal axes of a point cloud provide robust characteristics of a pose, through which a transformation can be constructed and applied on the point cloud to rectify it to a canonical view. More specifically, given a point cloud $\mathcal{X}$ of dimension $N \times 3$, the principal axes are the eigenvectors of the 3×3 covariance matrix that represents the three major directions the point cloud spans. As we can see in Figure 4.4, they generally provide sufficient match across different view points for our purpose; while our neighbor search and non-rigid registration approach described in the following sections can deal with remaining small misalignment. Therefore we can define a local coordinate frame based on the mean value of the point cloud and the three principle axes. Note that we do not know the positive direction of the principal axes, therefore we pick the largest principle axis and define its positive ("up") direction as the up-direction of the camera coordinate–assuming the camera is usually not up-side-down. Using the point cloud's mean value as the origin, we define four canonical coordinate frames by alternating the positive directions of the remaining two axes. For each canonical coordinate frame, we define a virtual depth camera that is away from the origin and looking into the depth direction. Each mesh model $\mathcal{M}_l$ is transformed into its own canonical coordinate frames and four synthesize depth maps are rendered, denoted as $\{P_l^i\}_{i=1}^4$.

For an input point cloud $\mathcal{X}$, we also compute such a transformation T but with the difference that we just pick a random positive direction for the remaining two axes. The transformed point cloud $\mathcal{X}^c = T(\mathcal{X})$ is in a view-independent canonical coordinate frame.

4.3.2 Nearest-Neighbor Search in Low-Dimensional Subspace

In $\mathcal{X}^c$ the viewpoint dependency is mostly removed. We can now search the synthesized depth maps to find the most similar pose. One could search in 3D space by comparing the point distances in 3D,we instead search in the PCA space of the image space and look for a P_l^i that is closest to $\mathcal{X}^c$ in the PCA space. More specifically, all synthesized depth maps $\{\mathcal{P}_l^i\}$ are vectorized and stacked into a matrix from which a PCA subspace and corresponding coefficient vectors $\{\lambda_l^i\}$ are learned.

$\mathcal{X}^c$ is re-synthesized as a standard depth image in the canonical view and vectorized to calculate PCA coefficients in this subspace. Finally, a most similar pose is identified by finding P_l^i closest to $\mathcal{X}^c$ in the PCA space, i.e. finding $< l, i >$ that satisfies

$$< l, i >= \underset{1 \le l \le N_c, 1 \le i \le 4}{\arg\min} \{||\lambda_l^i - \gamma||\} \tag{4.9}$$

Up to now we have obtained a point cloud $\mathcal{P}_l^i$ and its corresponding surface model $\mathcal{M}_l$ that are in similar pose as $\mathcal{X}^c$. Since $\mathcal{M}_l$ has a known joint configuration, an initial estimation of the joint configuration of $\mathcal{X}^c$ is obtained. Note everything is now defined

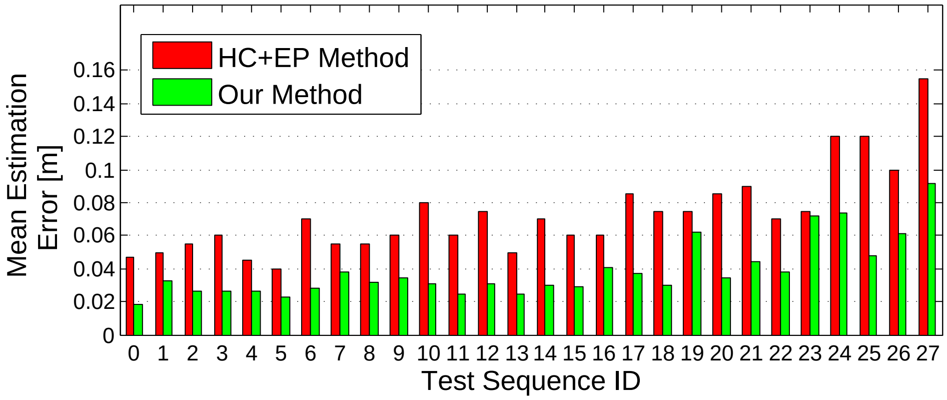


Figure 4.5: The mean estimation error of our approach on Stanford's public test dataset that consists of 28 sequences of motions,compared with the results reported in [52] using Hill Climbing combined with Evidence Propagation (HC+EP)

in a canonical coordinate frame, we need to apply the inverse transform T^{-1} to $\mathcal{M}_l$ and joint configurations so that they are in the input coordinate frame. Such transformation facilitates the refinement process discussed in the next section. For the sake of simplicity in notation, we assume $\mathcal{M}_l, \mathcal{P}_l^i$ are defined in the input coordinate frame from this point on.

4.4 Experimental Results

Our system is implemented in Matlab and evaluated mainly based on the public dataset provided by Stanford [52]. This dataset consists of 28 sequences of motions, of which about half contain 100 frames and the others have 400 frames, all recorded at 25fps with a depth camera of resolution 176 × 144. The motions range from simple movements such as lifting a hand to a very challenging tennis swing with severe occlusion and simultaneous movements of several body parts. Locations of 3D markers attached to the subject's body measured using a commercial active marker motion capture system are provided as ground truth. In order to compare with this ground truth, we choose a set of patches on our mesh model to approximate the position of markers according to their markers setup, as was done in [52]. Then estimation error is measured as

$$\bar{e} = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{1}{N_m}\sum_{i=1}^{N_m}||m_i - \hat{m}_i|| \tag{4.10}$$

where N_f and N_m are number of frames and markers respectively. m_i is measured ground truth of marker location and $\hat{m}_i$ is our estimation. Throughout our evaluation we use *the single motion database* described in the beginning of Section 4.1. Figure 4.5 shows our estimation error for all 28 test sequences, compared with the best result reported in [52], which combines Hill Climbing(HC) search and Evidence Propagation(EP). As we can see, our method achieves substantially higher accuracy, with a total mean error around 38*mm*, compared to their 100*mm*. It should be emphasized that we use a generic human model to generate our own database, no sequence from the test data is in our motion database; and we always use the entire database (e.g., the nearest neighbor search is global, instead of sequence specific). As a tradeoff for the high accuracy, the computational time is currently non-real time, mainly due to the non-rigid registration process. With our implementation in Matlab, the running time for each frame is between 60s and 150s, depending on the number of points and pose differences.

In the remainder of this section, we will show the effectiveness of various components of our pipeline. We use two representative sequences from the Stanford dataset for demonstration: sequence 21 of moderate complexity that includes mainly hand and feet movement and sequence 27 which is the most challenging tennis swing motion.

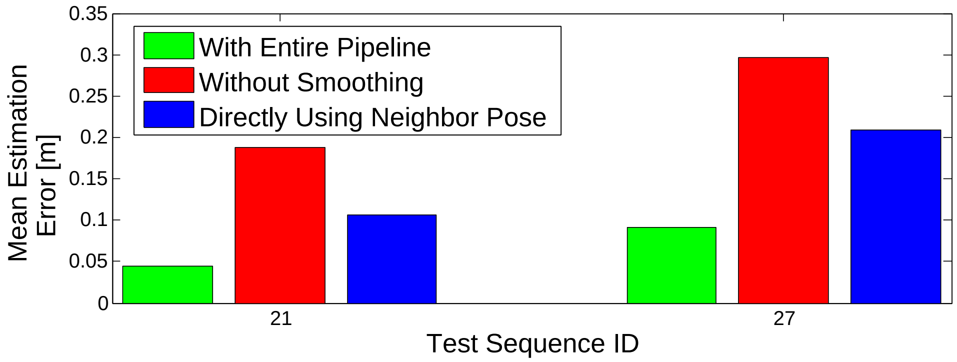


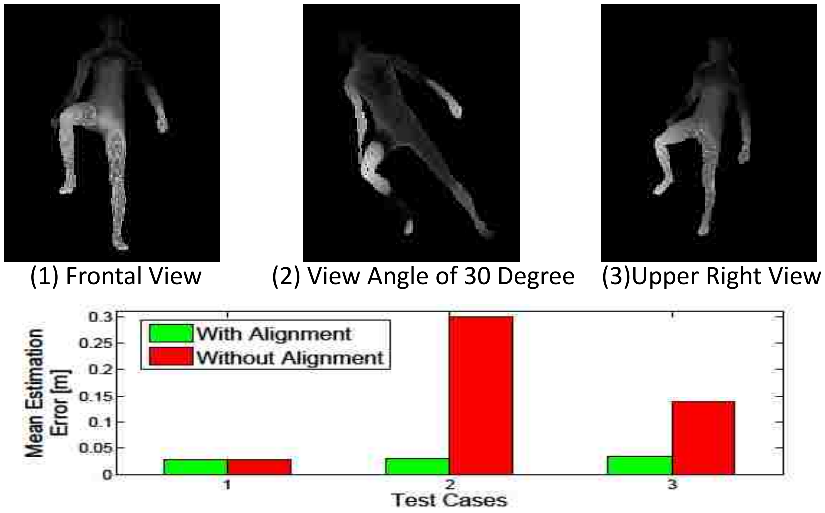
Figure 4.6: Comparison of estimation errors of three methods: using our entire pipeline, without smoothing and directly using neighbor pose without refinement. The results demonstrate the necessity of such processing components.

4.4.1 The Effectiveness of Denoising

The point cloud denoising procedure is important for our method to correctly estimate poses, due to the way it is designed. Simply applying this approach to an original point cloud with background objects removed gives unsatisfactory results as shown in Figure 4.6. The importance of our smoothing module arises from two reasons: in the presence of severe noise, neighbor search is erroneous and CPD is strongly perturbed.

4.4.2 The Effectiveness of Pose Refinement

Here we show both the effectiveness of our neighbor search approach and the necessity of pose refinement. In Figure 4.6 we see that directly representing the pose of an input point cloud by that of the identified neighbor sample results in higher estimation error. On the other hand, the error for sequence 21 is still acceptable, meaning that similar poses are in general correctly localized. In the meantime, the reason we have larger errors for sequence 27 is that the pose of the neighbor exemplar cannot properly approximate the input, since our database might not contain such substantially similar poses. In general, pose refinement



(1) Frontal View (2) View Angle of 30 Degree (3)Upper Right View

Figure 4.7: Viewpoint independency test. First row: examples of synthetic inputs (see text for explanation). Second row shows quantitative evaluations.

is required for accurate estimation.

4.4.3 Viewpoint Independency

The test sequences in Stanford's dataset were basically captured from the frontal view, and re-rendering from those partial point clouds with different viewpoints would result in even more incomplete data. To verify the issue of view independency, we test on three synthetic sequences rendered with our mesh model walking. Shown in the first row of Figure 4.7, the first sequence is captured from then normal frontal view; the second one with a roll angle of 30 degrees; and the last case with the camera looking from the upper right direction, tilting down. The quantitative results are shown in the second row of Figure 4.7. It can been seen easily what the detrimental effect of view-dependent input can be, if not handled.

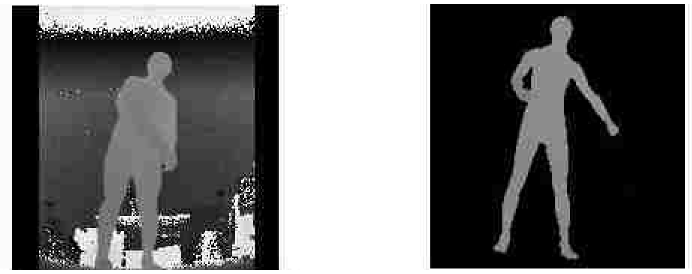


Figure 4.8: An example of our failure case (left), for which the most similar pose in our database (right) exhibits a very large difference in pose.

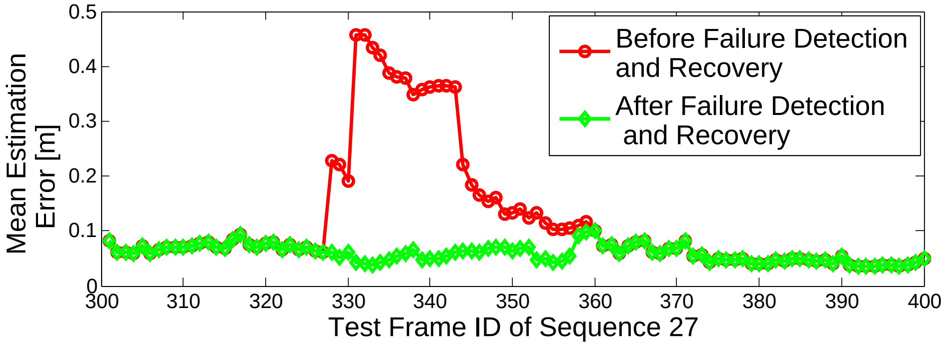


Figure 4.9: Estimation errors, with and without failure detection and recovery, of the test frames where failure cases happen and our failure detection and recovery method takes effects.

4.4.4 Failure Detection

In this experiment we use the input from frame 301 to frame 400 of sequence 27 as an example. Among these test frames, for around 35 frames the subject remains in a relative static pose for which our database does not have a similar one. The difference is shown in Figure 4.8. Under this situation, our failure detection and recovery mechanism takes effect and re-estimates poses for input through temporal information. The comparison of results with and without such detection and recovery is shown in Figure 4.9. As we can see, failure poses were effectively recovered. However, notice that for a majority of the test sequence in this dataset, our regular pipeline can generate good results.

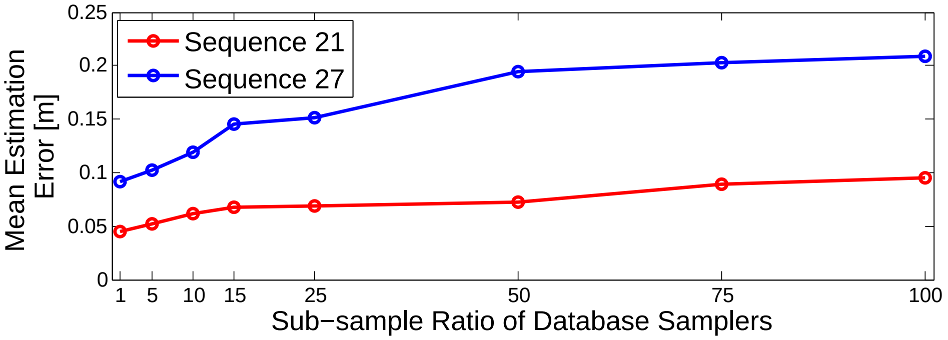


Figure 4.10: Estimation errors of our approach with a set of motion samples which is sub-sampled from our original database, with different ratios.

4.4.5 Database Dependency

In this test, we aim at quantitatively determining the relationship between our estimation result and the number of database samples. Originally our database contains around 19000 samples with the sampling rate of 120fps. We sub-sample it with different ratios up to 100, which corresponds to a minimum sampling rate of 1fps, and show the corresponding estimation errors in Figure 4.10. For sequence 21, only a small set of samples are sufficient. On the other hand, for the complicated movements in sequence 27, denser samples are required. Overall our method is quite insensitive to sampling rate.

4.4.6 Qualitative Evaluation using Kinect

We have performed a qualitative comparison using the pose tracking algorithm in [97] (as of the year of 2011). Figure 4.11 show some examples of visual comparisons. The primary reason that we didn't compare quantitatively is due to the lack of ground-truth data. The markers used in a typical optical motion capture systems (which are considered as golden standard) will interfere with the Kinect Sensor. Nevertheless even just visual inspection can clearly demonstrate the improved accuracy in our method. The deficiencies visually

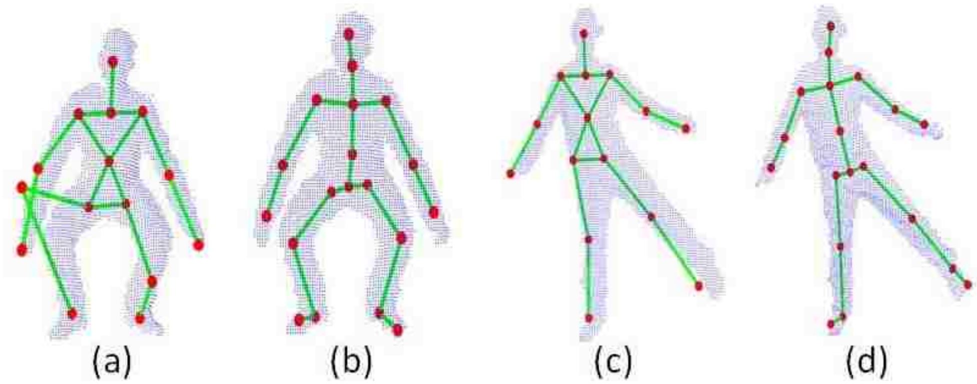


Figure 4.11: Examples of estimation results using pose tracking algorithms in [97]((a) and (c)) and our method ((b) and (d)), from depth images captured by Kinect.

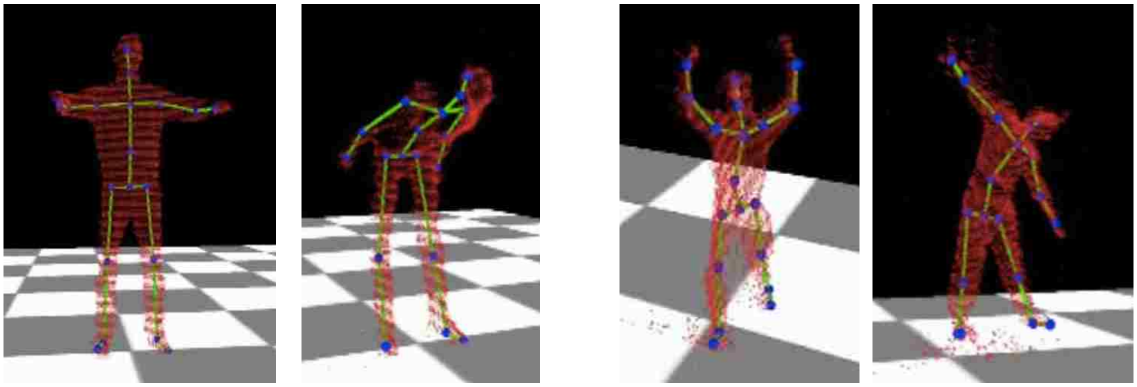


Figure 4.12: Examples of our experiments on two subjects with very different body sizes. The two images on the left are results on a 1.9m adult, while the other two on the right are results on a 1.6m child.

identified in [97] include (1) joint positions are not consistent with input depth maps when the subject is moving; (2) joint centers are unrealistically located on the surface, especially for the arms; and (3) catastrophic failure in simple motion (such as a crouch). A visual side-by-side comparison is presented in the supplementary video.

4.4.7 Body-size Invariance

Our method is capable of handling large body-size variations across subjects. Notice that the subject in the test above ($\approx 1.7m$) is different from both the subject in the public dataset and our template model. In order to further demonstrate this capability, we perform tests on a higher subject ($\approx 1.9m$) and a child ($\approx 1.2m$) as illustrated in Figure 4.12.

4.5 Conclusion

In this chapter, we present an effective pipeline that achieves highly accurate and robust pose estimation from a single depth image. The key insight is to combine data-driven pose detection with pose refinement. By using a prior database, we not only reduce the possible joint configuration space, but also provide an effective way to fill in the unobservable parts. Our pose refinement scheme can accommodate both pose difference and body-size difference. In addition we carefully design our pose detection algorithm to be view-independent. All these together dramatically reduce the size of the motion database – we only need motion samples synthesized from one generic human model. Quantitative evaluation shows that we achieve more than two times better accuracy than previous state-of-the-art (38mm vs. 100mm).

However, this approach, similar to other discriminative and hybrid motion capture approaches, does not preserve consistent skeletal structure over time which might be undesired in some applications. Starting from next chapter, we will introduce our two model-based algorithms that can provide real-time consistent pose and shape estimation, staring with preliminaries of our pose and shape representation.

Chapter 5 Real-Time Model-based Pose and Shape Estimation with Closest Point Strategy

In scenarios such as Virtual Try-On and medical applications, it is desired to maintain a consistent skeletal structure and shape during tracking. In this chapter, I present a generative tracker based on the Iterative Closest Point (ICP) strategy. Requiring only a generic human body template, our formulation is able to handle significant occlusion in a single depth map (≥ 50% of missing data) as well as maintain temporal consistency from the noisy depth input generated by commodity depth cameras. The key is our novel constraints that effectively guides the local optimization. The output of our tracking algorithm is a body mesh that accurately adapts to the user's motion and body shape. The mesh is complete and maintains the same topology over time.

Our tracking method is most similar to [50] and [62], using a twist-based pose representation. However, we propose a set of constraints that are experimentally found to

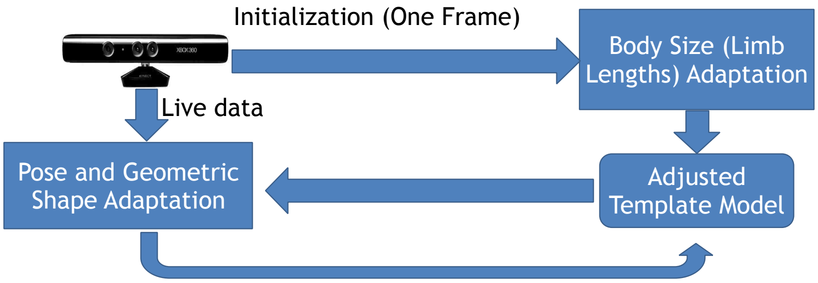

Initialization (One Frame)
Body Size (Limb Lengths) Adaptation
Live data
Pose and Geometric Shape Adaptation
Adjusted Template Model

Figure 5.1: The diagram of our pose tracking system. During initialization, our system takes in one single frame of depth map and adjusts the limb lengths of the template. For each frame of the live data, we perform pose and shape adaptation to adjust the template for better personalization.

be effective for accommodating monocular data and therefore improve the robustness of the tracker. Moreover, our linearization of the exponential map better satisfies the small quantity requirement for first-order approximation compared to [50] and leads to linear optimization compared to the nonlinear one in [62]. There are three components in our tracking system: twist-based pose estimation, surface adaptation and body size estimation during initialization as shown in Figure 5.1. Our template and the twist-based articulated deformation model has been introduced in Chapter 3. In the following sections, we first describe our pose estimation method based on the twist-representation followed by the shape and body size adaptation.

5.1 Pose Tracking via Closest Points

The goal of pose tracking is to estimate the change in pose given the current configuration Θ^t and an observation of the surface vertices in a new configuration Θ^{t+1}. Therefore, if the surface point correspondences between these two poses are given, we can use Equation 3.24 and estimate the change of pose by minimizing the following energy function

$$E_c = \sum_{m=1}^{M} \left\| \omega_m^{t+1} \left(\boldsymbol{v}_m(\Theta^{t+1}) - \boldsymbol{c}_m^{t+1} \right) \right\|^2 \tag{5.1}$$

$$\approx \sum_{m=1}^{M} \left\| \omega_m^{t+1} \left(\boldsymbol{v}_m(\Theta^t) + I_m^t \boldsymbol{\Delta\xi}_g^t + \sum_{k=1}^{n} \beta_{i,k} \hat{\xi'}_k^t \boldsymbol{v}_m(\Theta^t) \Delta\theta_k^t - \boldsymbol{c}_m^{t+1} \right) \right\|^2 \tag{5.2}$$

where target surface point $\boldsymbol{v}_m(\Theta^{t+1})$ and its correspondence $\boldsymbol{c}_m^{t+1}$ are weighted by ω_m^{t+1}. Since $\boldsymbol{v}_m(\Theta^t)$ is known, the change of pose $\Delta\Theta^t = [\boldsymbol{\Delta\xi}_g^t, \Delta\theta_1^t, \cdots, \Delta\theta_K^t]^T$ can be obtained by solving this linear equation.

In reality, the true correspondences are not known a priori and therefore should be estimated. Similar to [50] and [62], we adopt the simple yet popular Iterative Closest Point

(ICP) strategy. Consequently, Equation 5.2 is solved iteratively with the point correspondences updated at each iteration. The strategies used to find correspondence are very important to the performance of the tracker. Rusinkiewicz et al. [100] conducted comparative studies of several alternatives for each component in the ICP framework for single rigid object registration. We also investigated these options and found that the following strategies worked well in our case:

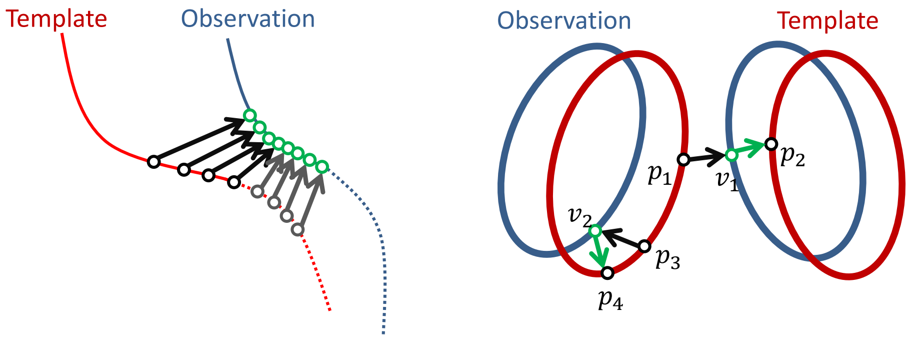
1. 3D closest point for correspondence finding;
2. Distance and normal thresholding for correspondence pruning;
3. Rejection of correspondences containing edge points.

The projection based scheme used in [50] is found to be more sensitive to occlusions and therefore discarded. KD-tree is used for speeding up the closest point search.

In order to make our system more robust in accommodating monocular data, we additionally propose the following strategies:

1. Visibility constraint;
2. Relaxed bijective consistency constraint;
3. Edge-to-edge correspondences.

Notice that we first reject correspondences containing edge points following the suggestions in [100], and then explicitly construct edge-to-edge point correspondences. Such choices are designed to avoid correspondences between inner points and edge points that sometimes are problematic. A typical situation is when part of the body moves out of the view and false edges are created at the boundary. This constraint can avoid the matching of these points and thus help the tracker better deal with such partial observation. In the



(a) Visibility Constraint (b) Relaxed Bijective Consistency Constraint

Figure 5.2: 2D examples illustrating the effectiveness of two of our constraints. The arrows connect a point to its closest point on the other surface. In (a), the dash lines representing occluded regions. The visible points (black circles) attempt to move to the right as desired; while the invisible points (the gray circles) tend to move the surface upward and could lead to incorrect local optima if not excluded. In (b), since p_1 and p_2 belong to two different segments, our relaxed bijective consistency constraint will reject $< p_1, v_1 >$, so will the original version. However, the pair $< p_3, v_2 >$ will also be rejected by the original version. By contrast, our relaxed version will accept it, and therefore, could guide the tracker more effectively.

following, we will first discuss the first two constraints for pruning correspondences and then demonstrate how we use the edge-to-edge correspondences to better guide the tracker.

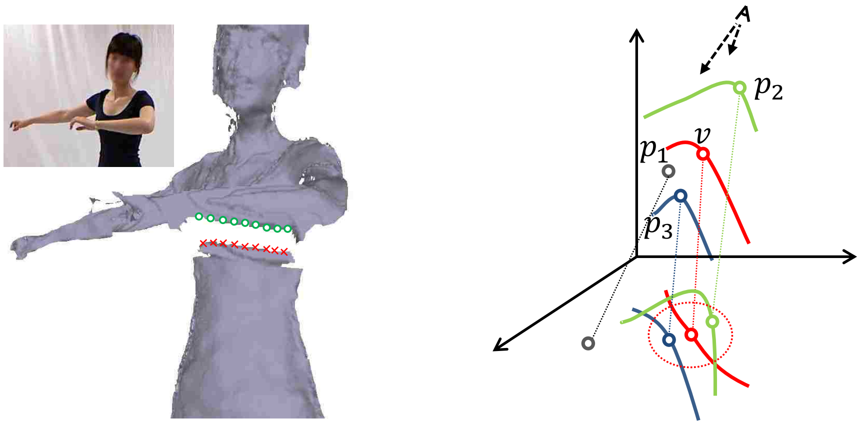
The visibility constraint means only visible points are used to construct point correspondences. On one hand, this strategy is important for dealing with monocular data by avoiding plenty of unnecessary computation and erroneous correspondences containing invisible points. Similar to the rejection of inner-to-edge correspondences mentioned above, the visibility constraint prevents the part of body that is out of camera view from causing errors. On the other hand, this constraint might reduce the power to handle large rotations, in which case the invisible points close to the visibility boundary are important for driving the surface towards its correct orientation. Therefore, we relax the constraint by including points close to the visibility boundary. More specifically, we render a depth map for the surface mesh and reject points whose projected depths are larger then the corresponding rendered depths by a certain threshold.

The relaxed bijective consistency constraint is designed to prevent the local closest point search from driving a surface vertex towards an observed point that is from another segment. Originally, the bijective consistency constraint requires a pair of points to be closest to each other among their own point sets to be considered as a correspondence [136]. However, this strictly bijective consistency constraint could be sensitive to noises and generally requires a relatively large number of iterations to converge. To overcome this limitation, we propose a *relaxed bijective consistency* that considers the consistency of joint belonging. Such constraint between a pair of points $< \boldsymbol{v}_m, \boldsymbol{c}_m >$ can be expressed as

$$b(m) \equiv b(f(\boldsymbol{c}_m)) \tag{5.3}$$

where $b(m)$ denotes the index of the joint that vertex m belongs to (same as in Chapter 3), and $f(\boldsymbol{c}_m)$ represents the index of the closest point of $\boldsymbol{c}_m$ on the surface mesh. This constraints requires that the closest point of the observation $\boldsymbol{c}_m$ on the surface mesh and the given surface vertex $\boldsymbol{v}_m$ belong to the same body segment. One could further set distance or normal thresholds between these two points; although we do not find this critical as evidenced by experimental observations. For our skinning mesh, we define $b(i)$ as the joint with maximum skinning weight for vertex i. Figure 5.2 illustrates how the proposed constraint can prevent incorrect matching situations while still preserving effectiveness in identifying correct ones.

Besides the correspondences between surface points, we further enforce edge constraints to guide the tracker, similar to [50, 51]. First of all, edge points are extracted from the depth maps of the observed surface as well as our surface mesh. Edges that are



(a) True (green) and false (red) edge points

(b) Advantage of our two stage edge correspondence search

Figure 5.3: Edge point correspondences. (a) The false edge points due to self-occlusion are not used as they do not correspond to true shape boundary. (b) The dash lines represents projection, and the red dash circle indicates the range of candidate points to match v based on 2D measurement. With closest point in 3D, v is matched to a noise p_1. With closest point in 2D, it is matched to p_2. With our two stage approach, it can be mapped to the point p_3 on the correct target. However, it cannot handle the situation that projection of p_1 or p_2 lies in the red circle. In most cases, it is better than direct 3D or 2D closest point strategy.

not on the silhouette boundary are mainly due to self-occlusion. For these edge points, we only keep those points from the closer part, and ignore the occluded part as they do not correspond to real surface boundary, as shown in Figure 5.3(a). For each of the template edge points, we find k_s nearest points in the 2D image plane, and keep the one closest in 3D space for reasons illustrated in Figure 5.3(b). The distance and normal thresholding are again applied. However, in this case, the normals are 2D normals computed from the edge contour, instead of 3D vertex normals that are generally inaccurate for edge points of the observed surface.

With a set of edge point correspondences $\{< \boldsymbol{v}_m(\Theta^t), \boldsymbol{c}_m^{t+1} > | m \in \mathcal{E}^t\}$ extracted, the edge constraint requires the projections of the surface points to lie on the projections of their correspondences, denoted as $\{< u(\boldsymbol{c}_m^{t+1}), v(\boldsymbol{c}_m^{t+1}) > | m \in \mathcal{E}^t\}$. Given the projection matrix P, the edge constraint can be formulated as [51]

$$\begin{bmatrix} \boldsymbol{p}_r^1 - u(\boldsymbol{c}_m^{t+1}) \cdot \boldsymbol{p}_r^3 \\ \boldsymbol{p}_r^2 - v(\boldsymbol{c}_m^{t+1}) \cdot \boldsymbol{p}_r^3 \end{bmatrix} \cdot \boldsymbol{v}_m(\Theta^t) = \begin{bmatrix} u(\boldsymbol{c}_m^{t+1}) \cdot \boldsymbol{p}_t^3 - \boldsymbol{p}_t^1 \\ v(\boldsymbol{c}_m^{t+1}) \cdot \boldsymbol{p}_t^3 - \boldsymbol{p}_t^2 \end{bmatrix}, \quad i \in \mathcal{E}^t \tag{5.4}$$

where $\boldsymbol{p}_r^i (i = 1, 2, 3)$ is the i^{th} row of the first 3×3 sub-matrix of P, and $\boldsymbol{p}_t^i$ is the i^{th} element of the fourth column of P. For simplicity, we denote the 2×3 matrix on the left-hand side as H_m^{t+1} and the 2D vector on the right-hand side as $\boldsymbol{h}_m^{t+1}$. Since we want the vertex positions under a new pose Θ^{t+1} to satisfy this constraint, we can combine Equation 3.24 and Equation 5.4 to form the edge energy function:

$$E_e = \sum_{m \in \mathcal{E}^t} \left\| \tau_m^{t+1} \Big(H_m^{t+1} (\boldsymbol{v}_m(\Theta^t) + I_m^t \cdot \boldsymbol{\Delta\xi}_g^t + \sum_{k=1}^{K} \beta_{m,k} \hat{\xi}'^t_k \boldsymbol{v}_m(\Theta^t) \Delta\theta_k^t) - \boldsymbol{h}_m^{t+1} \Big) \right\|^2 \tag{5.5}$$

Our final tracking energy function consists of a weighted combination of Equation 5.2 and Equation 5.5, as well as a regularization term that penalizes large pose change which is required according to the first order linearization:

$$E = E_s + \lambda_e E_e + \lambda_r \left(\left\| \boldsymbol{\Delta\xi}_g^t \right\|^2 + \sum_{k=1}^{K} \left\| \Delta\theta_k^t \right\|^2 \right) \tag{5.6}$$

The relative weight λ_r is set to 1, and λ_e is set as the inverse of the maximum of input depth map size in our experiments to avoid dependency on image size. For each new frame, this energy is iteratively optimized until the maximum movement of visible surface points is smaller than a given threshold, which is empirically set to 3mm in our experiments. Table. 5.1 summarizes the set of constraints and the parameters used for point matching in our tracker. The per-vertex weights $\{w_m\}$ and $\{\tau_m\}$ are currently set to 1 for points that satisfy these constraints and 0 otherwise.

Table 5.1: The constraints and parameters used for point correspondence construction.

	Distance	Normal	Ignore boundary	Visibility	Relaxed bijective
Surface	200mm	60°	Yes	Yes	Yes
Edge	200mm	90°	No	N/A	Yes

5.2 Surface Geometry Adaptation

After the pose has been estimated, the surface model should be fitted to the shape of the observation to ensure consistency between the final simulated clothes and the shape of the subject. We utilize both the captured surface geometry and edge information for this task. Note that the edge information from depth data is richer than silhouettes from color images. We rely on the same energy functions as in our pose tracker, namely Equation 5.2 and Equation 5.5. In general, the surface vertices are allowed to move in any direction. However, we constraint the movement of a vertex to be along its normal direction to partially overcome the ambiguities in monocular data. In fact, this constraint can partially prevent over-fitting the template to deformations of clothes. Therefore we only need to estimate the magnitudes of the movement, i.e. the displacements $\{d_m\}$. The energy function due to point correspondences, with the updated pose, then becomes

$$\begin{aligned} E_c^{\text{fit}} = &\sum_{m=1}^{M} \left\| \omega_m^{t+1} \Big(\boldsymbol{v}_m(\Theta^{t+1}) + \boldsymbol{n}_m^{t+1} \cdot d_m^{t+1} - \boldsymbol{c}_m^{t+1} \Big) \right\|^2 \\ &+ \lambda_e^{\text{fit}} \sum_{i \in \mathcal{E}} \left\| \tau_m^{t+1} \Big(H_m^{t+1} (\boldsymbol{v}_m(\Theta^{t+1}) + \boldsymbol{n}^{t+1} \cdot d_m^{t+1} - \boldsymbol{h}_m^{t+1}) \Big) \right\|^2 \end{aligned} \tag{5.7}$$

with H_m^{t+1} and $\boldsymbol{h}_m^{t+1}$ defined in Equation 5.4. The values of the per-vertex weights $\{\omega_m^{t+1}\}$ and $\{\tau_m^{t+1}\}$ are the same as in the pose tracker. Notice the vertex normals $\{\boldsymbol{n}_m^{t+1}\}$ are always calculated from the original surface mesh without displacement and then rotated based on the current pose. This strategy is designed to prevent surface distortion due to accumulated drifting in a long sequence of motions. Since before the fitting stage, the template mesh has been aligned to the observation with the estimated pose, we use a smaller distance threshold (50mm) for all correspondences, as well as a smaller angular threshold (60°) for edge correspondences. Similarly the global weight λ_e^{fit} is set to 2 to favor the edge correspondences.

Since this fitting process is performed for each frame independently, the fitted surface mesh might suffer from temporal jittering. This will eventually produce visual artifacts in cloth simulation. Therefore, we add another three energy functions to enforce temporal and spatial smoothness. The first term penalizes the distortion of the surface geometry via vertex Laplacian coordinates [51]; while the other two terms directly minimize large displacement changes temporally and spatially, respectively. The energy functions are defined as:

$$E_l^{\text{fit}} = \sum_{m=1}^{M} \left\| \sum_{<m,j>\in\mathcal{F}} (\boldsymbol{n}_j^{t+1} \cdot d_j^{t+1}) - \boldsymbol{n}_m^{t+1} \cdot d_m^{t+1} \right\|^2 \tag{5.8}$$

$$E_t^{\text{fit}} = \sum_{m=1}^{M} \frac{1}{{\sigma_m^2}^t} (d_m^{t+1} - \mu_m^t)^2 \tag{5.9}$$

$$E_s^{\text{fit}} = \sum_{<i,j>\in\mathcal{F}} (d_i^{t+1} - d_j^{t+1})^2 \tag{5.10}$$

Notice the first term is equivalent to minimizing the discrepancy of the Laplacian coordinates of the surface vertex with and without considering the displacement. μ_m^t and ${\sigma_m^2}^t$ are the mean and variance of the estimated displacements for vertex i up to frame t. One could alternatively measure the difference of displacement magnitude between consecutive frames. However, it does not prevent the surface from being overfitted in the presence of severe occlusions. Instead, our energy function forces the displacements to stabilize after sufficient observations are made, while still allowing flexibility for regions that are not rigid

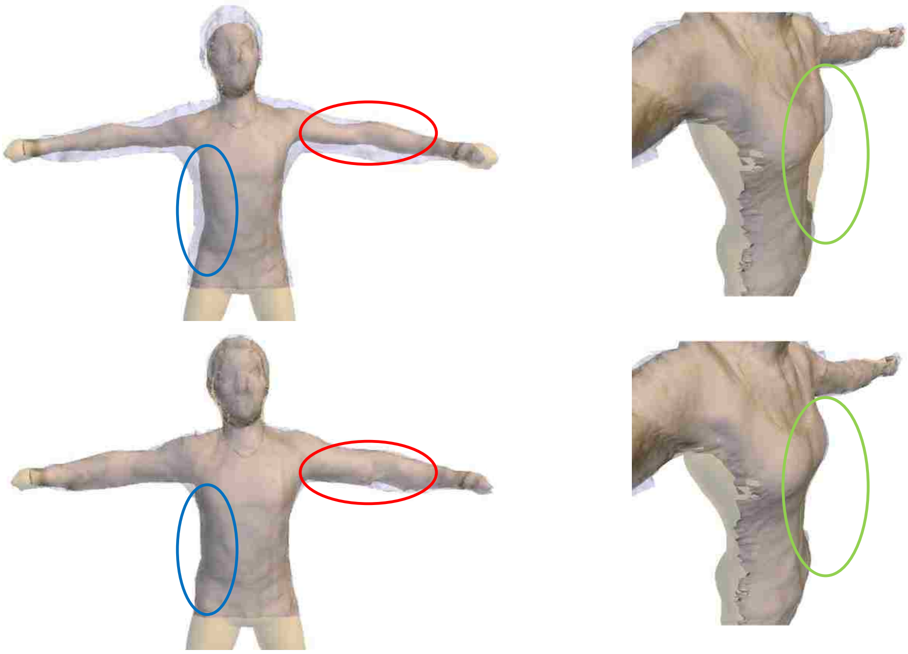


Figure 5.4: A male (left) and a female (right) example showing the surface estimation results. The first row shows the template without displacements overlayed on the input, while the second row shows the one with displacements. The highlighted regions demonstrate its effectiveness.

via the variance-based weighting. Both the means and variances are updated in an online fashion.

Our final fitting energy is then the weighted combination of all the terms defined above:

$$E^{\text{fit}} = E_c^{\text{fit}} + \lambda_l^{\text{fit}} E_l^{\text{fit}} + \lambda_t^{\text{fit}} E_t^{\text{fit}} + \lambda_s^{\text{fit}} E_s^{\text{fit}} \tag{5.11}$$

with empirical settings of $\lambda_l^{\text{fit}} = 4, \lambda_t^{\text{fit}} = 6$ and $\lambda_s^{\text{fit}} = 2$. The term E_t is set to take effect only after a certain number of frames (20 in our experiments) so that the means and variances are better estimated. For each frame, after the displacements are estimated, they are incorporated in the skinning model for pose estimation for the next frame. As the shape consistency increases via this fitting process, the accuracy of the pose estimation is also improved. Figure 5.4 shows two examples of adapting the shape of our template model to the observed meshes.

5.3 Body Size Adaptation

Generative approaches generally need to address the issue of body sizes variations between the template and the subject. The effectiveness of a pose tracker usually relies heavily on such consistency. Our method iterates between updating the pose and adjusting the limb lengths, similar to [53]. We utilize the differential bone coordinates introduced by Straka et al. [110]. While their method results in non-linear optimization when incorporating constraints for limb lengths (e.g body symmetry or fixed lengths for some limbs), our method is always linear.

The differential body coordinates is defined by Straka et al. [110] in a way similar to the Laplacian coordinates:

$$\boldsymbol{\eta}_m = \sum_{k=1}^{n} \alpha_{m,k}\Big(\boldsymbol{v_m} - (\boldsymbol{g}_k - (1-\gamma_{m,k})\boldsymbol{d}_k)\Big) \tag{5.12}$$

where $\{\alpha_{m,k}\}$ are the skinning weights, $\boldsymbol{g}_k$ and $\boldsymbol{d}_k$ are the position of joint k and the vector from its parent joint to $\boldsymbol{g}_k$, respectively. The coefficient $\gamma_{m,k}$ is chosen such that the line between $\boldsymbol{v}_m$ and $\boldsymbol{\eta}_{m,k}$ is orthogonal to the bone vector $\boldsymbol{d}_k$. Each vector $\boldsymbol{\eta}_{m,k}$ encodes the relative position of the vertex $\boldsymbol{v}_m$ to bone k, while the differential bone coordinate $\boldsymbol{\eta}_m$ accumulates over all the controlling bones of this vertex.

To adapt the limb lengths of our template model to the observation in pose Θ^t, we estimate a scale for each limb so that the scaled surface mesh best matches the observation. With the set of scales $\mathcal{S} = \{s_k\}$, the bone vectors under the given pose then become $\{s_k\boldsymbol{d}_k^t\}$

and the joint positions can be computed as

$$\boldsymbol{g}_k(\Theta_t, \mathcal{S}) = \boldsymbol{g}_r(\Theta^t) + \sum_{j=1}^{n} \delta_{j,k} s_j \boldsymbol{d}_j^t \tag{5.13}$$

where $\boldsymbol{g}_r(\Theta^t)$ is the global position (position of the root) in pose Θ^t. According to Equation 5.12, we can reconstruct the scaled vertex positions $\{\boldsymbol{v}_m(\Theta^t, \mathcal{S})\}$ as follows

$$\boldsymbol{v}_m(\Theta^t, \mathcal{S}) = \boldsymbol{\eta}_m + \sum_{k=1}^{n} \alpha_{m,k}\Big(\boldsymbol{g}_k(\Theta^t, \mathcal{S}) - (1 - \gamma_{m,k}) s_k \boldsymbol{d}_k^t\Big) \tag{5.14}$$

Substitute Equation 5.13 into Equation 5.14, we have

$$\begin{aligned}
\boldsymbol{v}_i(\Theta^t, \mathcal{S}) &= \boldsymbol{\eta}_i + \sum_{k=1}^{n} \alpha_{i,k}\left(\boldsymbol{g}_r(\Theta^t) + \sum_{j=1}^{n} \delta_{j,k} s_j \boldsymbol{d}_j^t - (1 - \gamma_{i,k}) s_k \boldsymbol{d}_k^t\right) \\
&= \boldsymbol{\eta}_i + \boldsymbol{g}_r(\Theta^t) + \sum_{k=1}^{n} \alpha_{i,k} \sum_{j=1}^{n} \delta_{j,k} s_j \boldsymbol{d}_j^t - \sum_{k=1}^{n} \alpha_{i,k}(1 - \gamma_{i,k}) s_k \boldsymbol{d}_k^t) \\
&= \boldsymbol{\eta}_i + \boldsymbol{g}_r(\Theta^t) + \sum_{j=1}^{n} \beta_{i,j} s_j \boldsymbol{d}_j^t - \sum_{k=1}^{n} \alpha_{i,k}(1 - \gamma_{i,k}) s_k \boldsymbol{d}_k^t \\
&= \boldsymbol{\eta}_i + \boldsymbol{g}_r(\Theta^t) + \sum_{j=1}^{n} \rho_{i,j} s_j \boldsymbol{d}_j^t \qquad (5.15) \\
\text{where} \quad & \rho_{i,k} = \beta_{i,k} - \alpha_{i,k}(1 - \gamma_{i,k}) \qquad (5.16)
\end{aligned}$$

The consistency of the scaled surface mesh and the observation is again measured via the distance between point correspondences. Specifically, we use the strategies described in Sec. 5.2 to construct point correspondences, and then minimize the energies in Equation 5.2 and Equation 5.5 parametrized on the scales:

$$E_c^s = \sum_{m=1}^{M} \left\| \omega_m\Big(\boldsymbol{v}_m(\Theta^t, \mathcal{S}) - \boldsymbol{c}_m^t\Big) \right\|^2 + \lambda_e^s \sum_{i \in \mathcal{E}} \left\| \tau_m\Big(H_m^t \boldsymbol{v}_m(\Theta^t, \mathcal{S}) - \boldsymbol{h}_m^t\Big) \right\|^2 \tag{5.17}$$

Substituting Equation 5.15 into Equation 5.17, we obtain the following energy function:

$$
\begin{aligned}
E_c^s = \sum_{m=1}^{M} \left\| \omega_m \Big(\sum_{k=1}^{n} \rho_{m,k} s_k \boldsymbol{b}_k^t + \boldsymbol{\eta}_m + \boldsymbol{g}_r(\Theta^t) - \boldsymbol{c}_m \Big) \right\|^2 \\
+ \lambda_e^s \sum_{m \in \mathcal{E}} \left\| \tau_m \Big(H_m \big(\sum_{k=1}^{n} \rho_{m,k} s_k \boldsymbol{b}_k^t + \boldsymbol{\eta}_m + \boldsymbol{g}_r(\Theta^t) \big) - \boldsymbol{h}_m \Big) \right\|^2
\end{aligned} \tag{5.18}
$$

In addition, we add a regularization term that enforces scale consistency between a set of pre-defined bone pairs $\mathcal{B} = \{< j,k >\}$. The consistency enforces symmetry, e.g. left leg vs. right leg, as well as connected bone consistency, e.g. left upper leg vs. left lower leg. Therefore our final scale estimation energy function can be written as:

$$
E^s = E_c^s + \lambda_r^s \sum_{<j,k> \in \mathcal{B}} \upsilon_{j,k} (s_j - s_k)^2 \tag{5.19}
$$

In our experiments, we set the weights $\{\upsilon_{j,k}\}$ to 1 for body symmetry constraints and 0.5 for the connected bone consistency. To perform the body size adaptation, our method iterates between the pose estimation and the scales estimation. However, we require each step to converge before switching to another one (see Algorithm 1). This procedure is more stable and effective than switching at each iteration. The body size fitting is performed only during initialization for each subject to prevent over-fitting in the tracking procedure, for example due to partial observation. Figure 5.5 shows an example of the template scaled to fit to the observation. The pose in this figure (the calibration pose) is preferable for the scale estimation as it implicitly encodes the position of the real joints and therefore helps to provide correct estimates of the scales.

5.4 The Tracking Pipeline

For a new subject, our system starts with the body size adaptation and then moves to the regular tracking-fitting procedure. In general, both a global transformation and a rough initial pose should be given due to the local nature of our method. The initial global transformation can be estimated via principal axes if the entire body of the subject is observable; via other discriminative approaches [104,133]; or simply via human interaction. We do not intend to address this general problem but instead assume the rough initial alignment is given. As for the local body pose, we assume the that subject starts with a T-pose. However, as long as it is not dramatically different from the starting pose of our template, our tracker is able to drive the template to the initial subject pose. In practice, a frontal view pose is preferable due to its relatively small ambiguities. The entire pipeline of our tracking system is summarized in Algorithm 1.

5.5 Experimental Results

We tested our system on an Intel Core i5-2500K 3.3GHz CPU with an NVIDIA Tesla C2075 GPGPU processor. Our input data are captured using a single Kinect camera with a resolution of 640×480 and sampling frequency of 30FPS . As our tracker does not require the entire body to be observed, except for the body size adaptation step, there is no specific restriction on the camera position and orientation. However, even though our tracking algorithm can deal with some noise, we do assume the input data are segmented and only the part of the surfaces belonging to the subject is provided to our tracker. Assuming the camera is fixed during data acquisition, we fit a plane to the ground and remove points that

```
Data: Captured surface sequence, template model, calibration pose $\Theta^c$ and initial
      tracking pose $\Theta^0$.
Result: Scales $\mathcal{S}$, body poses $\{\Theta^t\}$ and displacements $\{d^t_m\}$.
begin Body size adaptation
    Initialize the template with pose $\Theta^c$ and Scales $\mathcal{S}' = \{1\}$
    while Scales $\mathcal{S}$ not converged do
        Update template with scales $\mathcal{S}$
        while Pose not converged do
            Construct correspondences
            Estimate pose via Equation 5.6
        end
        while Scales $\mathcal{S}'$ not converged do
            Construct correspondences
            Estimate scales via Equation 5.19
        end
        Set $\mathcal{S} = \mathcal{S}'$
    end
end
Initialize the template with pose $\Theta^0$ and scales $\mathcal{S}$
for Each frame t do
    while Pose $\Theta^t$ not converged do
        Construct correspondences
        Estimate pose via Equation 5.6
    end
    Construct correspondences
    Estimate displacements via Equation 5.11
    Update template with the displacement;
end
```

Algorithm 1: Our pose tracking pipeline.

are close to the plane by some threshold (20mm for our test data). The background can be removed via depth thresholding in simple scenario. In our setup, we put a curtain on the background and also use plane fitting to remove the background points, yet with a larger distance threshold (200mm). In the rest of this section, we discuss the performance of both our pose tracker.

The template mesh model we use contains 12894 vertices. The number of vertices in the segmented input data is generally between 20K and 45K as the distance of the subject

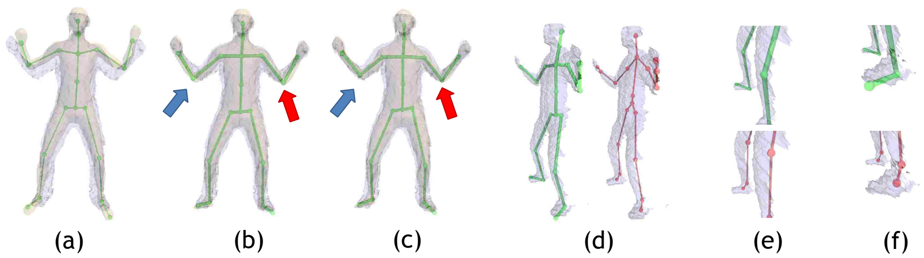


Figure 5.5: An example showing results of our body size adaptation algorithm. In the first row, the input mesh, template mesh and template skeletons are shown together. (a) the initial step of the algorithm. (b) results with only pose estimation. (c) the adapted results. Notice the improvement for the joints indicated by the arrows. The second row compares our adapted skeleton, with the skeleton estimated by Kinect SDK [4]. (d) our skeleton (green) and the Kinect skeleton side by side. (e) and (f) comparisons of these two skeletons with close-up views for the legs and the feet, respectively. Discussions are provided in the text.

from the camera changes. Our tracking system is implemented on the GPU using CUDA and processes 40 to 60 frames per second, depending on the degree of change in pose, i.e. speed of motion. The body size adaptation usually takes less than 100ms to converge and is only performed during initialization. The overall tracking quality is shown in our supplemental video.

In Figure 5.4, we show two representative examples of our surface adaptation results. Notice that when the subject is far away from the camera, and the captured geometry does not reflect the true body shape, our method will only attempt to fit to the observation and cannot overcome this limitation of the sensor. In our accompanied video, the consistency of the adapted template and the captured surface can be visually checked, which we believe is in general sufficient for our application.

Similarly, our body size adaptation algorithm can effectively capture the user's body size as shown in Figure 5.5. Our pose tracker can correctly align the template and the

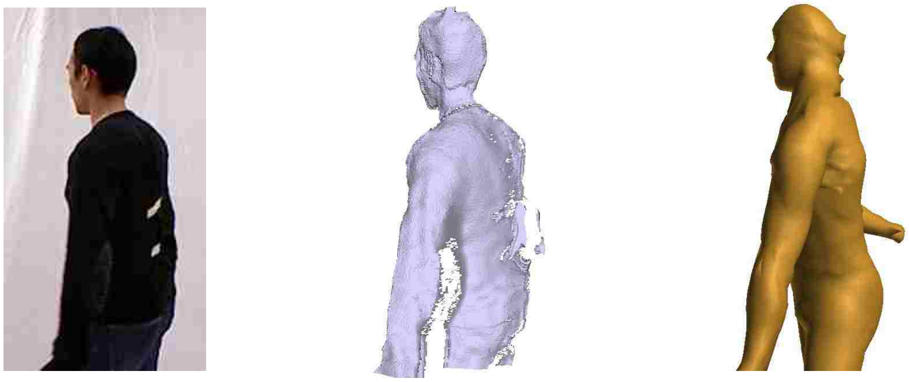


Figure 5.6: A failure case for our pose tracking. The torso of the template (right) it not properly rotated to fit to the input surface (middle). In addition, the right arm is problematic as well due to lack of constraint.

observation in the calibration pose (from (a) to (b)), despite of the actual pose difference. Secondly, as highlighted with the arrows in Figure 5.5(b) and (c), the refined arm joints are closer to their true positions, meaning the arms are properly scaled to fit to the user's body size. The second row compares our results with the skeletons estimated by the Kinect SDK [4]. As we can see from the close-up views in (e) and (f), our estimated joints more properly reflect the true joint locations, for example of the feet.

More evaluations are presented in Section 8.1, where this pose tracking component is combined with a cloth simulation engine to deliver a realistic Virtual Try-On system.

5.6 Limitations and Conclusion

Due to the inherent ambiguity for pose tracking using monocular data, as well as the local nature of our tracking method, our tracker could fail in some situations. One such case we observed is torso rotation while the body is only partially observed (Figure 5.6). With such a severe occlusion, the point matching is ambiguous because the torso (the majority of the visible parts) is close to a cylindrical shape. To solve this issue, one option is to predict the pose with some motion model as in [51]. Another limitation is the lack of constraints for invisible parts, as also can be seen from the right arm of the template in Figure 5.6. Our visibility constraint can deal with it in most cases, however a more sophisticated approach might be to incorporate the free space constraint [53] into our framework. The algorithm presented in Chapter 6 improves this technique through probabilistic correspondences association and is less sensitive to local minima.

Chapter 6 Real-Time Model-based Pose and Shape Estimation with Probabilistic Correspondences Association

In this chapter, we present a novel (generative) articulated pose estimation algorithm that **does not require explicit point correspondences and captures the subject's shape automatically during the pose estimation process.** This algorithm relates the observed data with our template using Gaussian Mixture Model (GMM), without explicitly building point correspondences. The pose is then estimated through probability density estimation under articulated deformation model parameterized with exponential maps [23]. Consequently, the algorithm is **less sensitive to local minima and well accommodates fast and complex motions**. In addition, we develop a novel shape estimation algorithm within the same probabilistic framework that is seamlessly combined with our pose estimation component.

Our pose tracker uses a skinning mesh model as template as introduced in Chapter 3. Specifically, the template consists of four components, the surface vertices $\mathcal{V} = \{\boldsymbol{v}_m^0 | m = 1,\cdots,M\}$, the surface mesh connectivity $\mathcal{F}$, the skinning weights $\mathcal{A}$ and the skeleton. The goal of pose estimation is to identify the pose Θ, under which the deformed surface vertices, denoted as $\mathcal{V}(\Theta) = \{\boldsymbol{v}_m^\Theta | m = 0,\cdots,M\}$, best explain the observed point cloud $\mathcal{X} = \{\boldsymbol{x}_n | n = 1,\cdots,N\}$.

6.1 The General Probabilistic Model

Our algorithm assumes that the observed point cloud $\mathcal{X}$ follows a Gaussian Mixture Model (GMM) whose centroids are the deformed template vertices $\mathcal{V}(\Theta)$. Similar to [89], an extra uniform distribution is included to account for outliers. Therefore, the probability of each observed point $\boldsymbol{x}_n \in \mathcal{X}$ can be expressed as

$$p(\boldsymbol{x}_n) = (1-u)\sum_{m=1}^{M} p(\boldsymbol{v}_m^{\Theta})p(\boldsymbol{x}_n|\boldsymbol{v}_m^{\Theta}) + u\frac{1}{N} \tag{6.1}$$

$$p(\boldsymbol{x}_n|\boldsymbol{v}_m^{\Theta}) = \frac{1}{(2\pi\sigma^2)^{d/2}}\exp\Big(\frac{-||\boldsymbol{x}_n - \boldsymbol{v}_m^{\Theta}||^2}{2\sigma^2}\Big) \tag{6.2}$$

where d is the dimensionality of the point set ($d = 3$ in our case) and u is the weight of the uniform distribution that roughly represents the percentage of the outliers in $\mathcal{X}$. Here a single variance parameter σ^2 is used for all Gaussians for simplicity. Similar to [35, 89], we assume uniform distribution for the prior, that is $p(\boldsymbol{v}_m^{\Theta}) = 1/M$.

Under this probabilistic mode, pose estimation is cast as a probability density estimation problem that minimizes the following negative log-likelihood:

$$E(\Theta, \sigma^2) = -\sum_{n=1}^{N}\log\Big(\sum_{m=1}^{M}\frac{1-u}{M}p(\boldsymbol{x}_{\boldsymbol{n}}|\boldsymbol{v}_m^{\Theta}) + \frac{u}{N}\Big) \tag{6.3}$$

which is normally solved iteratively using the Expectation-Maximization (EM) algorithm [40]. During the E-step, the posteriors $p^{\text{old}}(\boldsymbol{v}_m^{\Theta}|\boldsymbol{x}_n)$ are calculated using the parameters estimated from the previous iteration based on Bayes rule:

$$p_{mn} \equiv p^{\text{old}}(\boldsymbol{v}_m^{\Theta}|\boldsymbol{x}_n) = \frac{\exp\Big(\frac{-||\boldsymbol{x}_n - \boldsymbol{v}_m^{\Theta}||^2}{2\sigma^2}\Big)}{\sum_{m=1}^{M}\exp\Big(\frac{-||\boldsymbol{x}_n - \boldsymbol{v}_m^{\Theta}||^2}{2\sigma^2}\Big) + u_c} \tag{6.4}$$

where $u_c = \frac{(2\pi\sigma^2)^{d/2}uM}{(1-u)N}$. During the M-step, the parameters are updated via minimizing the following complete negative log-likelihood (upper bound of Equation 6.3):

$$\begin{aligned} Q(\Theta, \sigma^2) &= -\sum_{n,m} p_{mn}\Big(\log\big(\frac{1-u}{M}p(\boldsymbol{x}_{\boldsymbol{n}}|\boldsymbol{v}_m^{\Theta})\big) + \log\frac{u}{N}\Big) \\ &\propto \frac{1}{2\sigma^2}\sum_{n,m} p_{mn}||\boldsymbol{x}_n - \boldsymbol{v}_m^{\Theta}||^2 + \frac{d}{2}P\log\sigma^2 \end{aligned} \tag{6.5}$$

$$\text{where } P = \sum_{n,m} p_{mn}; \quad \sum_{n,m} \equiv \sum\nolimits_{n=1}^{N}\sum\nolimits_{m=1}^{M} \tag{6.6}$$

So far, the probabilistic model is independent of the form of deformation model in $\mathcal{V}(\Theta)$. Cui et al. [35] and Myronenko et al. [89] used this model for rigid and non-rigid point set registration. Different from their work, we derive the pose estimation formulation under the articulated deformation model, which is more suitable for a large variety of articulated-like objects (e.g., human and many animals).

6.2 The Tracking Algorithm

With the assumption of quasi-articulated motion, we again adopt the LBS deformation model parameterized with exponential maps introduced in Chapter 3. The core of our tracking algorithm is the combination of the twist-based articulated deformation model with the probabilistic framework in Sec. 6.1. Substitute Equation 3.24 into Equation 6.5, we get the following objective function (superscript ignored for notational simplicity):

$$\begin{aligned} Q(\boldsymbol{\Delta\Theta},\sigma^2) &= \frac{1}{2\sigma^2}\sum_{n,m}\Big(p_{mn}\|\boldsymbol{x}_n - \boldsymbol{v}_m - I_m\boldsymbol{\Delta\xi}_g - \sum_{k=1}^{K}\beta_{m,k}\hat{\xi}'_k\boldsymbol{v}_m\Delta\theta_k\|^2\Big) + \frac{d}{2}P\log\sigma^2 \\ &= \sum_{n,m}\frac{p_{mn}}{2\sigma^2}\|\boldsymbol{x}_n - \boldsymbol{v}_m - A_m\boldsymbol{\Delta\Theta}\|^2 + \frac{d}{2}P\log\sigma^2 \qquad (6.7) \end{aligned}$$

$$\text{where } A_m = [I_m \;\; \beta_{m,1}\hat{\xi}'_1\boldsymbol{v}_m \;\; \cdots \;\; \beta_{m,K}\hat{\xi}'_K\boldsymbol{v}_m] \qquad (6.8)$$

$$\boldsymbol{\Delta\Theta} = [\boldsymbol{\Delta\xi}_g^T \;\; \Delta\theta_1 \;\; \cdots \;\; \Delta\theta_K]^T \qquad (6.9)$$

In order to solve for the parameters $\{\boldsymbol{\Delta\Theta},\sigma^2\}$, the partial derivatives of $Q(\boldsymbol{\Delta\Theta},\sigma^2)$ over the parameters are set to zero to obtain the following equations:

$$\sum_{n,m}\frac{p_{mn}}{\sigma^2}A_m^T A_m\boldsymbol{\Delta\Theta} = \sum_{n,m}\frac{p_{mn}}{\sigma^2}A_m^T(\boldsymbol{x}_n - \boldsymbol{v}_m) \qquad (6.10)$$

$$\sigma^2 = \frac{1}{dP}\sum_{n,m}p_{mn}\|\boldsymbol{x}_n - \boldsymbol{v}_m\|^2 \qquad (6.11)$$

```
Initialize the template with previous pose
Sample a subset of points from each point set
while Pose not converged do
    E-step: Compute posteriors via Equation 6.4.
    M-Step:
        • Minimize Equation 6.14 for $(\mathbf{\Delta\Theta}, \sigma^2)$.
        • Update template vertices via Equation 3.6.
end
```

Algorithm 2: The pose estimation procedure.

These two equations comprise the core of our novel tracking algorithm. To better regularize the optimization, we add the following two terms to the objective function:

$$E_r(\mathbf{\Delta}\Theta) = ||\mathbf{\Delta}\Theta||^2 \tag{6.12}$$

$$E_p(\mathbf{\Delta}\Theta) = \sum_{k=1}^{K} (\theta_k^{\text{prev}} + \sum_{\tau=1}^{t} \Delta\theta_k^{\tau} - \theta_k^{\text{pred}})^2 \tag{6.13}$$

The term E_r ensures that the solution complies with the small pose change assumption during linearization in Equation 3.24. In E_p, $\{\theta_k^{\text{prev}}\}$ are the joint angles from previous frame, and $\{\theta_k^{\text{pred}}\}$ are the predicted joints angles using linear third order autoregression similar to [51]. The second term penalizes a solution's large deviation from the prediction, assuming relatively continuous motions in tracking scenario. This term is helpful in dealing with occlusions, in which case the joints corresponding to invisible parts can be relatively well constrained.

Our complete objective function is the weighted sum of these three terms:

$$E = Q(\mathbf{\Delta\Theta}, \sigma^2) + \lambda_r E_r(\mathbf{\Delta}\Theta) + \lambda_p E_p(\mathbf{\Delta}\Theta) \tag{6.14}$$

The partial derivative of E_r and E_p over $\mathbf{\Delta}\Theta$ are added to Equation 6.10 and the entire linear system is solved at each iteration for the pose update $\mathbf{\Delta}\Theta$. Equation 6.11 is solved for the Gaussian variance σ^2. After each pose update, the surface vertices are updated via the skinning deformation in Equation 3.6. In the E-step, the posteriors are calculated according to Equation 6.4. The procedure iterates until the maximum surface vertex movements is below a certain threshold ($1mm$ in our experiments). **Note that the small pose update is only enforced between two iterations, while large pose change between two frames is allowed.**

Since the computational complexity is in the order of MN, we use only a subset from each point cloud during the optimization. For the template mesh, random sampling strategy is used. The observed point set is uniformly sub-sampled based on the regular image grid. The pose estimation process for each new frame is summarized in Alg. 2.

6.3 Monocular Scenario

In monocular setup, the missing data introduces additional difficulties for the algorithm above. Specifically the algorithm attempts to use all given template vertices to fit the observed data. However, since the template surface is complete while the observed surface is partial, the observed partial surface will typically end up inside the complete template surface (between frontal and back surfaces). An intuitive strategy is to use only the visible part of the template from previous frame, as being adopted by Helten et al. [62]. However, such strategy can not well handle rotation of body parts, as the visible parts will change. Towards this end, we propose a two-step coarse-to-fine strategy to handle this situation. In the first step, the entire set of template vertices are used for sampling, and the pose are updated until convergence. In most cases, the body parts will be very close to their correct places. Then the visibility of the template surface is determined and only the visible part

```
begin Step One
    Exclude template points outside the camera view;
    Perform Alg. 2;
end
begin Step Two
    Perform visibility test and exclude invisible points from the template;
    Perform Alg. 2;
end
```

Algorithm 3: Pose estimation for monocular data.

are used to refine the pose.

Another type of missing data is the occlusion of an entire body segment, either due to self occlusion or camera field of view limitation. By using only visible points in the second refining step, these two issues could be partially resolve, as the joint angles of the corresponding invisible parts will not be updated. However, they might still be affected during the first step. Therefore, we limit our sampling candidates to the set of template vertices inside the camera view. Besides, we rely on the autoregression prediction in Equation 6.13 to constraint the occluded parts that are inside the camera view. With these strategies, our algorithm can effectively and reliably estimate the pose using only one single depth camera. The entire pose estimation procedure is summarized in Alg. 3.

6.4 Template Shape Adaptation and Initialization

The consistency of body shape (limb lengths and body part girths) between the template and the subject plays a critical role in pose estimation. In this section, we describe our novel algorithm for automatic body shape estimation, followed by the initialization procedure of our tracker.

6.4.1 Limb Lengths Estimation

In order to adjust the limb lengths of template to fit the subject, existing methods either assume presence of a personalized template model [51] or estimate the body size before tracking [62, 128]. We follow the later strategy because apparently it is more general. However, our method requires neither parametric model as in [62], nor body part detectors as in [128]. Instead, we adopt the parametrization of surface vertices on limb length scales introduced in Section 5.3 and utilize the probabilistic model in Sec. 6.1 to estimate the optimal body size.

The adjustment of the limb lengths are achieved by introducing a scale for each bone, denoted as $\mathcal{S} = \{s_k\}$. The idea of our limb lengths adaptation is to estimate the scales, such that the vertices of the scaled template defined in Equation 5.14 maximizes the objective function defined in Equation 6.7. Therefore, substitute Equation 5.14 into Equation 6.7 and change the unknown from pose to the scales, we obtain the following objective function for limb length scales estimation:

$$Q(\mathcal{S}) = \sum_{nm} \frac{p_{mn}}{2\sigma^2}\Big(\sum_{k=1}^{K} \rho_{mk} s_k \boldsymbol{d}_k + \boldsymbol{\eta}_m + \boldsymbol{g}_r - \boldsymbol{x}_n\Big) \qquad (6.15)$$

Again, setting the partial derivatives of the above objective function over the scales to zero yields:

$$\begin{aligned} &\textstyle\sum_{j=1}^{K} \Big(\frac{1}{\sigma^2} \boldsymbol{d}_k^T \boldsymbol{d}_j \sum_{n,m} p_{mn} \rho_{mj} \rho_{mk}\Big) s_j = \frac{\boldsymbol{d}_k^T}{\sigma^2} \cdot \\ &\sum_{n,m} p_{mn} \rho_{mk} (\boldsymbol{x}_n - \boldsymbol{\eta}_m - \boldsymbol{g}_r); \quad k = 1, \cdots, K \end{aligned} \qquad (6.16)$$

As in Section 5.3, we further enforce consistency of the estimated scales for a set of

```
while Scales not converged do
    Estimate pose $\Theta$ using Alg. 3;
    Calculate the joints $\{\boldsymbol{g}_k, \boldsymbol{g}_r\}$ and $\{\boldsymbol{d}_k\}$ from $\Theta$;
    Compute $\{\boldsymbol{\eta}_m\}$ according to Equation 5.12;
    Estimate scales $\mathcal{S}$ via Equation 6.16 and Equation 6.17;
    Update the template with the scales $\mathcal{S}$
end
```

Algorithm 4: Limb lengths estimation procedure.

joints pairs in order to assure a reasonable overall shape of the scaled template:

$$\lambda_s \omega_{i,j} s_i = \lambda_s \omega_{i,j} s_j; \quad < i, j > \in \mathcal{B} \tag{6.17}$$

Here the global weight λ_s leverages the relative importance of this regularization term with respect to the data term in Equation 6.16, and $\omega_{i,j}$ represents the relative strength of the similarity for the pair (s_i, s_j) among others. For human and animals, we set $\omega_{i,j} = 1$ for symmetric parts (e.g left/right arm) and 0.5 for connected bones. Combining Equation 6.16 and Equation 6.17, we can solve for the scales so that the template is best fit to the observed points under our probabilistic model.

Notice that the parametrization in Equation 5.14 requires unchanged pose. However, before the template is appropriately scaled, the pose might not be well estimated. Therefore, we iterate between pose estimation and scale estimation until the estimated scales converge. The procedure is summarized in Alg. 4. The effectiveness of our template limb length adaptation is illustrated in Figure 6.7 and our supplemental video.

6.4.2 Geometric Shape Adaptation

Besides limb length adaptation, we further develop an automatic method to capture the overall geometry of the subject directly inside the pose estimation process, which does not require the subject to perform any additional specific motions as in [62]. The key insight is that upon convergence of pose estimation, the maximum of posteriors $p(\boldsymbol{v}_m|\boldsymbol{x}_n)$ over all $\{x_n\}$ naturally provide information about the point correspondences. Moreover, the corresponding posteriors can serve as a measure of uncertainty. With a sequence of data, we can treat each such correspondence as an observed samples of our target adapted surface vertex $\hat{\boldsymbol{v}}_m$. Consequently, the weighted average over all the samples can be used to represent our adapted template:

$$\bar{\boldsymbol{v}}_m = \sum\nolimits_f \omega(\boldsymbol{x}^f_{(m)})\boldsymbol{x}^f_{(m)} \Big/ \sum\nolimits_f \omega(\boldsymbol{x}^f_{(m)}) \tag{6.18}$$

where $\boldsymbol{x}^f_{(m)}$ is the correspondence identified via maximum of posteriors at frame f. The weight $\omega(\boldsymbol{x}^f_{(m)})$ is designed to take into account both the uncertainty based on the posterior and the sensor noise based on depth, and is defined as follows:

$$\omega(\boldsymbol{x}^f_{(m)}) = p(\boldsymbol{v}_m|\boldsymbol{x}^f_{(m)})/[\boldsymbol{x}^f_{(m)}]^2_z \tag{6.19}$$

where the quantity $[\cdot]_z$ denotes the z (depth) component.

In order to ensure smoothness of final adapted surface and to further handle noises, we allow the movement of a surface vertex only along its original normal. Consequently, a displacement d_m is estimated for each vertex $\boldsymbol{v}_m$, by optimizing the following objective function:

$$E_d = \sum_{m=1}^{M} \left(\omega_m \left\| d_m \boldsymbol{n}_m - (\bar{\boldsymbol{v}}_m - \boldsymbol{v}^0_m) \right\|^2 + \lambda_d \left\| d_m \right\|^2 \right) + \lambda_e \sum_{<m,l>\in\mathcal{F}} \left\| d_m - d_l \right\|^2 \tag{6.20}$$

where the weight $\omega_m = 1$ if the vertex has correspondence up to the current frame and 0 otherwise. The first term moves the vertex to the projection of the weighted sum defined in Equation 6.18 on the normal $\boldsymbol{n}_m$. The second term penalizes large movements and the third one enforces smoothness between connected vertex pairs. In our implementation, we prune correspondences based on euclidean distance in each frame to remove noises, and perform this adaptation only every L ($L = 5$ in our experiments) frames because nearby frames in general provide little new information.

6.4.3 Template Initialization

As opposed to most existing methods that require prior knowledge of initial pose, our method can handle large pose variations. In general, our tracker only requires knowledge of the rough global orientation of the subject, for example, whether the subject is facing towards the camera or to the left, etc. The local configuration of each limb can be effectively derived using our tracking algorithm in most cases. Please see our supplemental materials for examples. For pose tracking applications, the limb length estimation process described in Sec. 6.4.1 is performed using the first F frames ($F = 5$ in our experiments), as it requires repeated pose estimation and is relatively more time-consuming. Notice that the algorithm favors all segments of the articulated object being visible, as the scales of the invisible parts can only be estimated via the regularization term in Equation 6.17 and might not be accurate. In addition, poses that resolves more joint ambiguities are preferred in this process. For example, an arm bending pose better defines the length of both the upper arm and forearm than a T-pose.

6.5 Experiments

In order to take advantage of the parallelism of the computations in our algorithm, especially the calculation of posteriors, we implement our pipeline on GPU with CUDA.

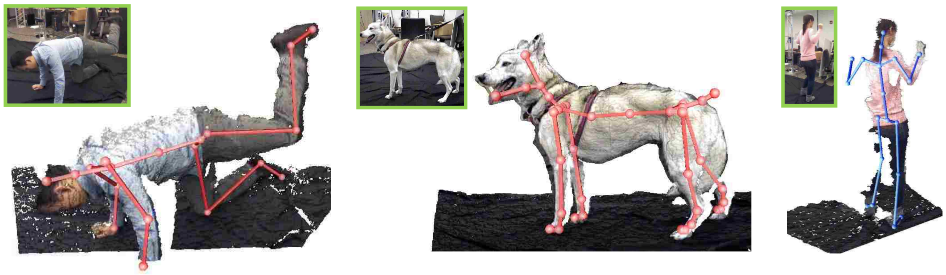
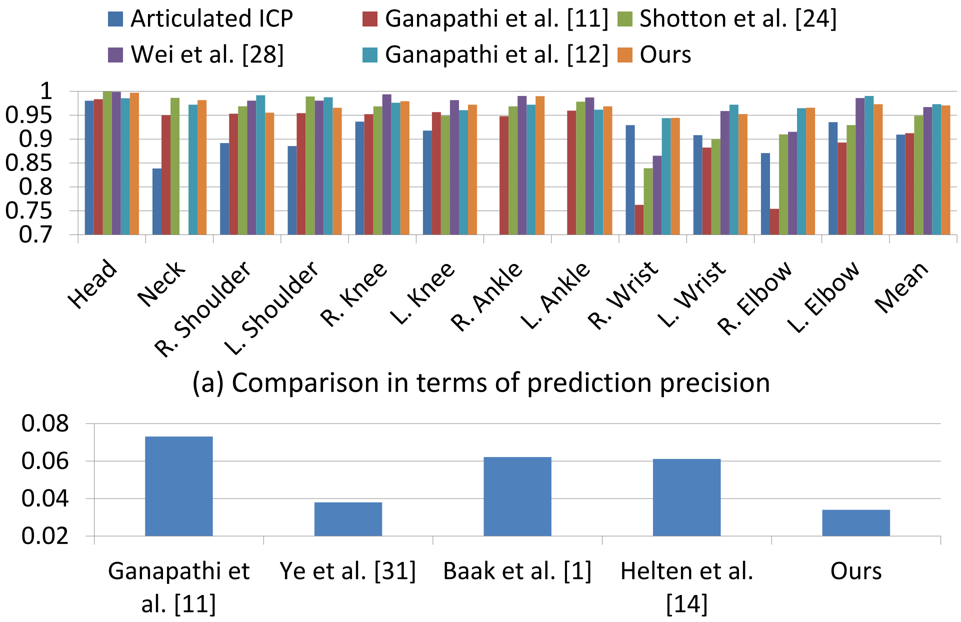


Figure 6.1: Our novel algorithm effectively estimates the pose of articulated objects using one single depth camera, such as human and dogs, even with challenging cases.

With our current un-optimized implementation, each iteration of our pose estimation together with geometric shape adaptation takes about 1.5ms on average, with sub-sampling of around 1000 points for each point set. The running time is measured on a machine with Intel Core i7 3.4GHz CPU and Geforce GTX 560 Ti graphics card. Since our algorithm normally requires < 15 iterations in total to converge during tracking, the entire pipeline runs at real time. In our experiments, we assume segmentation of the target subject is relatively simple using depth information. Therefore, the computational time for segmentation is not considered here.

Parameters: The parameters in our algorithm are empirically set to fixed values throughout our experiments and are listed in Table 6.2. The only exception is the u in Equation 6.1, which denotes the degree of noise and is data dependent. It is set to 0.01 unless significant noises are present. Although in other related methods [35, 89], σ^2 is initialized from the input data directly, we found that such strategy generally largely overestimates the value of σ^2 and will try to collapse the template to fit the input point cloud. Different from shape registration applications, for articulated shapes tracking, it would destroy the temporal information and introduce extra ambiguities. Therefore we use a fixed value instead. Due



(a) Comparison in terms of prediction precision

(b) Comparison in terms of marker distance errors (unit = meter)

Figure 6.2: Quantitative evaluation of our tracker on the SMMC-10 dataset [52] with two error metrics. Notice that in (b), although the method by Ye et al. [133] achieves comparative accuracy, their reported computation time is much higher.

to the multiplier $\frac{1}{\sigma^2}$ in Equation 6.7, which is normally $\geq 10^4$, the regularization terms generally require large global weights.

Table 6.1: The three datasets we use for quantitative evaluations.

	SMMC-10	EVAL	PDT
Subjects	One male	Two males, one female	Three males, one female
Data size	28 Sequences, 100 or 400 frames each (~50% each case)	7 sequences per subject, around 500 frames per sequence	4 sequences per subject, 1000~2000 frames per sequence
Motion complexity	Relatively simple	Moderate to complex (cart-wheels, hand standing, sitting on floor, etc.)	Moderate to complex (jumping, sitting on floor, dancing, etc.)
Ground truth data	Markers	Joints	Joints + Transformations

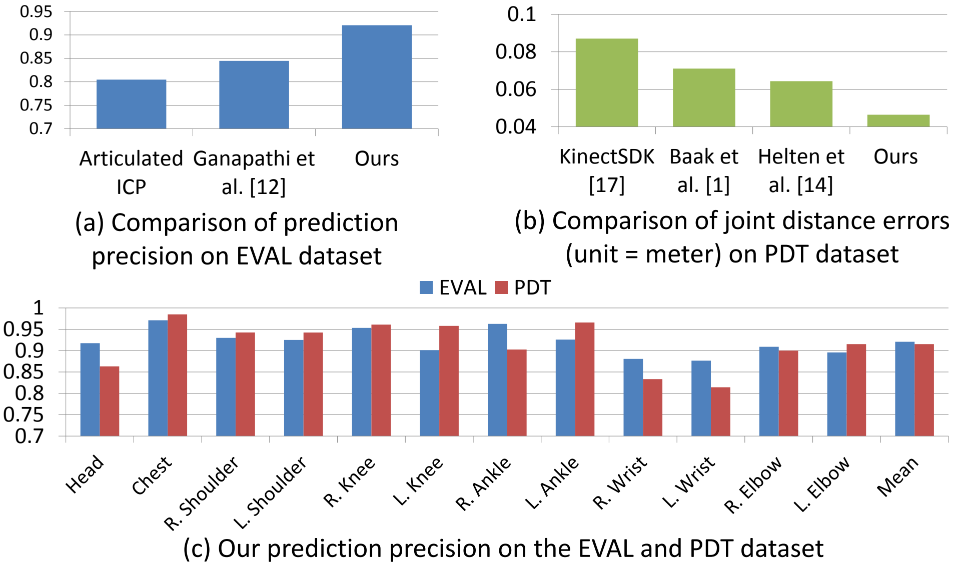
Table 6.2: The parameter settings for our experiments.

	Equation 6.14		Equation 6.17	Equation 6.20	
Initial σ^2	λ_r	λ_p	λ_s	λ_d	λ_e
$0.02^2(m^2)$	1000	500	1000	1	0.1

6.5.1 Tracking Accuracy Evaluations

The accuracy of our algorithm is evaluated on three publicly available datasets, namely SMMC-10 [52], EVAL [53] and PDT [62], that contain ground truth joint (marker) locations. A summary of these datasets is provided in Figure 6.1. Due to the discrepancy of ground truth data format provided and joint definitions across trackers, different methods for accuracy measurement are needed. For the SMMC-10 and EVAL datasets, we use the same strategy as in [133]. Specifically, we align our template to one single frame and mount the corresponding markers to our template. The markers are then transformed with our estimated pose and directly compared with the ground truth. For the PDT dataset, we follow the strategy described in their paper [62]. The estimated joints are transformed via the provided transformations to local coordinate system of each corresponding joint, and the mean of displacement vectors for each joint is subtracted to account for joint definition difference.

Previous methods reported their accuracy according to two different error metrics, directly measurement of the euclidean distance between estimated and ground truth joints, or the percentage of correctly estimated joints (euclidean distance error within $0.1m$). Therefore, we show our results in both ways for comparison purpose. The accuracy of our tracker on the SMMC-10 dataset is shown in Figure 6.2. In this relatively simple dataset, our method outperforms most existing methods, and is comparable with two recent



(a) Comparison of prediction precision on EVAL dataset

(b) Comparison of joint distance errors (unit = meter) on PDT dataset

(c) Our prediction precision on the EVAL and PDT dataset

Figure 6.3: Quantitative evaluations of our tracker's accuracy with comparisons with the state of the arts.

work [53, 128]. The comparisons on the other two more challenging datasets are shown in Figure 6.3. The results clearly show that our method outperforms all the compared methods by a large margin. That means our approaches can handle complex motions more accurately and robustly, while still achieves real-time performance. (The numbers for the compared methods are reproduced from their own papers, except for the Articulated ICP and the KinectSDK of which the numbers are obtained from [53] and [62], respectively.)

As mentioned before, our tracker well handles complex motions. The tracking results of various complex poses from our own data and the publicly available datasets are shown

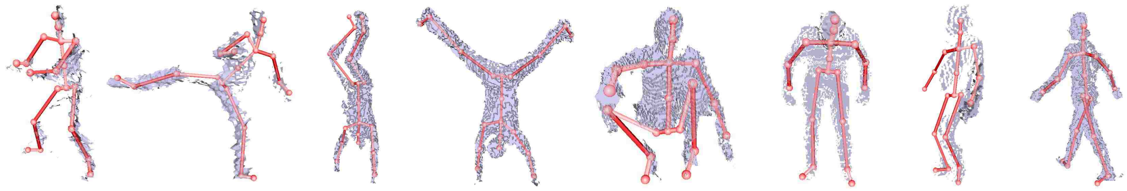


Figure 6.4: Visualization of example tracking results on the publicly available datasets.

in Figure 6.1 and in Figure 6.4 respectively. Furthermore, visual comparisons between our results and the skeletons estimated by KinectSDK are shown in Figure 6.5.

Non-human Subjects Our algorithm can also be applied to articulated objects other than human beings. To demonstrate its applicability, we test on data captured from a dog. A major challenge in this data is the severe self occlusions. Some of our results are show in Figure 6.1 and Figure 6.6. Note that for discriminative and hybrid approaches, a database of this animal will be needed; while we only need a skinning template. Besides, none of existing generative methods have reported results on this type of data, except with multi-view setup that resolves the severe occlusion [51].

6.5.2 Shape Adaptation Accuracy

To quantitatively evaluate the performance of our body shape adaptation algorithm, we compare our results on the PDT dataset [62] with their ground truth shape data. Note that in our pipeline, the body shapes are **automatically** deduced during the tracking process,

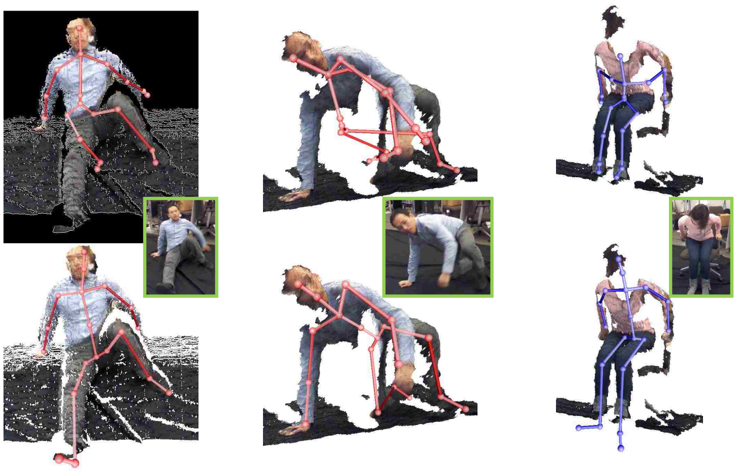


Figure 6.5: Visual comparison of our results (second row) with KinectSDK [4] (first row) on some relatively complex poses.

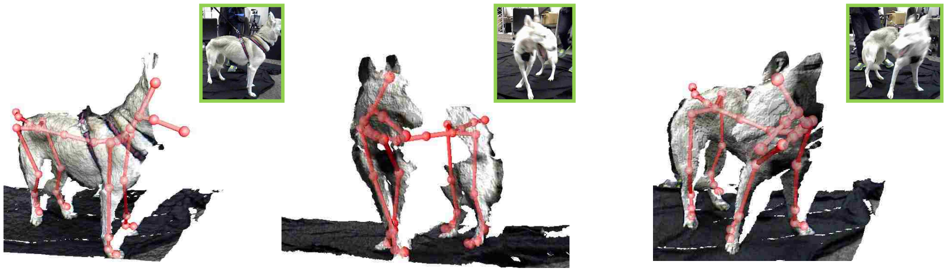


Figure 6.6: Example results of our algorithm applied to dog data.

while a specific calibration procedure is required in [62]. To measure fitting errors, we estimate the pose of the ground truth shape using our algorithm and deform our template accordingly. On average, we achieve comparative accuracy as Heltel et al. [62]: 0.012*m* v.s 0.010*m*. To compare, the mean fitting error reported by Weiss et al. [129] on their own test data is around 0.010*m*. It should be emphasized that [62] requires an extra calibration procedure and [129] assumes small motion between their input, while our method directly operates on the dynamic input. It should also be pointed out that the ground truth shapes from [62] were obtained by fitting SCAPE models to data from scanner. Thus they do not well capture the effects of subject's clothing, which is captured by our algorithm through fitting with the input data. In Figure 6.7, the effectiveness of our adaptation procedure is visualized.

6.5.3 Application in Non-rigid Shape Registration

Since our tracking algorithm (Sec. 6.2) can robustly handle large pose variations, and our template adaptation algorithm (Sec. 6.4) can effectively captures the body shape of the subject, our entire algorithm can be readily used for registering a collection of shapes from a same category of articulated objects. To demonstrate the idea, we apply our algorithm

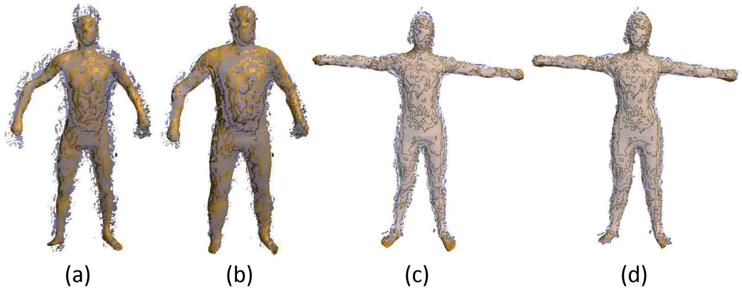


Figure 6.7: Visual results of our shape adaptation algorithm. (a) and (c) are results of only pose estimation (male and female respectively), while (b) and (d) are results with shape adaptation during tracking. Input meshes are overlaid on each result. Notice the adjustment of limb lengths, in particular the arms and feet of the male subject.

to register our template to a set of human scans made available by Hasler et al. [58]. In this process, we iteratively perform our pose estimation, limb length adjustment and surface adaptation components until convergence. Note that since there is no visibility issue, we skip the second step in Alg. 3. Throughout all the tests, we start with the identical template (same shape and limb lengths) in T-pose. Some examples results are shown in Figure 6.8. Despite of the large variety of poses and shapes of the subjects, our algorithm has shown excellent performance. Note that the only requirement for initialization is the rough global orientation of the input data, as our algorithm does not infer any semantic information and cannot automatically resolve such large degree of ambiguity. In the case where the limb lengths differ dramatically, for example $2m$ vs. $1m$, a global scaling will need to be applied first. In general, our algorithm can be used to provide accurate non-rigid registration between shapes, and existing algorithms, such as nonrigid ICP, can be used to further refine the registration. Moreover, the skeleton information in our template can be naturally transferred to the new shapes after registration and automatic rigging can be achieved automatically.

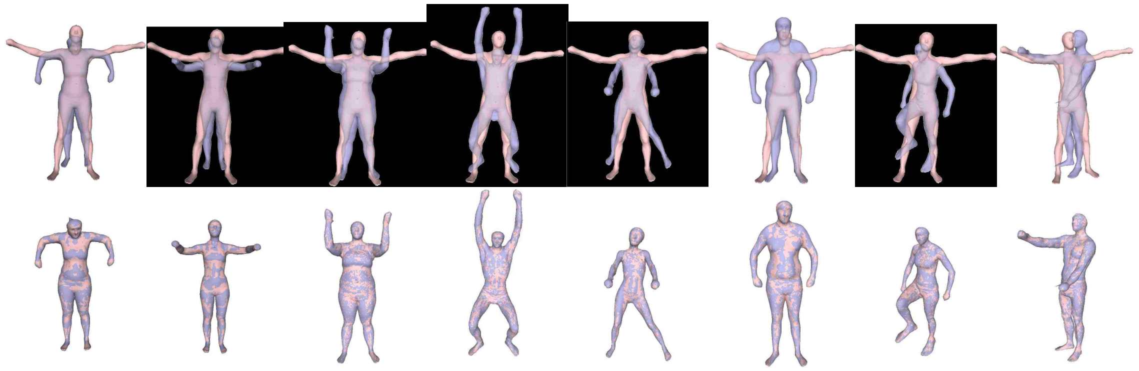


Figure 6.8: Results of applying our algorithm for shape registration. The first row shows the initial state (input) for our algorithm. The second row shows the overlaid shapes after registration. Our algorithm automatically handles substantial shape and pose variations.

6.6 Conclusion and Future Work

In this chapter, we present a novel algorithm for simultaneous pose and shape estimation for articulated objects using one single depth camera. Our pipeline is fully automatic and runs in real time. Through extensive experiments, we have demonstrated the effectiveness of our algorithm, especially in handling complex motions. In particular, we have shown results of tracking an animal, which have not been demonstrated in previous methods with monocular setup.

Our pose tracker could be further improved, for example by taking into account free space constraints as in [53]. Besides, the computational complexity can be greatly reduced through fast Gauss transform during computation of the posteriors [89]. A future direction for exploration is a scheme for adaptive limb lengths estimation along with pose estimation, instead of simply using the first few frames, which usually does not provide complete desired information.

Chapter 7 Dynamic Human Avatar Reconstruction

Our previous methods are only capable of recovering the overall shape of the subject without rich geometric details, which require more sophisticated 3D reconstruction procedure. 3D shape reconstruction is a challenging task that has been extensively studied for decades due to its vast applications. High quality reconstruction can be achieved using surrounding cameras [38, 121]. However, such setup requires careful calibration and is not widely available. By contrast, a monocular setup is highly preferred in various real-life applications. The availability of low cost commodity depth sensors, such as Microsoft Kinect, has made static scene modeling substantially easier than ever [91]. However, the limitation to static scene prevents broader applications in which the scene or the subject may move and deform.

Researchers have been exploring ways to accommodate deformable objects for decades. Successes have been made using color cue to track the motion and then reconstruct shapes [10, 79]. Nonetheless, it is well known that color cue is sensitive to illumination and view changes. Alternatively geometric information, if made available, could be more robust and has been successfully adopted for shape matching and reconstruction. Under the case of small motions, promising results can be achieved using off-the-shell low cost depth sensors [34, 78, 129]. Using high quality input, articulated motions can also be accommodated [28]. However, to the best of our knowledge, no existing method can achieve quality shape reconstruction using dynamic input from a single low cost depth sensor. As the popularity of these sensors rapidly grows, methods that can be used for shape reconstruction

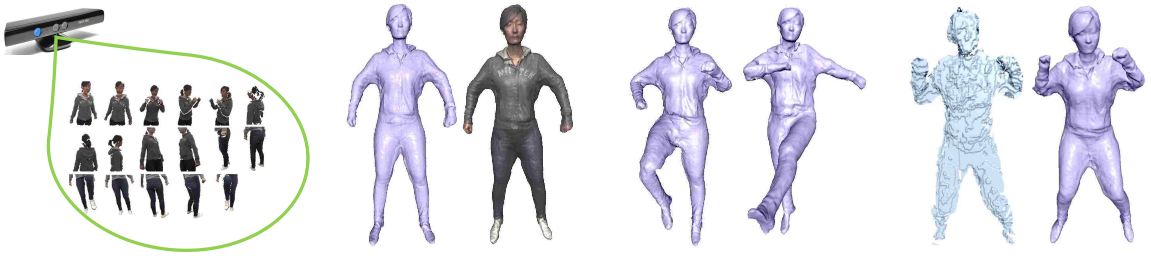


Figure 7.1: Taking a dynamic sequence (moving subject + potentially moving camera) as input , our method automatically selects a sparse subset of frames (left) that contains relatively rich and reliable geometric information and reconstructs a quality watertight model (middle left). The reconstructed model is animatable and can be used for various interesting applications, including pose manipulation (middle right) and space-time resolution enhancement for 3D videos (right).

without posing strict deformation requirements are desired.

In this chapter, we present a system with several novel components that can effectively reconstruct the shape of a moving quasi-articulated entity, such as human, using one single Kinect sensor. We use our probabilistic model-based tracker [134] (Chapter 6) to estimate the poses of the subject. From the tracking information, we develop an automatic frame selection scheme that selects only a sparse subset of key frames with relatively rich and reliable observations. Our system then aligns these key frames to a common reference pose with the assistance of our generic template. The key frames are then globally aligned and combined for shape reconstruction. We choose not to use the entire sequence since registering all of them is both computationally expensive and technically fragile, and typically leading to overly smoothed output. A proxy model is first reconstructed based on the initially aligned key frames and is then used to guide further refinement of key frame alignment. Moreover, to reduce the chance of losing captured details, we develop an adaptive part selection strategy that prunes unreliable observations from each key frame before using them for reconstructing a watertight model. Figure 7.9 shows the outputs of the various

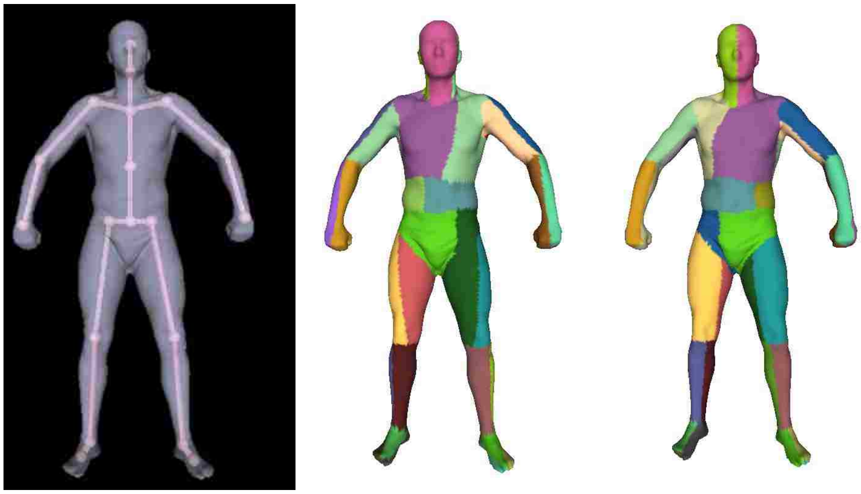


Figure 7.2: A single mesh from the SCAPE database is rigged and skinned and used as the template for our pose estimation. The mesh is segmented into sub-regions for coverage estimation. Each of the right two images shows sub regions without overlap. We use a combination of these two segmentations.

steps of our system. Compared to the current state of the art in single-sensor modeling, our system is much more user-friendly: a user can simply move/turn in front of the camera to create her own posable avatar. Our “scanning-by-part” approach allows the user to move closer to the camera to take advantage of the high-lateral and depth resolutions these depth cameras offer in near range, leading to more details in the final model. Figure 7.1 shows an example of our reconstructed model as well as different applications of this animatable model.

The two major components of our system, namely automatic key frame selection and global alignment are described in Section 7.1 and Section 7.2, respectively. We validate our method through the experiments presented in Section 7.3 and conclude with discussions of some limitations of our method in Section 7.4.

7.1 Confidence-based Frame Selection

Our method starts by selecting a sparse subset of effective frames out of the entire depth sequence. Due to the lack of semantic information from the raw input, we achieve this by leveraging the semantic information from a generic template model. Specifically, we use our probabilistic model-based pose tracker [134] to estimate the pose of the subject. Body part information is then identified, along with reliability measure of the observed surface in each frame, which are later used for key frame selection. For pose tracking, the generic skinning mesh, as shown in Figure 7.2, is used as the template.

There are several desired properties for the set of key frames. First of all, they should provide maximal coverage over the part of subject's body that is observed in the sequence. Secondly, they should capture as rich details as possible. Based on the property of depth sensors, frames captured at near range are favored as a consequence. As our method relies on the output of the pose tracker, the third requirement is high confidence on the estimated pose, i.e. low pose error. Last but not the least, we would like to favor frames with small motions compared to nearby frames in order to avoid large noise caused by motion blur. In the rest of this section, we describe in detail our evaluation of the reliability for each input frame followed by our strategy of frame selection.

7.1.1 Confidence Evaluation

Aimed at quasi-articulated objects, we consider the reliability in a per-part fashion. Due to the occlusion in the monocular data, a single rigid part is almost never completely observed. Therefore, we divide each part of the template into several sub regions and evaluate the reliability for each sub region for each input frame. Small sub regions will provide more accurate evaluation, for example for the degree of details, however, will very likely result in substantially more key frames being selected. We experimentally found that dividing each rigid segment into 6 overlapping sub regions, as shown in Figure 7.2, is a good balance. The sub regions are designed to share large overlaps in order to ensure partial overlaps between frames that are required in the non-rigid alignment described in Section 7.2. In the rest of the paper, we denote each sub region as $\mathcal{R}_k, k \in [1, \cdots, K]$ and the corresponding vertex indices as $\mathcal{I}_k = \{i | \boldsymbol{u}_i \in \mathcal{R}_k\}$, where $\boldsymbol{u}_i$ is a vertex on the template. In the following, we describe the exact formulation of the confidence measure for each sub region in each frame based on the criteria mentioned earlier.

Coverage Confidence

For each frame t, the visibility of each vertex $\boldsymbol{u}_i^t$ on the aligned template is determined. We then use the ratio of visible points for each sub region as the coverage confidence value ($|\cdot|$ denotes the cardinality when used on a set throughout the paper):

$$c_c^t(k) = \frac{\left|\{\boldsymbol{u}_i^t | i \in \mathcal{I}_k, \boldsymbol{u}_i^t \text{ is visible}\}\right|}{\left|\{\boldsymbol{u}_i^t | i \in \mathcal{I}_k\}\right|} \tag{7.1}$$

Geometric Details Confidence

The level of details of a sub region contained in a frame is evaluated through both its distance and motion. The inverse of the average distance of the sub region can serve as a straightforward measure of the score in terms of distance:

$$c_d^t(k) = \frac{1/\sum_i u_{i,z}^t}{\max_t\{1/\sum_i u_{i,z}^t\}}, \quad i \in \mathcal{I}_k \tag{7.2}$$

where the value $u_{i,z}^t$ is the z (depth) component of the vertex $\boldsymbol{u}_i^t$. The average movement of the surface points of the aligned template on that sub region are used to evaluate the speed of motions. The motion-based confidence value is then defined as:

$$c_m^t(k) = 1 - \left|\max_t\{\frac{d_k^t}{p_k}, 1\} - \epsilon_m\right| \tag{7.3}$$

$$\text{where} \quad d_k^t = \frac{1}{|\mathcal{N}(t)||\mathcal{I}_k|}\sum_{t'\in\mathcal{N}(t)}\sum_{i\in\mathcal{I}_k}\|\boldsymbol{u}_i^t - \boldsymbol{u}_i^{t'}\|_2 \tag{7.4}$$

One might initially consider $1 - \frac{d_k^t}{\max_t\{d_k^t\}}$ as the motion-based confidence. However, due to the inaccuracy in pose estimation, the maximum value $\max_t\{d_k^t\}$ might be exaggerated, and the normalized values will be squeezed and contain insufficient discriminative power. Therefore, in Equation 7.3, we cut off the value at the threshold p_k, which is set to be the 90% percentile of the set $\{d_k^t\}_t$ in our experiments. The subtraction of ϵ_m in Equation 7.3 is designed to favor frames with small motions over static frames, that is, values close to ϵ_m after normalization are preferred over 0. The reason for such consideration is to ensure effective key frame upsampling and will be made clear in Section 7.2.2. The value of ϵ_m is set to 0.1 throughout our experiments. The set $\mathcal{N}(t)$ denotes the nearby frames of t. We use previous and next frames except for boundary frames where only the nearby frame from one side is used.

Pose Confidence

We measure the confidence of the estimated pose via the mismatch of the projected region in the image space between the input frame and the aligned template. Since there is no semantic information regarding the body parts from the input frame, we compute a single pose confidence value for each frame, instead of for each sub region or body part. The confidence value is therefore defined as:

$$c_p^t = 1 - \frac{\sum_{i,j} \mathrm{XOR}\big(M_I^t(i,j), M_T^t(i,j)\big)}{\sum_{i,j} M_I^t(i,j) + \sum_{i,j} M_T^t(i,j)} \tag{7.5}$$

Here M_I^t is the binary mask of the input frame and M_T^t is the binary mask of the projected region of the aligned template, while $\mathrm{XOR}(\cdot)$ denotes the exclusive or operation.

Our per sub region confidence $c^t(k)$ is then a normalized product of these four individual confidence values:

$$\begin{aligned} \hat{c}^t(k) &= c_c^t(k) \cdot c_d^t(k) \cdot c_m^t(k) \cdot c_p^t; \\ c^t(k) &= \frac{\hat{c}^t(k)}{\max_t\{\hat{c}^t(k)\}} \end{aligned} \tag{7.6}$$

7.1.2 Frame Selection Strategy

With the per sub region confidence values determined, our method uses a greedy strategy to select a sparse subset of key frames. A naïve approach is to select for each sub region the frame with highest score over the sequence. Unfortunately, this approach will result in a large set of frames being selected. In the worst case, we will get as many key frames as the sub regions.

To solve this issue, our method relaxes the strict requirement of maximum confidence value, and treats equally the frames with confidence over a given threshold τ_c. Our frame selection strategy is formalized in Algorithm 5. During the selection process, the greedy strategy always locates the frame with maximum confidence across all sub regions that are not marked and all frames in the frame pool, which initially contains the entire sequence. The sub region corresponding to this maximum confidence is then marked. Additionally,

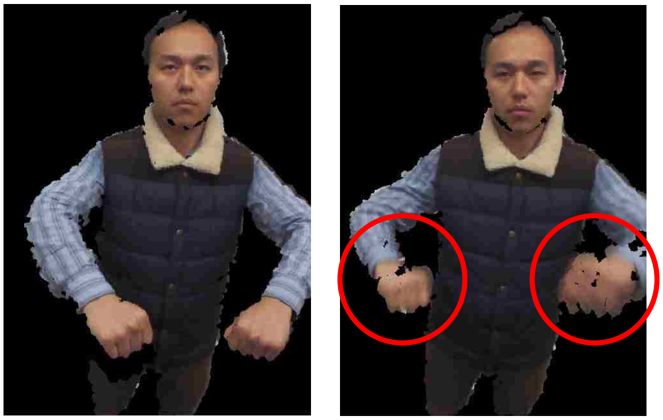


Figure 7.3: Frames with large motion blur (right image) might be selected if $\{c_m^t(k)\}$ in Equation 7.6 is excluded.

```
Data: Input sequence, template model, threshold $\tau_c$
Result: Key frames $\mathcal{F}$
Compute the confidence values $\{c^t(k)|\forall(t,k)\}$ (Equation 7.6).
$\mathcal{F}$ = empty set.
$\mathcal{P}$ = frames indices of the entire sequence, $\{1,2,\cdots,F\}$.
$\mathcal{K} = \{1,2,\cdots,K\}$
while $\mathcal{K}$ not empty do
    Locate $f = \arg\max_{t\in\mathcal{P}}\{c^t(k)|k\in\mathcal{K}\}$;
    Construct the set $\mathcal{J} = \{j|c^f(j) > \tau_c, j\in\mathcal{K}\}$;
    $\mathcal{K} = \mathcal{K}\backslash\mathcal{J}$ (set difference, same below);
    $\mathcal{P} = \mathcal{P}\backslash\{f\}$ ;
    $\mathcal{F} = \mathcal{F}\cup\{f\}$;
end
```

Algorithm 5: The automatic frame selection strategy.

all sub regions in this frame with confidence values above the threshold τ_c are also marked. The frame is removed from the pool. The process stops when all sub regions are marked.

With τ experimentally set to 0.8, this procedure normally selects 10 to 20 key frames, as shown in Figure 7.1 and Figure 7.9. The key frames contain near range scans of different body parts with little motion blur. It should be emphasized that if the motion-based confidence measures $\{c_m^t(k)\}$ are excluded in Equation 7.6, closely captured frames with large motion blur might be selected. Figure 7.3 shows an undesired frame being selected when the motion-based confidence measure is not considered for one of our test sequence. By contrast, a frame with comparative quality for the particular body part (the forearms in this case), yet with almost no motion blur, will be chosen instead by our method. Therefore, our motion-based confidence measure is essential for reliable key frame selection.

7.2 Template Guided Key Frame Merging

The second major component of our pipeline is a merge of the selected frames into a consistent, complete model. In most cases, the subject is under very different poses across the key frames as shown in Figure 7.9. Therefore, it is, in general, very difficult to localize reliable color-based features across the key frames to assist the alignment. In addition, the severe missing data, due to both self-occlusions and limited camera field-of-view, make it even more challenging to identify reliable 3D geometric features. Therefore, methods that rely on feature matching across frames, e.g. [10, 28, 79], cannot be directly used in this case. Moreover, pair-wise alignment is also not preferred due to two reasons. First of all, the number of non-rigid pair-wise alignment will be the square of the number of key frames, resulting in a considerable time-consuming process. Secondly and more importantly, the overlap between key frames might not be consistent over the entire body. Consequently, the outcome of the alignment in areas with little or no overlap will be unpredictable.

To resolve the issue, we rely on the aligned template to guide the key frame alignment. Similar to many other methods, such as [28, 78], we achieve this in two stages, namely initial and global alignment. During the first stage (Section 7.2.3), we deform each frame to a reference pose according to the aligned template, and perform refinement by rigid alignment of discriminative parts. The second stage aims at merging the roughly aligned meshes. Due to the potentially small overlap across different key frames, we further divide this stage into two steps. Firstly, a proxy model is created by fusing the roughly aligned meshes from the first stage via Poisson Reconstruction [71]. All selected frames are aligned towards the proxy model, while the proxy model is actively updated after each alignment. The second step is a global non-rigid alignment as used in [78].

The quality of the merged complete model directly relies on the quality of each key frames. Even though our key frames are selected in a way that favors rich details, the noise of commodity depth sensors makes their quality substantially lower than desired. Following the idea of KinectFusion [91], we fuse multiple nearby frames together to obtain an upsampled scan for each key frames. However, in our case, we should perform non-rigid alignment, rather than rigid alignment. Since our non-rigid alignment technique is based on the Embedded Deformation Model (EDM) [77, 113], we will first briefly describe the formulation of the EDM as well as our extension in Section 7.2.1 and then present the details of our alignment framework in the remainder of this section.

7.2.1 Embedded Deformation Model

The Embedded Deformation Model uses a simplified version of a mesh as a control mesh (deformation graph) to deform the original one. The underlying assumption is that a local small region is close to a plane, and therefore its motion can be represented via only a few nodes. Denoting the set of nodes in the deformation graph as $\mathcal{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, each vertex $\boldsymbol{v}_i \in \mathcal{V}$ of the original mesh is deformed by its L controlling nodes as follows:

$$\Phi(\boldsymbol{v}_i) = R_g\Big(\sum_{j=1}^{L} w_{ij}[A_j(\boldsymbol{v}_i - \boldsymbol{x}_j) + \boldsymbol{x}_i + \boldsymbol{b}_j]\Big) + \boldsymbol{t}_g \tag{7.7}$$

where $[R_g\ \boldsymbol{t}_g]$ is the global rigid transformation, and $[A_j\ \boldsymbol{b}_j]$ is the local affine transformation of node $\boldsymbol{x}_j$. The weight w_{ij} defines the influence of node $\boldsymbol{x}_j$ on vertex $\boldsymbol{v}_i$, subject to $\sum_j w_{ij} = 1, \forall i$.

To deform a mesh to align with a target mesh, a weighted sum of the following three energy terms are iteratively minimized to estimate the deformation parameters $\{R_g, \boldsymbol{t}_g, A_j, \boldsymbol{b}_j\}$:

$$E_f = \sum_{i=1}^{M} \omega_i^2 \Big(\|\Phi(\boldsymbol{v}_i) - \boldsymbol{c}_i\|_2^2 + \rho|\boldsymbol{n}_i^T(\Phi(\boldsymbol{v}_i) - \boldsymbol{c}_i)|^2\Big); \tag{7.8}$$

$$E_r = \sum_{i=1}^{N} \|A_i^T A_i - I\|_F^2 \tag{7.9}$$

$$E_s = \sum_{(i,j)\in\mathcal{E}} \mu_{i,j}\|A_i(\boldsymbol{x}_j - \boldsymbol{x}_i) + \boldsymbol{x}_i + \boldsymbol{b}_i - (\boldsymbol{x}_j + \boldsymbol{b}_j)\|_2^2 \tag{7.10}$$

Here $\{\boldsymbol{c}_i, \boldsymbol{n}_i\}$ is the closest point and its normal on the target mesh with respect to $\Phi(\boldsymbol{v}_i)$ ($\boldsymbol{v}_i$ deformed with the most up-to-date deformation parameters). The weight ω_i controls the confidence of the correspondence and ρ leverages the relatively importance between the point-point metric and the point-plane metric. The first term drives the mesh toward its target. The second term minimizes the non-rigidity of the local transformation and prevents arbitrary surface distortion (I is the identity matrix). The last term serves as the smoothness term and assures the similarity of the local transformations between connected nodes weighted by $\{\mu_{i,j}\}$ ($\mathcal{E}$ is the set of edges in the deformation graph).

We extend the original EDM in two ways, in order to better accommodate the articulated motions of the object. First of all, we utilize the skinning information encoded in our template to adapt the weights $\mu_{i,j}$ used in Equation 7.10. The basic idea is to enforce

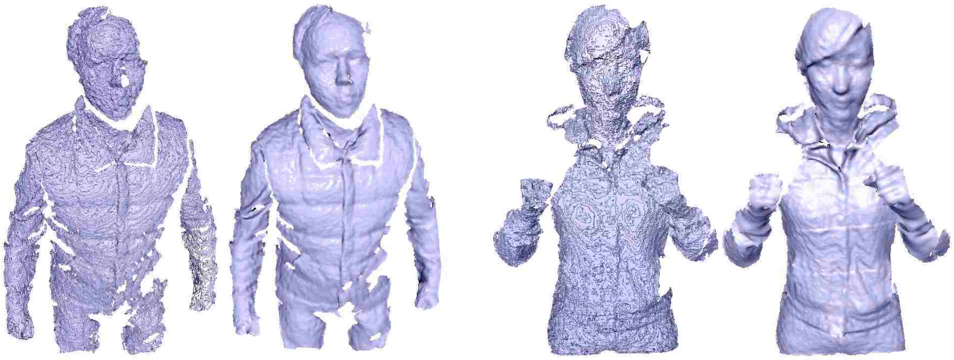


Figure 7.4: The original input meshes and our upsampled meshes. The upsampled meshes are smoother, contain more reliable geometric information, have small holes filled both through fusion with nearby frames.

relatively small smoothness constraint at regions with potential large deformation. Therefore, the deformation model will allow relatively large flexibility at non-rigid region, while enforcing strong smoothness in almost rigid part. Specifically, we set

$$\mu_{i,j} = \frac{\hat{\mu}_{i,j}}{\max_{(i,j)}\{\hat{\mu}_{i,j}\}}; \quad \text{where } \hat{\mu}_{i,j} = \frac{1}{||\boldsymbol{s}_i - \boldsymbol{s}_j||_2 + \epsilon_s} \tag{7.11}$$

The vector $\boldsymbol{s}_i$ contains the skinning weights of the vertex on the aligned template that is closest to the point $\boldsymbol{v}_i$ in the input mesh. The parameter ϵ_s controls the upper bound of the unnormalized weights and is set to 0.5.

Our second extension comes in the form of an extra energy term, aiming at preserving the edge length of the deformation graph:

$$E_e = \sum_{(i,j)\in\mathcal{E}} \left(||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)||_2^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2\right)^2 \tag{7.12}$$

We found this extra term useful in preventing collapsing in the case of small overlap. Therefore, our extended energy function for EDM is as follows:

$$E = \lambda_f E_f + \lambda_r E_r + \lambda_s E_s + \lambda_e E_e \tag{7.13}$$

where $\lambda_f = 10, \lambda_r = 1000, \lambda_s = 100$ and $\lambda_e = 1000$ are used throughout our experiments. The deformation parameters are optimized via the Levenberg-Marquardt algorithm. Due to the large number of vertices in the input mesh, we normally sample a subset to use in Equation 7.8. In our implementation, we use the mesh simplification technique in [54] to generate both this subset and the deformation graph.

7.2.2 Key Frame Upsampling

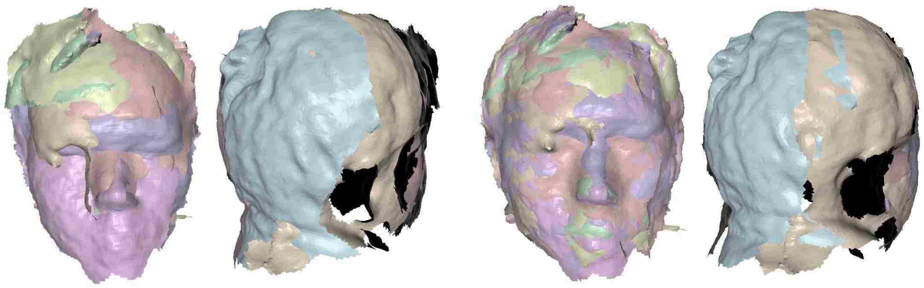
The purpose of the upsampling is to improve the quality of the selected key frames. We achieve this by fusing information from nearby frames. Specifically, for each key frame, we align several nearby frames (6 in our experiments) towards it via Embedded Deformation Model and then perform volumetric fusion as in [91]. Our motion-based confidence defined in Section 7.1.1 ensures small motions around key frames and consequently large overlaps for EDM to align well. In our motion-based confidence definition, one of our criteria was to favor frames with slight motion over no motion. Because when the scene is almost static with respect to the camera, the range maps captured using Microsoft Kinect sensor provide almost identical measurements. In this case, the key frames cannot be effectively upsampled. Therefore, in general we favor frames with slight motions that would provide extra measurement information. Figure 7.4 demonstrates the effectiveness of this upsampling step. The upsampled key frames serve as the input for the later stages of our pipeline.

7.2.3 Template Guided Initial Alignment

During our initial alignment, all key frames are deformed to a reference pose that could be pre-defined or an estimated pose from any captured input. For each vertex on the aligned template, a rigid transformation can be calculated that transforms it to the target position in the reference pose through Linear Skinning Blending (LSB). A naïve approach is to deform the input mesh using both the skinning weights and the rigid transformation of its closest point on the aligned template. However, as shown in Figure 7.6, this simple approach could result in substantial artifacts.

Alternatively, we refine the target vertex positions determined by LBS and rely on EDM to refine the deformation parameters with a proper initialization. This is achieved in three steps. First of all, we refine the mesh deformed using LBS through globally alignment of discriminative body parts. The idea is to globally align certain rigid sub parts across all frames to improve the consistency using the technique proposed in [98]. The body part in the key frame is identified by simply transferring the semantic part information from the aligned template using nearest neighbor strategy. However, not all sub parts are sufficiently discriminative for reliable global alignment. For human subject, we perform the global alignment only on the head and the torso. A typical example of this refinement is shown in Figure 7.5. Afterwards, we deform the mesh from LBS through Laplacian warping [108] using the vertices on the discriminative parts as control points.

In the second step, we use the rigid body transformation of the closest point of each node ($\boldsymbol{x}_i$) of the deformation graph to initialize the EDM deformation parameters ($\{R_g, \boldsymbol{t}_g, A_i, \boldsymbol{b}_i\}$)



(a) Before Global Part Alignment

(b) After Global Part Alignment

Figure 7.5: A typical example of the improvement through the global alignment of discriminative body parts.

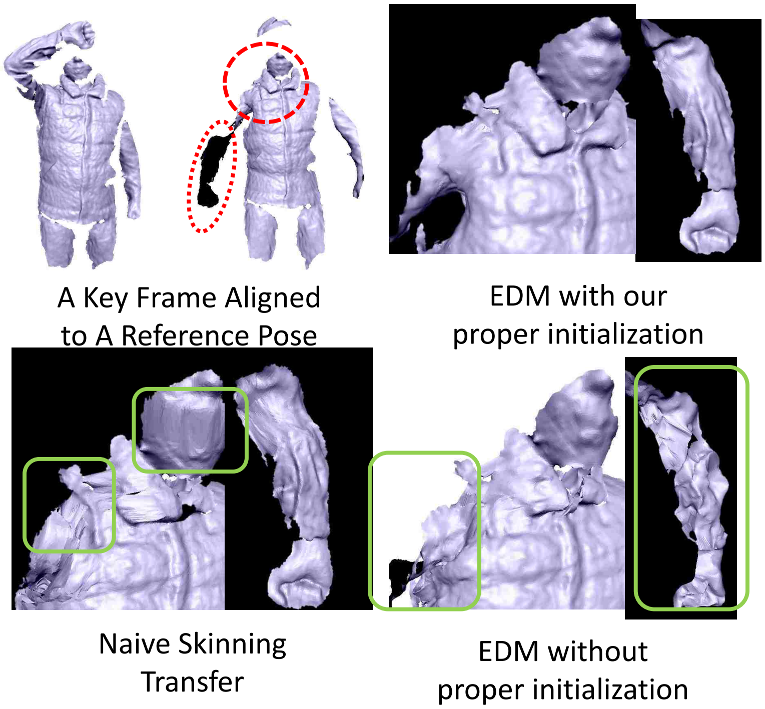
according to the following relationship:

$$\begin{cases} [R_g\ \boldsymbol{t}_g] = [R_g^s\ \boldsymbol{t}_g^s] \\ R_g^s A_i = R_i^s \\ \Phi(\boldsymbol{x}_i) = R_i^s \boldsymbol{x}_i + \boldsymbol{t}_i^s \end{cases} \Rightarrow \begin{cases} [R_g\ t_g] = [R_g^s\ \boldsymbol{t}_g^s] \\ A_i = (R_g^s)^T R_i^s \\ \boldsymbol{b}_i = (A_i - I_3)\boldsymbol{x}_i + \boldsymbol{t}_i^s \end{cases} \tag{7.14}$$

Here $\{R_i^s, \boldsymbol{t}_i^s\}$ are the body space transformation of the closest point and $\{R_g^s, \boldsymbol{t}_g^s\}$ are the global transformation of the template, both from the key frame pose to the reference pose.

During the third step, we use the target positions from first step as the fixed correspondences (the $\{\boldsymbol{c}_i\}$ in Equation 7.8) and use EMD to estimate the deformation parameters. Since not all of these correspondence are reliable due to noises, we further prune them by removing those on edges with length changes or on faces with area changes over a certain percentage.

An example of a key frames deformed using various strategies is shown in Figure 7.6. Our result suffers from substantially less artifacts compared to the one using naïve approach above or the one without proper parameter initialization. This example demonstrates that a proper initialization of the deformation parameters are necessary, due to the potentially large pose difference between the key frames and the reference pose.


A Key Frame Aligned to A Reference Pose

EDM with our proper initialization

Naive Skinning Transfer

EDM without proper initialization

Figure 7.6: An upsampled key frame aligned to the reference pose with our approach is shown on the top left. Close-up views of the two circled regions for various alignment approaches are shown to demonstrate the effectiveness of our template guided initial alignment. Some problematic regions produced by the alternative approaches are highlighted by green rectangles.

7.2.4 Global Non-rigid Alignment

A wildly used strategy for global alignment, both rigid and non-rigid, starts with identifying correspondences between pairs with potential overlap and then simultaneously estimate the deformation parameters for all inputs based on the correspondences [28, 78]. Ideally, we would construct the correspondences from the output of our initial alignment. Nonetheless, the inaccuracy in the initial alignment makes such correspondences unreliable.

To resolve this issue, we propose to first construct a proxy model from the initially aligned key frames, and then perform non-rigid alignment for each key frame towards this proxy model to better identify correspondences. Specifically, we use Poisson Reconstruc-

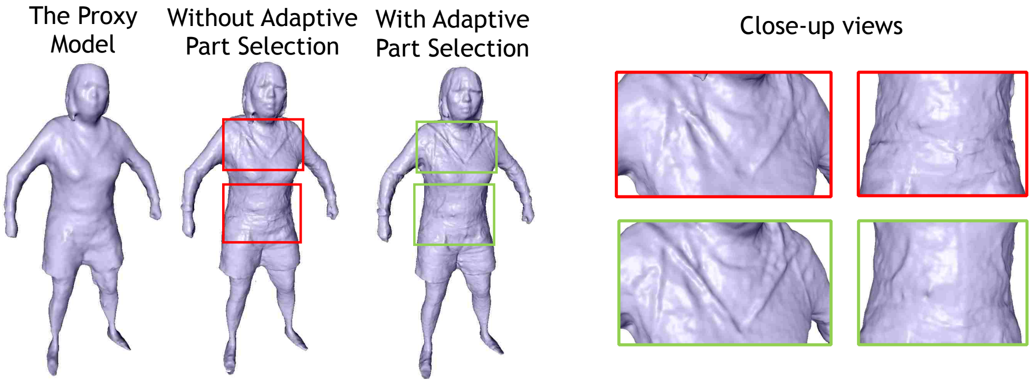


Figure 7.7: An example of our proxy model and two models reconstructed with and without our adaptive part selection scheme. The over-smoothed proxy model is only used to assist our global alignment. The close-up views of the two reconstructed models show that some geometric details are corrupted when all key frames are used for reconstruction. By contrast, the adaptive part selection scheme reduces the chance of loosing details.

tion [71] to build a water tight model using all the initially aligned key frames. Due to some misalignment, the reconstructed model might have bumpy surfaces. Therefore, we perform strong smoothing on the reconstructed model to obtain our proxy model. An example of our proxy model is shown in Figure 7.7. Afterwards, all key frames are aligned towards the proxy model using EDM, taking the deformation parameters estimated in Section 7.2.3 as the starting point.

However, cares need to be taken during the process. If all key frames are independently aligned towards the proxy model, the geometric information from an aligned key frame will not affect the alignment of the others and we might achieve little to no improvement. Instead, frames with overlap should maintain consistent alignment towards the proxy model. We tackle this by determining an alignment order in which two consecutive key frames that will be aligned to the proxy model have minimum non-overlap, and updating the proxy model with aligned key frames in an incremental way. Here the *minimum non-overlap* criteria is used instead of *maximum overlap* because a frame with maximum overlap with aligned key frames might also have parts with no overlap, of which the alignment will be unpredictable.

The determination of alignment order again works in a greedy-like fashion as formalized in Algorithm 6, based on the confidence measure defined in Section 7.1.1. The first frame in $\mathcal{F}$ (see Algorithm 5), denoted as f_0, is selected as the first frame. Then the frame with minimum sum of confidence values for the sub regions that are **not** covered up to this point is chosen as the next frame, until all frames are selected. In our experiments, we set the parameter $\epsilon_\alpha = 0.3$, which can be interpreted as the lower bound for the confidence value of a sub region that will be considered as well covered. To update the proxy model after a key frame is aligned, for each vertex $\boldsymbol{v}_i$ on the proxy model, the intersection along its normal with the aligned key frame is located, denoted as $\boldsymbol{a}_i$. Then a weighted average is calculated as $\boldsymbol{v}'_i = (w_i^p \boldsymbol{v}_i + w_i^a \boldsymbol{a}_i)/(w_i^p + w_i^a)$, and the weights are updated as $w_i^p = w_i^p + w_i^a$. For simplicity, we set $w_i^p = 0$ initially and $w_i^a = 1$ for all key frame vertices. One could further explore uncertainty in the key frame vertices to set the weights, for example as in VRIP [36].

After all key frames are aligned to the proxy model, we identify correspondences between each pair of key frames. We then use the approach proposed in [78] extended by incorporating our edge length energy term E_e in Equation 7.12 to simultaneously deform and align all key frames. The edges are constructed by searching for a certain number (4 in our implementation) of nearest neighbors for each selected point of a key frame in the correspondence pool and pruning pairs with closest points on the template lying on different body parts.

```
Data: $\{c_t(k)\}$ and $\mathcal{F}$ (see Algorithm 5) and a parameter $\epsilon_\alpha$.
Result: Ordered key frames $\hat{\mathcal{F}}$.
Initialize $\hat{\mathcal{F}} = \{f_0\}; \mathcal{F} = \mathcal{F} \backslash \{f_0\}; r_k = c^{f_0}(k) \geq \epsilon_\alpha, \forall k \in [1, \cdots, K]$.
while $\mathcal{F}$ not empty do
    for $\forall t \in \mathcal{F}$ do
        $\alpha^t = \sum_{k=1}^{K} c^t(k)(1 - \delta(r_k))$, where $\delta(\cdot)$ is an indicator function.
    end
    Locate $f = \arg\min_{t \in \mathcal{F}} \{\alpha^t\}$;
    $\mathcal{F} = \mathcal{F} \backslash \{f\}$;    $\hat{\mathcal{F}} = \hat{\mathcal{F}} \cup \{f\}$ ;
    $r_k = 1$,    where $c^f(k) \geq \epsilon_\alpha, \forall k \in [1, \cdots, K]$;
end
```

Algorithm 6: Our greedy strategy to determine the order of aligning the key frames to the proxy model.

At this stage, the final step of building a water-tight model might be performed using the aligned key frames. However, taking all of them into consideration for all body parts might result in certain loss of details. The reason is that a frame with relatively high quality scan on one body part might, and in fact usually will, suffer from low quality for some other body parts. Therefore we propose an adaptive scheme, again based on the confidence values, to select the confident body parts of each key frame for the final model reconstruction. Specifically, for each sub region defined earlier, we select two frames with highest confidence values and then keep the **entire body part** corresponding to the sub region for both frames based on the body part information transferred earlier from the aligned template. Now each key frame might have a sub set (typically not all) of the scanned surface that has relative high fidelity. After our adaptive selection step, we apply Poisson Reconstruction [71] to build our final water-tight model. The comparison between results with and without our adaptive selection is shown in Figure 7.7. Even though the improvement might not be significant, this adaptive scheme not only introduces little overhead when determining the confident sub parts, but also reduce the amount of data used for final reconstruction

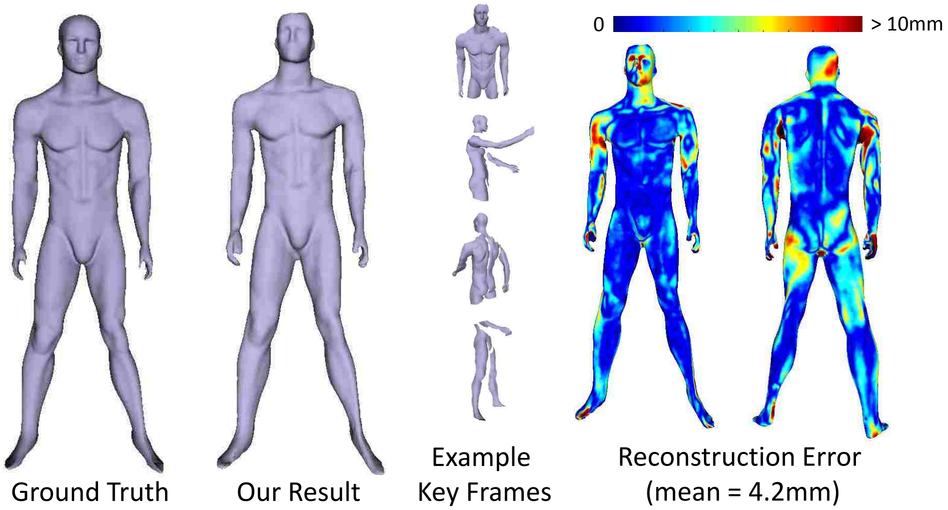


Figure 7.8: Quantitative evaluation of our method on a mannequin under different poses.

and consequently the computational time.

7.3 Results

To validate our approach, we conduct both quantitative and qualitative evaluations, on a mannequin and several subjects with various body shapes and apparels, respectively.

7.3.1 Quantitative Evaluations

We use the mannequin shown in Figure 7.8 to measure our reconstruction error. Different from [78] whose mannequin is static, we articulate the arms of our mannequin in different positions to synthesize various subject poses. We then capture a set of depth maps for each pose with slight view changes and perform our key frame upsampling to obtain a relatively high quality input. Totally 12 key frame image sets are captured, among which four of the upsampled key frames are shown in Figure 7.8. We then compare our reconstructed output with a model acquired using a structured light scanner with 1*mm* accuracy [74]. To align these two models, we estimate the body pose of the ground truth model and use it as the reference pose. Both the ground truth and our reconstructed model, as well as the error plot, are shown in Figure 7.8. On average, the error is less than 5*mm*. Notice that one area with major errors is the head, which is in fact partially due to the slight head pose difference between the ground truth and the reconstructed model caused by our pose estimation inaccuracy. Though our accuracy is slightly lower than that of [78], it should be emphasized that we consider pose variations while their quantitative evaluation is based on completely static model.

7.3.2 Human Subject Examples

For our experiments on real data, the subjects are asked to move in front of the camera performing various motions. We consider both static camera and moving camera scenarios. When the camera is static, we normally split the data acquisition into two phases by positioning the camera on the top and on the bottom to capture different regions of the subject's body in close range. However, this is not a requirement as our frame selection scheme will automatically select the optimal subset of frames regardless of the camera position. In the case of moving camera, the camera continuously changes the location and orientation while the subject is also moving, for example turning himself/herself. (The input videos of these subject data are provided in our supplemental video.)

Figure 7.9 shows our results on four different subject data with various body sizes, apparels and motion complexities. The first three are captured with a static camera and the last one is captured with a moving camera. The first row demonstrates that our key frame selection scheme successfully selects a sparse subset of frames that provide close-range measurement of the majority of the body. With our template guided initial alignment, all the key frames are roughly aligned in the reference pose as presented in the second row. We use a default pose as reference for all these experiments, though it could be rather arbitrary. The results after our global alignment in the third row clearly demonstrate the improvement over the initial state. The reconstructed models presented in the fourth row successfully preserve the surface details with only small artifacts.

We use a simple strategy to colorize the model. For each point on the model, intersections along its normal direction with all the globally aligned key frames are calculated. The colors of the intersection points are obtained via linear interpolation on the corresponding triangle. The median among the colors of the intersection points is assigned to this model point. For the model points with no intersection with any key frames, the colors are hallucinated through Poisson Blending. The colorized models in the last row of Figure 7.9 shows that this simple strategy produces visually pleasing results. Small artifacts might be observed when there are dramatic illumination changes for some body part, for example the back of the forearms of the third model is slightly brighter than other areas. A more sophisticated technique [19] can be used to achieve more consistent texture reconstruction [78].

It should be pointed out that for the second subject, the back of both arms are not observed during the entire sequence due to self occlusion. In fact, such phenomenon is not rare when the subject attempts to remain in the same pose during the scanning. Although the unobserved areas still can be reconstructed, as is also shown in our results, the hallucination normally produces only a smooth surface. Therefore, it further supports the desire of allowing the subjects to move during the scanning as well as the advantage of our method

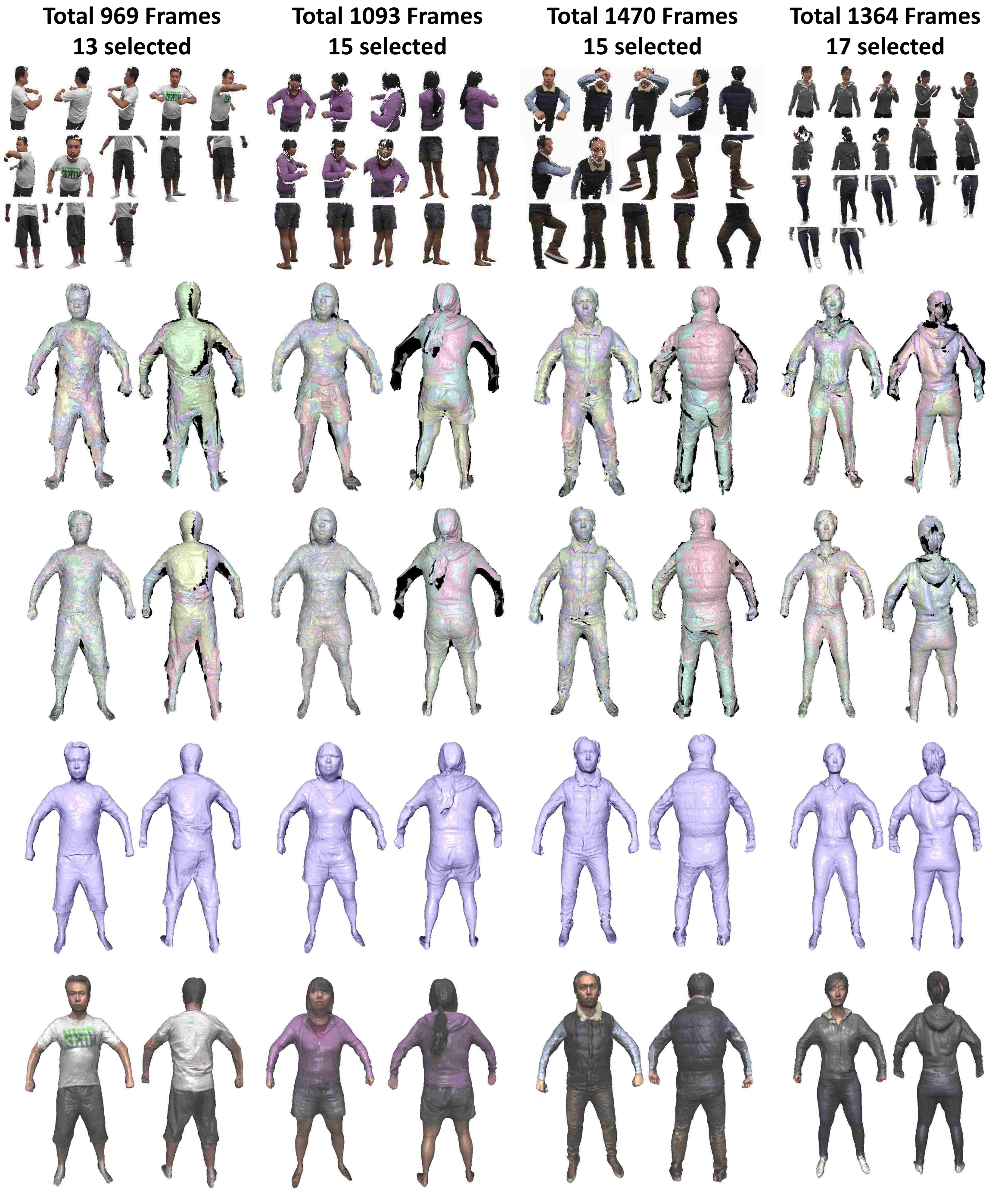


Figure 7.9: Our results on four different subject data. For each example, the set of automatically selected key frames are shown on the top (Section 7.1). The second row shows the key frames with our template guided initial alignment (Section 7.2.3). The globally aligned key frames are in the third row (Section 7.2.4). The final water-tight models without and with coloring are shown in the last two rows.

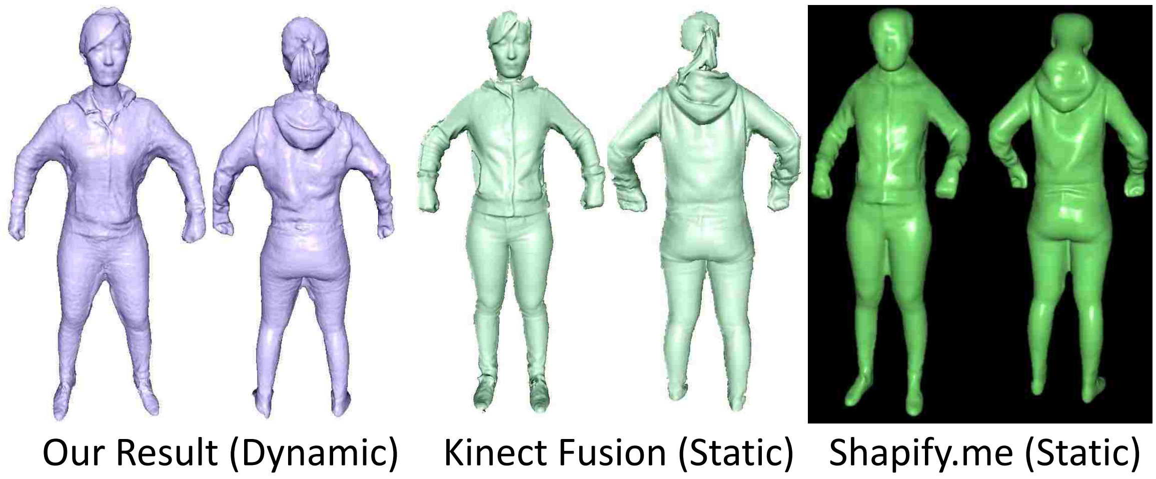


Figure 7.10: A visual comparison of our result dynamic input with models scanned using KinectFusion and Shapify.me with the subject holds static.

in this regard.

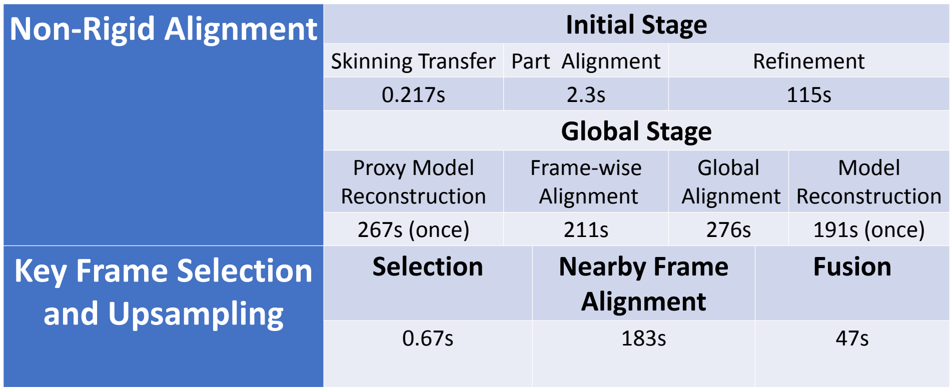
7.3.3 Comparisons

Most existing methods for shape scanning require the subject to remain as static as possible. To the best of our knowledge, no existing methods can effectively reconstruct a high quality complete 3D model from a dynamic depth video. The work by Chang and colleague [28] is one of few approaches that can accommodate articulated motions. However, they require high quality input and can not produce proper reconstruction if directly applied to data from commodity depth cameras, even with super-resolution, as shown by Cui et. al [34]. To shed some light on how our reconstruction quality is compared to existing shape scanning methods, we use two selected methods to scan one of our four subjects when she holds still during the acquisition process.

The two methods we compare, as shown in Figure 7.10, are KinectFusion [91] and Shapify.me [7]. The KinectFusion scan shown in Figure 7.10 only contains the frontal part of the subject, because it is very tedious and difficult to scan a whole person without getting into unacceptable drifting. Shaify.me is an implementation of the system proposed in [78]. During our tests, it provides smooth model, yet less details than ours.

7.3.4 Computational Performance

The majority components of our approach are implemented using Matlab. Figure 7.11 provides a detail breakdown of the computational cost of each major component in our pipeline. The most time consuming part is various non-rigid alignment steps using EDM as expected. With 15 key frames, it takes about three to four hours. In our experiments, our deformation graphs normally have around 1500 nodes, each with 12 deformation parameters. Therefore, we are optimizing for a total of over 270K parameters during the global non-rigid alignment. Besides, consider the amount of motion in our input, we believe the time performance is acceptable. Moreover, the implementation can be further optimized with C/C++ or even CUDA for speedup.



Non-Rigid Alignment	**Initial Stage**			
	Skinning Transfer	Part Alignment	Refinement	
	0.217s	2.3s	115s	
	Global Stage			
	Proxy Model Reconstruction	Frame-wise Alignment	Global Alignment	Model Reconstruction
	267s (once)	211s	276s	191s (once)
Key Frame Selection and Upsampling	**Selection**	**Nearby Frame Alignment**	**Fusion**	
	0.67s	183s	47s	

Figure 7.11: The computational time is calculated over our four subject data and provided here as per frame measurement, except those marked as *once*.

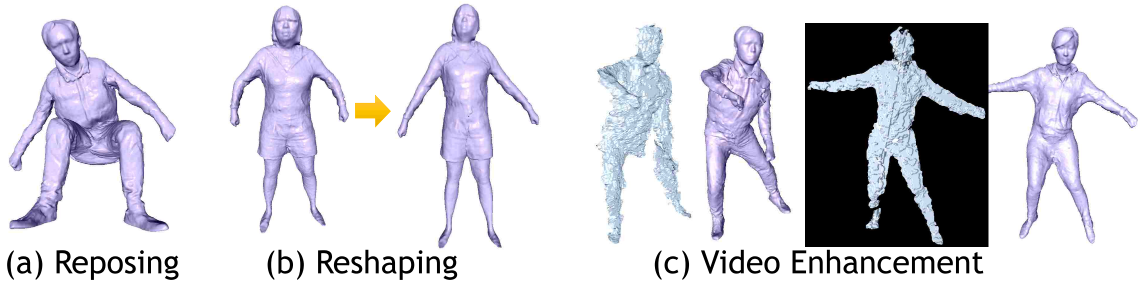


Figure 7.12: Several examples of adapting the reconstructed models for various tasks. (a) The model is skinned into a novel pose. (b) The model is made higher and slimmer. (c) The model is used for space-time resolution enhancement of 3D video.

7.3.5 Post Reconstruction Applications

Since our method utilizes a skinned template, our reconstructed models can be naturally turned into an animatable model. Specifically the skeleton of our template is already embedded inside the models. Therefore, the models can be automatically skinned by transferring and smoothing skinning weights from our template model without requiring extra automatic rigging techniques that will be needed for some other methods such as [78]. Various tasks can be achieved from this point and a few examples are demonstrated in Figure 7.12.

First of all, the models can be easily re-posed into any novel pose by skinning as the example shown in Figure 7.12(a). Secondly, the model can be easily reshaped by scaling the skeleton and then the surface points accordingly. A more interesting application is to utilize the reconstructed model for space time resolution enhancement of 3D videos. There are various ways to achieve this task. One can directly perform mesh registration to align the model toward each input frame. Alternatively, we estimate the pose of the subject in the low resolution video, and then use a variant of the technique described in Section 7.2.3

to deform the mesh. Specifically, we build a deformation graph for the model and use the tracked pose to initialize the deformation parameters. Then we perform refinement by directly aligning the model towards the observed input via EDM. It should be emphasized that the proper initialization, which is enabled by the associated skeleton, reduces the computational time as well as the risk of drifting. Two example frames are shown in Figure 7.12(c) and the improvement in quality is evidently significant. (The entire video is provided in our supplemental video.)

7.4 Limitations and Conclusion

Our system currently uses a generic template and a depth-based tracker to partially overcome the challenges raised due to motions and non-rigid deformation. It has its pros and cons. On one hand, by using a depth-based pose tracker, our method allows the subject to move relatively freely during the acquisition process. On the other hand, sufficient observations on the subjects body are required in order to perform reliable tracking, which may require more frames than methods such as [78, 91]. Tracking is also proven to be very challenging for the low-quality data we are dealing with. One artifact in some of our models is the shrink of volume on body extremities. This is caused by the partial penetration between the template and the key frames. The union of these under-template frames results in a shrank shape. To improve tracking accuracy, robust statistics and the adaptation of combined RGB-D flow tracking ([64]) may be useful.

Another limitation of our method is the lack of capability to accommodate large deformations caused by the apparel. A closer look on the arms of the third subject in Figure 7.9 shows that the surface is not as smooth an the others, because the shirt wore by the subject undergoes relatively large and irregular non-rigid deformations during the subject motion. Such deformation violates the as rigid as possible locally assumption of EDM and our current system does not incorporate strategies to resolve it. A local adjustment of the aligned meshes might be useful for reducing the artifacts caused by such non-rigid deformations. However, without sophisticated and special purpose models, the true motions of garments are intractable in general.

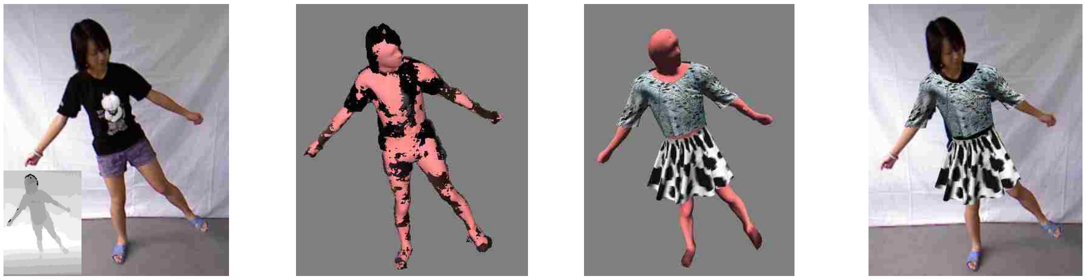
Despite of the limitations mentioned above, our approach can still accomplish the task of dynamic shape reconstruction under various circumstance. It allows the user to move freely in front the camera to easily create a posable 3D avatar with details of clothing etc. This is made possible with our carefully designed frame selection and non-rigid alignment framework. We hope our method will inspire further exploration of other researchers in this direction, for example providing real-time feedback to tell the user which part is yet to be scanned.

Chapter 8 Applications: Virtual Try-On and Smart Health

8.1 Virtual Try-On with a Single Commodity Depth Camera

The concept of *Virtual Try-On*, due to its large commercial potential, has been explored before. The general idea is to track the user's motion, in either 2D or 3D [44, 118, 137], and synthesize clothes that can be overlayed on the user's image. Due to the complexity of human motion and the computational cost of cloth simulation, different systems have different trade-offs. Some treated virtual clothing as textures (e.g., [137]), over-simplifying the interactions between the user and the clothing; some required a pre-made avatar that is either quite crude or difficult to adapt to the user motion and shape (e.g., [44]).

We believe that an ideal virtual try-on system should realistically and efficiently simulate virtual clothing that reacts accurately to the user's body shape and motion. This aspect is especially crucial for users to correctly evaluate the appearance of different clothing on them, thus greatly improving their acceptance to such systems. Unfortunately, none of the existing systems satisfy all these constraints as far as we know. Using data acquired from commodity depth cameras in cloth simulation seems to be a natural solution to this problem. In fact, using depth maps to handle cloth-body collision is a well-studied GPU-based technique as shown in [56, 61, 73]. Unfortunately, the captured depth maps are often noisy and incomplete, with the frame rate limited by the hardware. One possible solution to this problem is to use multiple cameras, but camera synchronization is a challenging problem in the real world as no commodity depth camera currently supports inter-camera



(a) Input RGBD Data (b) Adapted Template (c) Cloth Simulation (d) Virtual Try-On Feedback

Figure 8.1: Our Virtual Try-On system takes the RGBD data in (a) as input and captures the pose and body shape of the user accurately as shown in (b). It then realistically simulates virtual clothing on the user, so the user can examine the appearance of personalized virtual clothing, as illustrated in (c) and (d). Our system robustly handles a wide range of human motions and shape variations.

synchronization.

In this chapter, we present a system that combines our model-based pose and shape estimation algorithm (Chapter 5) with cloth simulation techniques for Virtual Try-On. With our system, users can dress themselves up in different clothes in a virtual environment. The key to our system is our real-time tracking algorithm that delivers meshes that are complete and maintains the same topology over time, which makes it ideal for use in physically-based cloth simulation. As a result, our system is able to provide a more realistic virtual try-on experiences as the example in Fig. 8.1 shows.

The rest of this section is organized as follows. In Sec. 8.1.1, we review some of the state-of-the-art approaches that are related to Virtual Try-on. The approaches used for our cloth simulation and final image composition are discussed in Sec. 8.1.1 and Sec. 8.1.2, respectively. Experimental results are shown in Sec. 8.1.3, while conclusion and a discussion of future work are presented in Sec. 8.1.4.

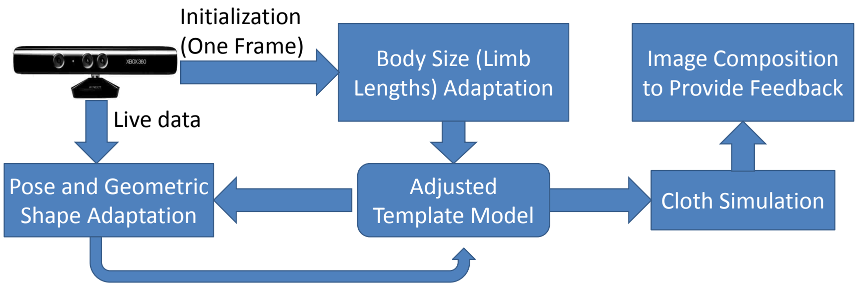

Initialization
(One Frame)
Body Size (Limb Lengths) Adaptation
Image Composition to Provide Feedback
Live data
Pose and Geometric Shape Adaptation
Adjusted Template Model
Cloth Simulation

Figure 8.2: The diagram of our virtual try-on system. The left part is the same as in Figure 5.1 and is provided here for completeness. After performing pose and shape adaptation, the adjusted template is used by our cloth simulator to drive the virtual apparel. Then a composite image of the virtual apparel and the input are delivered to the user to provide virtual try-on experience.

8.1.1 Related Work

Cloth Simulation

Recent research in physically based cloth simulation has resulted in a number of simulation approaches [18,24,32,69] to improve the realism and efficiency of cloth animation. Survey articles [31,66,90] have summarized current state-of-the-arts. Among these methods, our work is particularly related to two types of novel simulation techniques.

Data-Driven Cloth. The first type contains data-driven techniques that combine synthetic or captured data with physically based simulation. Wang and his collaborators [124] developed a clothing wrinkle database and used human poses to guide fine wrinkle synthesis for clothing animation. De Aguiar and his colleagues [37] ignored physical models and proposed a purely data-driven model to efficiently generate wrinkle details instead. Feng and colleagues [42], as well as Kavan and collaborators [70], proposed data-driven models that can enrich coarse cloth simulation with fine details from data. While our data represents a human body rather than cloth, we think those techniques can further extend our work in the future, by combining data with synthetic or captured cloth data.

Cloth-Body Collision. The second type handles cloth-body collision by using synthetic depth maps. Due to their compatibility with the standard graphics pipeline, these techniques [56, 61, 73] can be accelerated by graphics hardware and are often used in GPU-based cloth simulation systems. For more accurate collision detection, 3D distance fields can also be efficiently constructed by GPU acceleration as Sud et al.l [112] and Morvan et al. [86] demonstrated. Since these techniques considered depth maps (or distance fields) as intermediate representations from known body shapes, they could easily construct multiple depth maps from the input mesh model, and thus provide surface normal information to improve collision accuracy.

Virtual Try-On

Compared with the cloth simulation, virtual try-on and personalized 3D garment design is a much less studied problem. Most existing systems treat virtual clothing as static texture patches and use image-based rendering techniques to virtually drive the cloth [59]. Many methods rely on a pre-captured database with subjects in a large variety of poses to find a best match and perform local refinement [41, 60, 115, 137]. While these methods, to a large extend, ignore the interaction between users and the clothes, some pioneered this area by combining real-world data with physically based cloth simulation. There are two main strategies to animate virtual clothing in a virtual try-on system. A straightforward and robust way is to create an avatar that has the same body shape as the user, and then simulate virtual clothing on it. The body size can either be specified by the user input [45, 88, 118], or using depth sensors [111]. While these techniques can accurately model virtual clothing on a static body shape, they cannot easily handle body motions. The triMirror system [118] simulated virtual clothing on a moving avatar, whose motion was controlled by the user's skeleton pose. However, as its result showed, their system seemed to use a pre-defined avatar which did not exactly match the user's body shape. Alternatively, it is preferable to obtain body shape from depth data. One such example is the Fitnect [44] system. While it successfully animated part of the clothing by body motion, the rest still needed to be static. In addition, it only treated clothes as a piece of cloth in front of the user, and it had difficulty in forcing the clothes to follow body motion exactly. Compared to the existing methods, our system can effectively capture the pose and shape of the user, and provide realistic cloth simulation.

Body-Guided Clothing Simulation

After we capture the human motion from a sequence using the technique presented in Chapter 5, we use them to guide clothing animation in physically based cloth simulators. Since physically based cloth simulation is known for its large computational cost, we need to find a good balance between the simulation quality and the computational cost. Our idea is to incorporate two simulators into our system: a realtime cloth simulator for interactive preview purposes and an offline cloth simulator to generate high-quality clothing animation. At the beginning, the user chooses the realtime simulation mode to quickly examine how clothing behaves when it is draped on his/her body. After that, if the user would like to check more clothing details, he/she can switch to the offline simulation mode, under which the system generates either a high-quality clothing shape under a static pose in a few seconds, or the whole high-quality clothing animation in minutes or even hours.

Near Realtime simulation. Our system needs a fast cloth simulator to simulate clothing motion and cloth-body interaction in near real time and preferably in real time. A simple yet effective strategy is to use a coarse mesh to represent each clothing piece. Such low-resolution cloth simulation cannot generate high-quality details such as wrinkles and folds, but its efficiency allows the user to quickly examine the clothing in different poses. In addition, we choose a simple mass spring model to handle both in-plane and bending deformation of cloth. We use Hooke's law to model the spring stiffness and solve the whole simulation using an implicit time integrator.

Compared with dynamic simulation, a more challenging problem is how to handle collisions and friction in real time. To achieve that, we create a set of virtual depth maps from virtual camera views around the body, similar to the other realtime collision techniques proposed in [56, 61, 73]. In particular, one of the virtual cameras should be collocated with our actual camera, and its depth map needs to be in high resolution to enable accurate cloth-body collisions in that view for the user to observe later. We note that this depth map should also be created virtually using the adapted template mesh, rather than using the raw depth map input, which may contain noises and errors. Once the algorithm detects a cloth vertex sufficiently close to the body, it applies a position-based constraint to move it away from the body. For simplicity, we ignore self intersections and consider cloth-body collisions only. In practice, we found that self collisions are rare under low-resolution in many simulation cases.

In the real world, the friction between the clothing and the human body can be highly complex, especially when the body performs dramatic motions, such as stretching or kicking. Our previous experience shows that the use of Coulomb's law and a small set of frictional parameters, as in many other simulators, is often insufficient to produce the desired clothing effects. Specifically, the clothing can either be too "rough", which suppresses its movement over the human body, or too "smooth", which causes dramatic clothing deformation. Our simple solution is to introduce a number of anchor points where the clothing piece is attached to the body. For shirts, the anchors are typically the shoulders; for pants and skirts, the anchors are typically the waist. We implement the anchoring points by setting them as position-based constraints on the clothing, so that the clothing can follow the body properly when the body moves over time.

Offline simulation. To produce highly detailed clothing animation in our offline simulation, we use the implicit Finite Element Method (FEM) to simulate in-plane cloth dynamics and the hinge edge bending model proposed by Bridson and colleagues [25] to animate bending deformation. Our implicit FEM solver is extended from the one developed by Volino and collaborators [122], which takes both nonlinear tensile deformation and nonlinear shearing deformation into consideration. As a result, we can incorporate the real-world material properties from the cloth elasticity database developed by Wang and collaborators [125] into the simulation to produce physically accurate clothing deformation behavior. Similar to the solver proposed by Volino and collaborators [122], our solver is not fully linearized and it is not unconditionally stable. However, compared with explicit solvers (that require the time steps to be approximately 10^{-6}s), it can robustly use orders-of-magnitude larger time steps even when handling highly stiff woven fabrics, which are commonly used to make everyday clothing.

Given a sequence of topologically consistent meshes, the handling of the interaction between the clothing and the human body is a relatively well defined problem. Here we use Continuous Collision Detection (CCD) [24] to detect and remove both cloth-body collisions and self collisions of cloth. We do not need to consider self collisions of the body, which are not supposed to affect the clothing animation result anyway. To speed up collision handling, we implement collision culling by using a regular grid data structure for spatial partitioning. Since the time step used by our simulator is small already, there is no need to use sub steps for collision handling. In practice, we run collision handling every three or four time steps, which is approximately $\frac{1}{900}$s. Similar to the realtime simulator, our offline simulator uses Coulomb's law to model cloth friction and sets a group of anchor points to prevent clothing from sliding a large amount.

8.1.2 Final Image Composition

The core of a virtual try-on system is the capability of providing visual feedbacks to the users. To achieve this, our system combines simulated cloth with the captured image data, and then displays them on a monitor. Correct visibility is the key to producing realistic try-on results. We found that this process can be easily done by using a two-pass rendering method under the basic OpenGL rendering pipeline. To begin with, we set the OpenGL projection matrix to be the same as the projection matrix of the depth sensor. During the first pass, we recover 3D locations of captured pixels according to their depth values, and we draw them as a combination of the human body and the background, using captured images as textures in OpenGL. In the second pass, we simply draw the simulated cloth, and OpenGL takes care of the visibility test using its depth buffer. This process is easy to implement and runs in real time (>100Hz).

8.1.3 Results and Discussions

We tested our system on an Intel Core i5-2500K 3.3GHz CPU with an NVIDIA Tesla C2075 GPGPU processor. Our input data are captured using a single Kinect camera with a resolution of 640×480 and sampling frequency of 30FPS . As our tracker does not require the entire body to be observed, except for the body size adaptation step, there is no specific restriction on the camera position and orientation. However, even though our tracking algorithm can deal with some noise, we do assume the input data are segmented and only the part of the surfaces belonging to the subject is provided to our tracker. Assuming the camera is fixed during data acquisition, we fit a plane to the ground and remove points that are close to the plane by some threshold (20mm for our test data). The background can be removed via depth thresholding in simple scenario. In our setup, we put a curtain on the background and also use plane fitting to remove the background points, yet with a larger distance threshold (200mm). In the rest of this section, we discuss the performance of both our pose tracker and cloth simulation components, and then show some final composite results.

Cloth Simulation Performance

Offline Performance. The clothing meshes used in our offline cloth simulator contain 20K to 50K vertices. The dynamic simulation time step is typically 1/3600s and collisions are handled every five time steps. For most examples, each frame takes approximately one

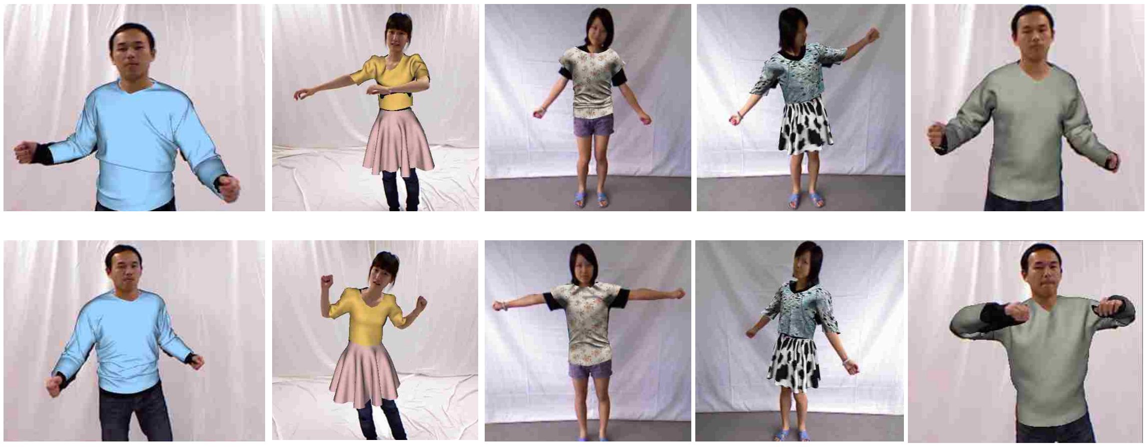


Figure 8.3: Sample final results on both male and female subjects. The results shown in the last column are generated using our near real-time simulation engine while the others are generated with higher quality offline simulation engine.

minute to simulate. We note that collision detection and handling is typically the bottleneck in our simulator and it uses at least 70 percent of the total computational cost.

Near Real-time Performance. Our simulator is also able to run at an interactive rate of 8 to 12FPS, where the clothing meshes are down sampled to 1K to 2K vertices. Since our data capture system still captures image data at 30FPS, this means the human body cannot move arbitrarily fast. The computational cost of this simulator depends on the resolution of the clothing meshes and the resolution of virtual depth maps. We note that the result of the low-resolution simulation does not contain high-quality details as in the high-resolution simulation, even though it runs orders-of-magnitude faster.

Composite Results

Fig. 8.1 and Fig. 8.3 show some examples of our final results, that are provided to the user for a virtual try-on experience. As can be seen here, our system can effectively deal with both male and female subjects, partial and full body observations. Notice that we use the same template model for both cases, and rely on our shape adaptation method to captured the body shape of the user and eventually provide realistic results. The cloth simulation enables our system to deal with various type of clothes, for example the long sleeve, short sleeve and skirt (Fig. 8.3). Notice the realistic simulated wrinkles on the cloth. We believe such effects can provide a superior virtual try-on experience compared to many existing image-based methods.

8.1.4 Conclusion

In this chapter, we demonstrate the effectiveness of combining a real-time template-based joint pose and shape estimation, with physically-based cloth simulation in a virtual try-on system. With our proposed constraints for pose tracking as well as our novel method for shape adaptation, our system can effectively capture the motion and shape of the user, and then deliver realistic cloth simulation results to provide a virtual try-on experience. The next step is to speed up the cloth simulation component in our system so that we can improve its performance and make it more practical for live demonstrations. Our long-term goal is to build a virtual try-on system, which is more efficient, accurate, robust, and convenient.

8.2 Pose Estimation for Movement Dysfunction Identification in Smart Health

In this chapter, we present the application of our pose tracking techniques in assisting physical therapists to identify movement dysfunction of the patients. Traditionally physical therapists have assessed kinematic performance through direct observation and more recently with 2D video systems. While both methods have yielded important insights about an individuals altered kinematics (walking, reaching, running etc) they are unable to reliably and precisely quantify the 3D kinematics contributing to injuries. Thus, a significant limitation with clinically important ramifications remains in the ability to quantify the 3D mechanics common to many injuries physical therapists treat. This further hampers the ability to effectively retrain someone to use the appropriate mechanics, resulting in poorer outcomes, slower progression, and more costly physical therapy visits. Moreover, the current state of the art for technology to directly measure 3D motion is expensive, cumbersome, and not widely available to physical therapists. Thus a strong need remains for the development of a portable, low cost, easy-to-use system to give physical therapists precise 3D measurements of movement dysfunction. This will lead to improved decision making for interventions and a means of objectively tracking outcomes.

In the long run we envision that physical therapists would be able to remotely monitor an individual's progress and provide feedback, decreasing the need for costly office visits. The physical therapist could also set alarms and triggers if the patient started to exhibit a motion pattern that was putting the patient at risk for injury. As a first step to achieving this long term goal we propose to develop and test the utility of a single 3D camera system to provide very precise and reliable 3D motion data in the laboratory, clinic and ultimately

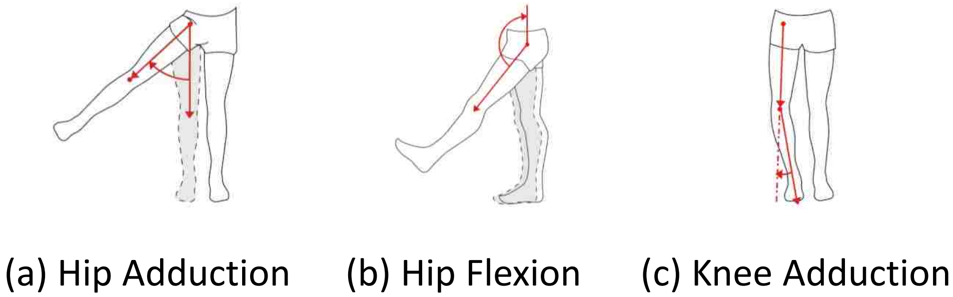

(a) Hip Adduction (b) Hip Flexion (c) Knee Adduction

Figure 8.4: The angles of interests for three different medical conditions in lower extremity. From left to right: (a) hip adduction angle for patellofemoral pain (PFP); (b) hip flexion angle for Anterior cruciate ligament reconstruction (ACL); and (c) knee adduction for knee osteoarthritis (knee OA).

at home. This type of tool could first be used to diagnosis individuals who have poor kinematic movement patterns related to the underlying medical condition, which could then be addressed through a clinic and home based kinematic retraining system.

In the application, we will focus on the motion analysis required in the following three medical conditions: patellofemoral pain (PFP), anterior cruciate ligament reconstruction (ACL), and knee osteoarthritis (OA). For PFP, the angle we believe is most critical is the hip adduction angle (pelvis dropping on the femur), previous research has shown the capability to detect clinically meaningful differences of 3 degrees between those with and without PFP. For ACL, we focus on the measurement of hip flexion: the femur flexing on the pelvis. We have found in recently collected data from our collaborators that those who have had an ACL reconstruction have 6 degrees less hip flexion at initial contact with the ground, leading to significantly greater impact forces as compared to healthy controls. For knee OA, the angle we focus on is knee adduction (tibia adducting on the femur), which is common in those in those with medial knee OA. Greater knee adduction angle, affects the knee adduction moment, a known biomechanical variable in the progression of OA. As the knee adduction angle increases the distance between the joint center and the ground reaction force increases and thus increasing the torque about the medial aspect of the knee. Very slight reduction in adduction, as little as 2 degrees, can reduce the knee adduction moment by 19%. For each of these conditions, the angles of interests are illustrated in Fig. 8.4.

Different from general motion capture scenarios, the medical applications has more stringent requirements. The first is accuracy: the required accuracy for diagnostic purpose is one degree of angular error. That translates to less than 10mm of positional error for large bone segment (such as the femur). Towards this end, we build the system on top of our model-based pose tracking algorithm (Chapter 5), with a scanned personalized model instead of a generic template as in the general scenario. The second requirement is anatomically-correct interpretation. In the traditional way of marker-based motion capture systems, marker placement must be precisely controlled to make sure the resulting pose is meaningful and consistent with medical standard . We address this issue by recording anatomically correct markers during the template model scanning process. In this work, we focus on medical conditions that are related to lower body movements. Doing so brings a number of benefits. First, the degrees of freedom in skeleton configuration are significantly reduced. Secondly, occlusion is less severe than in the upper body. Thirdly, the depth sensor can be placed at a closer range to the subject, effectively increasing the spatial resolution, and even depth resolution for stereo-based sensors such as Kinect.

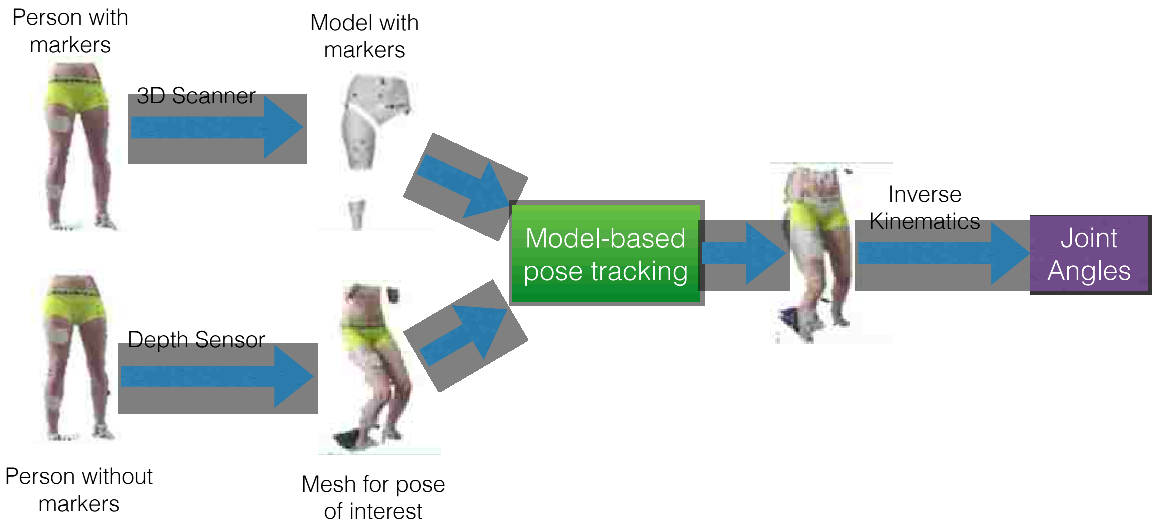


Figure 8.5: The diagram of our marker-less motion capture system for lower extremity joint angle assessment.

8.2.1 Data Collection and Processing

An overview of our system is provided in Figure 8.5. Since anatomical markers should be contained in the template model, anatomical markers are attached to the subject's lower body first. The subject is then scanned with a high precision Structured Light 3D scanner [74] to produce a template model with markers that can be repeatedly used for tracking purpose. Although rigging and skinning can be performed on the model, it requires considerable amount of effort is not necessary for our applications. Therefore, we manually segmented the model into different body parts and use it as an articulated model, namely with binary skinning weights.

For the assessment of movements, the subject does not need to wear any marker. As the subject performs the motion of interest, the depth sensor records the measurement data. With the scanned personalized template model, we apply our model-base pose tracking algorithm (Chapter 5) to estimate the pose of the subject's lower body in each frame. Via

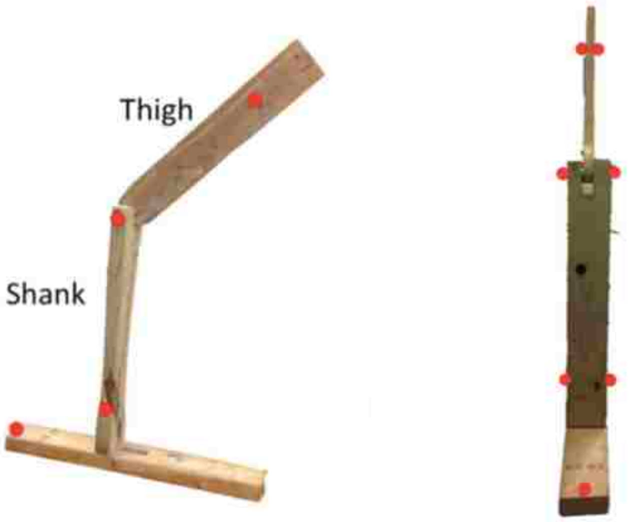


Figure 8.6: Our testing jig in sagittal (left) and frontal (right) view.

inverse kinematics, we can then calculate the joint angles from the pose. More details can be found in [102, 103].

8.2.2 Preliminary Experimental Results

Our estimation results are compared with the ground truth data collected using a 10 camera motion capture system (three Eagle cameras and seven Eagle 4 cameras from Motion Analysis Corp, Santa Rosa, USA [5]). In order for the marker system to capture ground truth data, the markers remain on the subject's body during movement assessment. For comparison, joint angle curves are aligned between systems using cross-correlation [33]. Up to the point of writing this dissertation, we have conducted two sets of experiments, one on an artificial jig(Figure 8.6) for flexion-extension measurements [103] and the other on human subjects for complete hip and knee joint angles measurements [102]. In the rest of this section, the results of these two sets of experiments are presented.

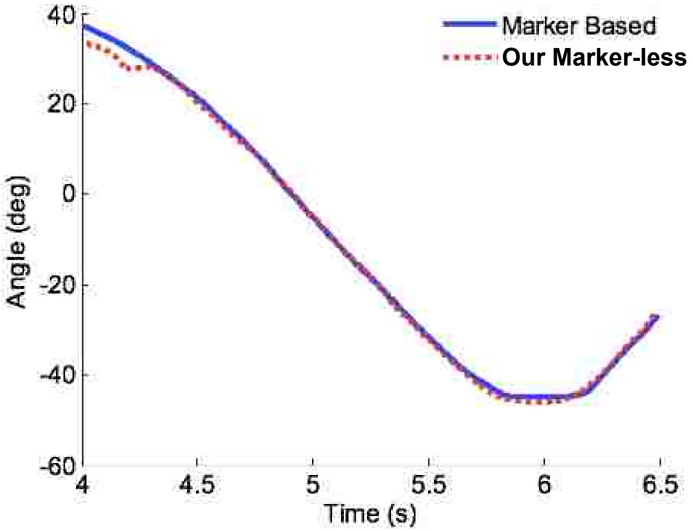


Figure 8.7: The flexion-extension angle measured by the marker-based system and our marker-less system. Positive angle denotes extension.

Experiments on an Artificial Jig

Our first set of experiments are conducted on an artificial jig with seven retroreflective markers as shown in Figure 8.6. These first step experiments are designed to evaluate the feasibility of applying our marker-less motion capture system to this medical application. For both systems, Visual3D (C-motion, Germantown, MD, USA [1]) is used to lowpass filter marker trajectories at 8 Hz and to compute the flexion-extension angle of the distal segment relative to the proximal segment.

The comparison of the results from two systems are shown in Table 8.1 and Figure 8.7. Both motion capture techniques produced similar results for calculations of sagittal plane motion (Table 8.1). In the static posture, both motion capture techniques are able to capture the joint angle within 0.9 degrees. When comparing the two systems during a flexion-extension motion, we find a similar pattern of motion with an average difference of 0.9 ± 1.0

degrees between the two systems (Figure 8.7). These results provide initial evidence that a precise and accurate algorithm can be developed for measuring kinematic data using our marker-less motion capture system in the sagittal plane.

Table 8.1: Comparison of flexion-extension angle measurement for a jig configuration of $-45.6° \pm 0.1°$.

Marker-based	Error in marker-based	Marker-less	Error in marker-less	Difference between two systems
−45°	0.6°	−44.7°	0.9°	0.3°

Human Subjects

The next step of our study is determining the accuracy of our system in the frontal and transverse plane besides the sagittal plane motions. Therefore, our second set of experiments involves 15 healthy people as our subjects (8 male, 7 female, height 1.702 ± 0.089

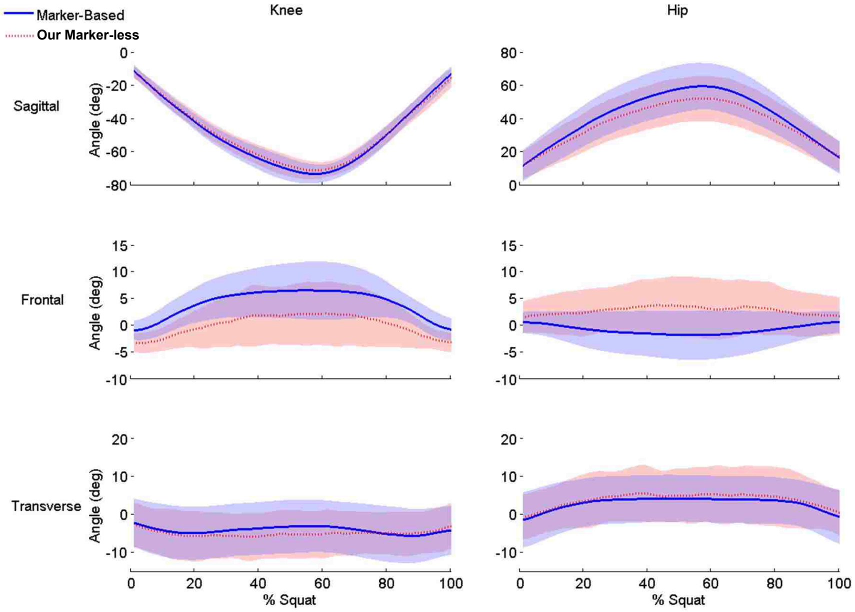


Figure 8.8: Ensemble curves (± standard deviation) of joint angles calculated by the marker-based and marker-less systems for a slow squat motion. The patterns of motion are similar for both the markerless and marker-based system. .

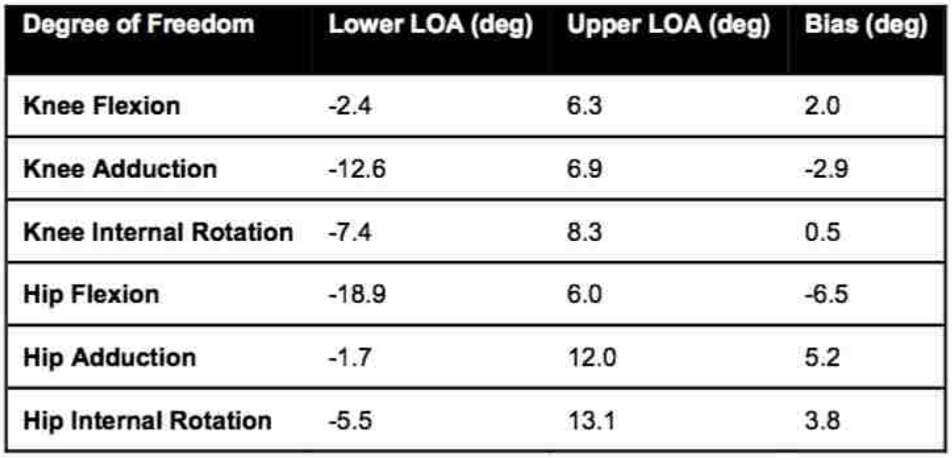
m, mass 67.9 ± 10.4 kg, age 24 ± 4 yrs, BMI 23.4 ± 2.2 kg/m^2). The subjects are asked to perform squat motion to 60 degrees of knee flexion as assessed using a manual goniometer, standing with feet placed in a standardized position [82].

The results compared to the ground truth data are presented in Figure 8.8. The patterns of motion are similar between systems where the difference between systems is greatest at the peak flexed position (i.e. bottom of the squat). The marker-less system underestimates peak hip flexion by 4.3 degrees (9% of the total range of hip flexion motion) ((Table 8.2) where the bias of hip flexion angles is -6.5 deg (Table 8.3). No significant differences are found in peak hip adduction, axial rotation, or knee angles (Table 8.2), in which the bias is ≤7 deg (Table 8.3). Peak joint angles show high between-trial reliability with ICC≥0.9 for both systems (Table 8.2). The marker-less system exhibits greater between-trial variability for all peak angles as quantified by the MDC (Table 8.2).

Table 8.2: Reliability and agreement of systems for calculating peak joint angles during a squat.

Degree of Freedom	Marker-less Value (deg)	ICC (-)	Minimal Detectable Change (deg)	Marker-based Value (deg)	ICC (-)	Minimal Detectable Change (deg)	P-value Comparing Marker-less with Marker-based	Average Absolute Difference Between Curves (deg)	Marker-less Excursion (deg)	Marker-based Excursion (deg)
Knee Flexion	-72 (5.4)	0.918	4.3	-74 (5.9)	0.950	3.6	0.462	1.4	60.4	63.1
Knee Adduction	3.5 (5.5)	0.959	3.1	7.1 (5.1)	0.996	0.9	0.183	4.0	7.9	8.3
Knee Internal Rotation	-0.5 (5.6)	0.977	2.3	-0.1 (6.5)	0.993	1.5	0.887	1.1	7.7	7.0
Hip Flexion	53.1 (13.4)	0.974	6	60.2 (14)	0.992	3.5	***0.009***	4.3	40.6	48.2
Hip Adduction	5.6 (4.3)	0.934	3.1	1.4 (2.6)	0.987	0.8	0.119	3.6	6.0	4.0
Hip Internal Rotation	7.7 (7)	0.964	3.7	5.4 (6.4)	0.998	0.9	0.381	0.9	9.7	7.7

Table 8.3: 95% limits of agreement (LOA) and the bias of the motion capture systems.



Degree of Freedom	Lower LOA (deg)	Upper LOA (deg)	Bias (deg)
Knee Flexion	-2.4	6.3	2.0
Knee Adduction	-12.6	6.9	-2.9
Knee Internal Rotation	-7.4	8.3	0.5
Hip Flexion	-18.9	6.0	-6.5
Hip Adduction	-1.7	12.0	5.2
Hip Internal Rotation	-5.5	13.1	3.8

8.2.3 Conclusion

The use of our marker-less motion capture system holds potential as a clinical surrogate for the assessment of 3D motion. In this study, we compare its ability to measure joint angles with current state of the art, a marker-based motion capture system. Both systems agree well for the shape of the motion calculated and has high between trial reliability. Our system delivers promising preliminary results in this direction. Further experiments and research are needed to investigate its applicability for faster motion including walking and running. There are several hardware limitation that needs to be overcome, such as limited frame rate of the depth sensor and motion blur that follows.

Chapter 9 Conclusion and Future Work

In this dissertation, we have explored the potential of using a single commodity depth sensor for motion and shape estimation for quasi-articulated objects. We have developed three algorithms that deliver state-of-the-art results. They achieve shape reconstruction with increasing levels of details. Built on top of our pose estimation algorithm, we further developed a system for high quality shape reconstruction. We apply our techniques in applications in various domains, including entertainment and health care and have delivered promising results. This chapter summarizes our contributions, limitations of our algorithms and future research extensions.

9.1 Contributions

Pose and shape estimation from monocular data is a challenging problem. We have demonstrated that by leveraging a pre-recorded motion database, our data-driven algorithm can reliably estimate pose of human subjects with various body sizes (Chapter 4). Statistical shape analysis using PCA allows fast similar shape retrieval from the database. Our novel view-independent shape encoding based on principal axes dramatically reduces the size of the data samples required. The utilization of non-rigid point cloud registration enable our technique to accommodate various body sizes. Combining these novel components, our algorithm, among one of the first works tailored for depth sensor, achieved state-of-the-art accuracy.

Our studies on model-based approaches advances the techniques used in the scenarios where temporal consistency is desired. Our first work relies on the widely used iterative closest point (ICP) strategy (Chapter 5). Such local search strategy is known to be sensitive to local minima. We have investigated novel constraints based on visibility, semantic information from template as well as edge information to guide the local optimization. Incorporating linearized twist-based representation for poses, our algorithm is capable to run in real time with GPU implementation. Based on the same framework, we have further proposed algorithms for automatic shape and body size adaptation for better personalization.

Our ICP-based algorithm has been successfully applied to two different applications, namely Virtual Try-On (Section 8.1) and Smart Health (Section 8.2). In the first application, we combine our algorithms with cloth simulation techniques to deliver a realistic virtual clothing system. The promising results have demonstrated the effectiveness of our techniques. For Smart Health application, we enhanced the performance of our pose tracking algorithm by using scanned personalized model as the template. Our preliminary results have demonstrated the potential of our algorithms for applications in the area where high accuracy is desired.

To further reduce the sensitivity of the ICP-based approach to local minima, we further proposed a model-based approach based on probabilistic correspondences association using the Gaussian Mixture Model (GMM) (Chapter 6). The soft correspondences association with GMM relates our template model with the observation in a more global fashion compared to closest point strategy. Our algorithm better accommodates fast and complex motion. The soft correspondences association is a fully parallel operation is particularly suitable to run on GPU. By embedded the articulated/skinned motion model based on exponential map that can be approximated with linearization, our approach achieved real-time performance with GPU implementation. Within the same probabilistic framework, we have developed algorithms for automatic shape and body size adaptation, similar to our ICP-based algorithm. Our algorithm achieved state-of-the-art accuracy compared to various existing methods as demonstrated by our extensive quantitative evaluations on publicly-available datasets. As an extension, we have shown that our algorithm can be used to register across a set of shape collections to build correspondences. Automatic rigging for the entire shape collection can then be achieved by transferring the information from our template.

Our above algorithms were able to estimate the body shape for the subject but only has limited personalization in terms of details. Therefore, we have proposed a system for high quality shape reconstruction based on our pose estimation technique. Our novel greedy frame selection strategy selects a few representative frames out of a entire sequence of images that cover as many areas of the body as possible with minimal number frames. The selected frames are then aligned together with our thoughtfully designed two-step non-rigid registration technique. Our system delivers high quality shapes while allowing the subjects to perform free-form movements. Our experiments have shown promising results on various subjects with different apparels.

9.2 Future Work

We have discussed the limitation and possible extensions of our techniques at the end of Chapter 4, 5, 6 and 7. In this section, we discuss more broadly on future research topics.

One of the current major limitations in depth-based pose estimation research is perhaps the lack of well designed evaluation benchmarks. There are only a few publicly available datasets that include ground truth [52, 53, 62]. However, the early one [52] contains only relatively simple frontal view data and might not be sufficient for evaluations. Although the later ones [53, 62] include more challenging poses, none of them provides good ground truth data. One of the major difficulty is the synchronization of the marker-based motion capture system that provides ground truth data with the depth sensor. Therefore, datasets that are better designed, containing more faithful ground truth data are desired for evaluation purpose.

Most of existing works using depth sensors including ours are focused on single person scenario. There have been a few exceptions. The KinectSDK [4] is capable of estimating skeletons of multiple people. Ye et al [131] tracks multiple people, however requires multiple depth sensors. Extension of our techniques and other model-based approaches by combining with segmentation algorithms is a possibility to accommodate multiple people, similar to the work by Liu et al [81]. However, the computational complexity can be significantly higher if joint optimization is performed on multiple subjects. Therefore, it is desired to investigate model-based approaches for multi-subject tracking without dramatically increasing the computational cost. In this aspect, discriminative methods based on machine learning techniques for body part detection [94] or segmentation [104] might be more appropriate. However, this line of research needs to improve the accuracy and robustness to occlusion. Hybrid approaches that combines the complementary characteristics of both categories could be a future trend. Some early promising results have been delivered, for example by Wei et al [128]. More seaming-less and effective ways of integrating techniques in these two categories is an interesting future topic to study.

Related to this line of research is the study of subjects' interaction with other objects or the environment. There have been some studies on hand motion capture during interaction [17, 126]. Yet, there is lack of research of full body human pose estimation in this context. Since human motion is naturally related to objects interacted with or the entire environment, taking these information into consideration could help pose estimation, for example in the case of severe occlusions.

Going a step further, actions and activities that a person performs is highly related to the poses. There have been extensive works on understanding of human actions and activities from color information [9] and depth information [135]. Recently, some researchers have started to consider performing these two tasks jointly [130] with a multi-view setup, given the inherent correlation. No work has been done using a single commodity depth sensor in this direction to the best of our knowledge. It is interesting and also could be promising for future investigation.

In the scope of shape reconstruction, existing work has very limited capability to accommodate dynamic scenes. Our algorithm severs as one of the first attempts to solve this problem, along with other recent works [75, 76, 139]. However, our algorithm requires a generic template while the others require either a multi-view setup [76] or personalized model [75, 139]. These assumptions still limit their usability in real work scenarios. Future research could further explore methods that does not require prior model while allowing the subject to perform free form movements.

One research direction that is common to both pose estimation and shape reconstruction is to combine the color cues with depth cues. Currently most existing approaches using depth sensors rely solely on depth information [13, 52, 53, 62, 133, 134]. However, some of the motions, such as arm twist, can be hardly captured by the depth sensor, but might be observed based on the color information. Shape reconstruction in dynamic scene is generally built upon successful estimation of the motion and therefore can be inaccurate in these cases. Therefore it is desired to study strategies to combine these two cues, along with other cues, such as empty space constraint [53], to achieve higher accuracy and fidelity for both pose estimation and shape reconstruction.

Bibliography

[1] C-motion research biomechanics. http://www.c-motion.com/products/visual3d/, 2014. 122

[2] Cyberware. http://cyberware.com, 2014. 4

[3] Mesa imaging. http://www.mesa-imaging.ch/home/, 2014. 3, 26

[4] Microsoft kinect. http://www.microsoft.com/en-us/kinectforwindows/, 2014. x, xi, 3, 11, 57, 58, 74, 129

[5] Motion analysis corp. http://www.motionanalysis.com, 2014. 121

[6] Photonic mixer devices (pmd). http://www.pmdtec.com, 2014. 3

[7] Shapify.me. https://shapify.me/, 2014. 101

[8] Vicon. http://www.vicon.com, 2014. 3, 5, 7, 27

[9] J. Aggarwal and M. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16:1–16:43, Apr. 2011. 130

[10] N. Ahmed, C. Theobalt, C. Rossl, S. Thrun, and H.-P. Seidel. Dense correspondence finding for parametrization-free animation reconstruction from video. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 1–8. IEEE, 2008. 78, 86

[11] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, pages 408–416, New York, NY, USA, 2005. ACM. 15

[12] Aristotle. On the motion of animals (translated by a. farquharson. originally published 350 b.c.e.). 1

[13] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE 13th International Conference on Computer Vision (ICCV)*, pages 1092–1099. IEEE, Nov. 2011. 13, 130

[14] P. Baerlocher. *Inverse kinematics techniques for the interactive posture control of articu-lated figures*. PhD thesis, Ecole Polytechnique Federale de Lausanne, 2001. ix, 20

[15] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker. Detailed human shape and pose from images. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 1–8. IEEE, 2007. 2

[16] L. Ballan and G. M. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, Atlanta, GA, USA, June 2008. 2, 10, 14

[17] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision (ECCV)*, Firenze, October 2012. 130

[18] D. Baraff and A. Witkin. Large steps in cloth simulation. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, pages 43–54. ACM, 1998. 108

[19] J. T. Barron and J. Malik. Intrinsic scene properties from a single rgb-d image. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 17–24. IEEE, 2013. 99

[20] G. Blais and M. Levine. Registering multiview range data to create 3d computer objects. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):820–824, 1995. 2

[21] A. Bleiweiss and E. E. G. Kutliroff. Markerless motion capture using a single depth sensor. In *ACM SIGGRAPH ASIA 2009 Sketches*, page 20. ACM, 2009. 12

[22] G. Borelli. *De Motu Animalium (On the Movement of Animals, translated from Latin to English by P. Maquet, Springer, 1989)*. Springer-Verlag, 1680/1681. 1

[23] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *Int. J. Comput. Vision*, 56(3):179–194, Feb. 2004. 10, 22, 60

[24] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In E. Fiume, editor, *ACM Trans. Graph., (Proc. of SIGGRAPH)*, volume 21, pages 594–603, 2002. 108, 113

[25] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proc. of SCA*, pages 28–36, 2003. 112

[26] B. J. Brown and S. Rusinkiewicz. Global non-rigid alignment of 3-d scans. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 15

[27] W. Chang and M. Zwicker. Automatic registration for articulated shapes. *Proc. of Symp. on Geom. Processing*, pages 1459–1468, 2008. 17

[28] W. Chang and M. Zwicker. Global registration of dynamic range scans for articulated model reconstruction. *ACM Trans. Graph.*, 30(3):26:1–26:15, May 2011. 2, 15, 17, 78, 86, 93, 101

[29] Y. Chen, Z. Liu, and Z. Zhang. Tensor-based human body modeling. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 105–112. IEEE, 2013. 15

[30] Z. Cheng and K. Robinette. Static and dynamic human shape modeling-a review of the literature and state of the art. Technical report, DTIC Document, 2009. 2

[31] K. Choi and H. Ko. Research problems in clothing simulation. *Computer Aided Design*, 37:585–592, 2005. 108

[32] K.-J. Choi and H.-S. Ko. Stable but responsive cloth. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 21(3):604–611, July 2002. 108

[33] R. A. Clark, Y.-H. Pua, K. Fortin, C. Ritchie, K. E. Webster, L. Denehy, and A. L. Bryant. Validity of the microsoft kinect for assessment of postural control. *Gait & posture*, 36(3):372–377, 2012. 121

[34] Y. Cui, W. Chang, T. Nöll, and D. Stricker. Kinectavatar: Fully automatic body capture using a single kinect. In *ACCV 2012 Workshop on Color Depth Fusion in Computer Vision*, Nov 2012. 4, 15, 17, 78, 101

[35] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3d shape scanning with a depth camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1, 2012. 16, 61, 62, 70

[36] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 1996. 2, 95

[37] E. de Aguiar, L. Sigal, A. Treuille, and J. K. Hodgins. Stable spaces for real-time clothing. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 29(4):106:1–106:9, July 2010. 108

[38] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. Graph.*, 2008. 2, 3, 10, 14, 15, 16, 78

[39] E. de Aguiar, C. Theobalt, C. Stoll, and H.-P. Seidel. Marker-less deformable mesh tracking for human shape and motion capture. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, 2007. 10

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977. 61

[41] J. Ehara and H. Saito. Texture overlay for virtual clothing based on pca of silhouettes. In *Mixed and Augmented Reality, IEEE/ACM International Symposium on*, pages 139–142, 2006. 109

[42] W.-W. Feng, Y. Yu, and B.-U. Kim. A deformation transformer for real-time cloth animation. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 29(4), July 2010. 108

[43] D. Fisher, M. Williams, and T. Andriacchi. The therapeutic potential for changing patterns of locomotion: An application to the acl deficient knee. In *ASME Bioengineering Conference*, 2003. 3

[44] Fitnect. Fitnect. http://www.fitnect.com, 2013. 106, 110

[45] Fits.Me. Fits.Me. http://fits.me, 2014. 109

[46] D. A. Forsyth, O. Arikan, L. Ikemoto, J. O'Brien, and D. Ramanan. Computational studies of human motion: Part 1, tracking and motion synthesis. *Foundations and Trends in Computer Graphics and Vision*, 1(23):77–254, 2006. 9

[47] A. Fossati, M. Dimitrijevic, V. Lepetit, and P. Fua. From canonical poses to 3d motion capture using a single camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1165–1181, 2010. 10

[48] R. M. Friborg, S. Hauberg, and K. Erleben. Gpu accelerated likelihoods for stereo-based articulated tracking. In *Trends and Topics in Computer Vision*, pages 359–371. Springer, 2012. 12

[49] J. Gall. *Filtering and optimization strategies for markerless human motion capture with skeleton-based shape models*. PhD thesis, Universitt des Saarlandes, Saarbrcken, Germany, 2009. 9, 10

[50] J. Gall, A. Fossati, and L. Van Gool. Functional categorization of objects using real-time markerless motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1969–1976, 2011. 42, 43, 44, 46

[51] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H.-P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 1746–1753. IEEE, 2009. 2, 3, 14, 22, 46, 48, 50, 59, 63, 66, 74

[52] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 755–762, 2010. ix, xi, 5, 6, 7, 14, 34, 35, 71, 72, 129, 130

[53] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time human pose tracking from range data. In *European Conf. on Comput. Vision*, 2012. 6, 12, 14, 52, 59, 72, 73, 77, 129, 130, 131

[54] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997. 90

[55] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Int. Conf. on Comput. Vision*, pages 415–422, 2011. 11

[56] N. K. Govindaraju, S. Redon, M. C. Lin, and D. Manocha. Cullide: interactive collision detection between complex models in large environments using graphics hardware. In *Proc. of Symp. on Geom. Processing*, pages 25–32, 2003. 106, 109, 111

[57] P. Guan, A. Weiss, A. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *Int. Conf. on Comput. Vision*, pages 1381–1388, October 2009. 15

[58] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H.-P. Seidel. A statistical model of human pose and body shape. In *Comput. Graph. Forum*, volume 28, pages 337–346. Wiley Online Library, 2009. 16, 76

[59] S. Hauswiesner, M. Straka, and G. Reitmayr. Temporal coherence in image-based visual hull rendering. *IEEE Trans. on Visualization and Computer Graphics*, 19(10):1758–1767, 2013. 109

[60] S. Hauswiesner, M. Straka, and G. Reitmayr. Virtual try-on through image-based rendering. *IEEE Trans. on Visualization and Computer Graphics*, 19(9):1552–1565, 2013. 109

[61] B. Heidelberger, M. Teschner, and M. Gross. Realtime volumetric intersections of deforming objects. In *Proc. of Vision, Modeling and Visualization*, pages 461–468, 2003. 106, 109, 111

[62] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *Proceedings of the 3rd joint 3DIM/3DPVT Conference (3DV)*, 2013. 6, 13, 14, 15, 42, 43, 64, 66, 68, 72, 73, 74, 75, 129, 130

[63] T. Helten, A. Baak, M. Müller, and C. Theobalt. Full-body human motion capture from monocular depth images. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 188–206. Springer, 2013. 9

[64] E. Herbst, X. Ren, and D. Fox. Rgb-d flow: Dense 3-d motion estimation using color and depth. In *IEEE Int. Conf. on Robotics and Automation*, pages 2276–2282. IEEE, 2013. 104

[65] D. Hirshberg, M. Loper, E. Rachlin, and M. Black. Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *European Conf. on Comput. Vision*, volume 7577 of *Lecture Notes in Computer Science*, pages 242–255. Springer Berlin Heidelberg, 2012. 15

[66] D. House and D. Breen. *Cloth Modeling and Animation*. AK Peters, 2000. 108

[67] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973. 2

[68] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. 17

[69] J. M. Kaldor, D. L. James, and S. Marschner. Simulating knitted cloth at the yarn level. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 27(3):65:1–65:9, August 2008. 108

[70] L. Kavan, D. Gerszewski, A. W. Bargteil, and P.-P. Sloan. Physics-inspired upsampling for cloth simulation in games. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 30(4):93:1–93:10, August 2011. 108

[71] M. M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 61–70, 2006. 87, 94, 96

[72] S. Knoop, S. Vacek, and R. Dillmann. Fusion of 2d and 3d sensor data for articulated body tracking. *Robotics and Autonomous Systems*, 57(3):321–329, 2009. 12

[73] D. Knott and D. K. Pai. Cinder: Collision and interference detection in real-time using graphics hardware. In *Proc. of Graphics Interface*, 2003. 106, 109, 111

[74] D. Lanman and G. Taubin. Build your own 3d scanner: 3d photography for beginners. In *SIGGRAPH '09: ACM SIGGRAPH 2009 Courses*, 2009. 2, 97, 120

[75] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, pages 175:1–175:10, 2009. 16, 130

[76] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM Trans. Graph.*, 31(1):2:1–2:11, Feb. 2012. 15, 130

[77] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1421–1430, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. 87

[78] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. Graph., (Proc. of SIGGRAPH Asia)*, 32(6), November 2013. 2, 6, 15, 17, 78, 86, 87, 93, 95, 97, 98, 99, 102, 103, 104

[79] M. Liao, Q. Zhang, H. Wang, R. Yang, and M. Gong. Modeling deformable objects from a single depth camera. In *Int. Conf. on Comput. Vision*, 2009. 78, 86

[80] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph.*, 2007. 29, 30

[81] Y. Liu, J. Gall, C. Stoll, Q. Dai, H.-P. Seidel, and C. Theobalt. Markerless motion capture of multiple characters using multiview image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2720–2735, 2013. 2, 129

[82] W. McIlroy and B. Maki. Preferred placement of the feet during quiet stance: development of a standardized foot placement for balance testing. *Clinical Biomechanics*, 12(1):66–70, 1997. 124

[83] T. B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.*, 81(3):231–268, 2001. 9

[84] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.*, 2006. 2, 9

[85] D. D. Morris and J. M. Rehg. Singularity analysis for articulated object tracking. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 289–296. IEEE, 1998. 10

[86] T. Morvan, M. Reimers, and S. E. High performance GPU-based proximity queries using distance fields. In *Comput. Graph. Forum*, volume 27, pages 2040–2052, december 2008. 109

[87] R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1994. ix, 18, 19, 22

[88] MVM. My Virtual Model. http://www.mvm.com/index.html, 2014. 109

[89] A. Myronenko and X. B. Song. Point-set registration: Coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010. 60, 61, 62, 70, 77

[90] A. Nealen, M. Mueller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Comput. Graph. Forum*, 25(4):809–836, 2006. 108

[91] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and Augmented Reality, IEEE/ACM International Symposium on*, ISMAR '11, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society. 2, 4, 16, 78, 87, 90, 101, 104

[92] Y. Pekelny and C. Gotsman. Articulated object reconstruction and markerless motion capture from depth video. In *Comput. Graph. Forum*, volume 27, pages 399–408. Wiley Online Library, 2008. 12

[93] M. Pharr and R. Fernando. *GPU Gems 2: Programming techniques for high-performance graphics and general purpose computation*. 2005. 21, 27

[94] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-time identification and localization of body parts from depth images. In *IEEE Int. Conf. on Robotics and Automation*, 2010. 10, 11, 13, 14, 129

[95] R. Plänkers and P. Fua. Articulated soft objects for multiview shape and motion capture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1182–1187, 2003. 10

[96] R. Poppe. Vision-based human motion analysis: An overview. *Comput. Vis. Image Underst.*, 108(1):4–18, 2007. 9

[97] Primesense. OpenNI. http://structure.io/openni, 2011. ix, 39, 40

[98] K. Pulli. Multiview registration for large data sets. In *Int. Conf. on 3D Digital Imaging and Modeling*, pages 160–168. IEEE, 1999. 91

[99] B. Rosenhahn, R. Klette, and D. N. Metaxas. *Human motion: understanding, modelling, capture, and animation*, volume 36. Springer, 2007. 1, 2, 9

[100] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Int. Conf. on 3D Digital Imaging and Modeling*, 2001. 44

[101] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3d deformable surface and articulated pose reconstruction. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 647–654. IEEE, 2010. 10

[102] A. Schmitz, M. Ye, G. Boggess, R. Shapiro, R. Yang, and B. Noehren. The measurement of in vivo joint angles during a squat using a single camera markerless motion capture system as compared to a marker based system. *Gait and Posture*. submitted. 121

[103] A. Schmitz, M. Ye, R. Shapiro, R. Yang, and B. Noehren. Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system. *Journal of biomechanics*, 47(2):587–591, 2014. 7, 121

[104] 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 *IEEE Conf. on Comput. Vision and Patt. Recog.*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society. 11, 14, 55, 129

[105] 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. *Int. J. Comput. Vision*, 87(1-2):4–27, 2010. 2

[106] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, volume 1, pages 390–397. IEEE, 2005. 10

[107] C. Sminchisescu and B. Triggs. Covariance scaled sampling for monocular 3d body tracking. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, volume 1, pages I–447. IEEE, 2001. 10

[108] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proc. of Symp. on Geom. Processing*, SGP '04, pages 175–

184, New York, NY, USA, 2004. ACM. 91

[109] C. Stoll, J. Gall, E. De Aguiar, S. Thrun, and C. Theobalt. Video-based reconstruction of animatable human characters. In *ACM Trans. Graph.*, volume 29, page 139. ACM, 2010. 2

[110] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof. Simultaneous shape and pose adaption of articulated models using linear optimization. In *European Conf. on Comput. Vision*, volume 7572 of *Lecture Notes in Computer Science*, pages 724–737. Springer Berlin Heidelberg, 2012. 10, 14, 15, 16, 52

[111] Styku. Skytu. http://www.styku.com, 2012. 110

[112] A. Sud, N. Govindaraju, R. Gayle, and D. Manocha. Interactive 3D distance field computation using linear factorization. In *Proc. of I3D*, pages 117–124, 2006. 109

[113] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 16, 17, 87

[114] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. 16

[115] H. Tanaka and H. Saito. Texture overlay onto flexible object with pca of silhouettes and k-means method for search into database. In *Machine Vision and Applications*, 2009. 109

[116] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 103–110, 2012. 11

[117] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *IEEE Trans. on Visualization and Computer Graphics*, 18(4):643–650, Apr. 2012. 14, 16, 17

[118] triMirror. triMirror. http://www.trimirror.com, 2014. 106, 109, 110

[119] R. Urtasun, D. J. Fleet, and P. Fua. Monocular 3d tracking of the golf swing. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, volume 2, pages 932–938. IEEE, 2005. 10

[120] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, volume 1, pages 238–245. IEEE, 2006. 10

[121] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. In *ACM Trans. Graph., (Proc. of SIGGRAPH)*, SIGGRAPH '08, pages 97:1–97:9, New York, NY, USA, 2008. ACM. 3, 10, 14, 15, 16, 78

[122] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph.*, 28(4):105:1–105:16, September 2009. 112

[123] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph.*, 28:15:1–15:15, May 2009. 2

[124] H. Wang, F. Hecht, R. Ramamoorthi, and J. O'Brien. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 29(4):107:1–, July 2010. 108

[125] H. Wang, J. O'Brien, and R. Ramamoorthi. Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 30(4):71:1–71:12, July 2011. 112

[126] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai. Video-based hand manipulation capture through composite motion control. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 32(4):43, 2013. 130

[127] E. F. Weber and W. E. Weber. *Über die Mechanik der menschlichen Gehwerkzeuge. Eine anatomisch-physiologische Untersuchung (Translation by P. Maquet and R. Furlong: Mechanics of the Human Walking Apparatus. Springer, Berlin, 1992) ”*. G̈ottingen, Dieterich, 1836. 2

[128] X. Wei, P. Zhang, and J. Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Trans. Graph., (Proc. of SIGGRAPH Asia)*, 31(6):188:1–188:12, Nov. 2012. 14, 66, 73, 129

[129] A. Weiss, D. Hirshberg, and M. J. Black. Home 3d body scans from noisy image and range data. In *Int. Conf. on Comput. Vision*, ICCV '11, pages 1951–1958, Washington, DC, USA, 2011. IEEE Computer Society. 14, 15, 75, 78

[130] A. Yao, J. Gall, and L. Van Gool. Coupled action recognition and pose estimation from multiple views. *Int. J. Comput. Vision*, 100(1):16–37, 2012. 130

[131] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance capture of interacting characters with handheld kinects. In *European Conf. on Comput. Vision*, pages 828–841, 2012. 13, 129

[132] M. Ye, H. Wang, N. Deng, X. Yang, and R. Yang. Real-time human pose and shape estimation for virtual try-on using a single commodity depth camera. *IEEE Trans. on Visualization and Computer Graphics*, 20(4):550–559, 2014. 6, 7, 13, 14

[133] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3d pose estimation from a single depth image. In *Int. Conf. on Comput. Vision*, pages 731–738. IEEE, 2011. xi, 5, 6, 13, 55, 71, 72, 130

[134] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, pages 2353–2360, June 2014. 6, 13, 14, 79, 81, 130

[135] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, and J. Gall. A survey on human motion analysis from depth data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 149–187. Springer, 2013. 130

[136] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *IEEE Conf. on Comput. Vision and Patt. Recog.*, June 2013. 15, 17, 46

[137] Z. Zhou, B. Shu, S. Zhuo, X. Deng, P. Tan, and S. Lin. Image-based clothes animation for virtual fitting. In *SIGGRAPH Asia 2012 Technical Briefs*, SA '12, pages 33:1–33:4, New York, NY, USA, 2012. ACM. 106, 109

[138] Y. Zhu, B. Dariush, and K. Fujimura. Kinematic self retargeting: A framework for human pose estimation. *Comput. Vis. Image Underst.*, 114(12):1362 – 1375, 2010. 10, 11

[139] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, et al. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Trans. Graph., (Proc. of SIGGRAPH)*, 2014. 130

VITA

NAME

Mao Ye

YEAR OF BIRTH

1987

PLACE OF BIRTH

Nan’an, Fujian, People’s Republic of China

EDUCATION

- July 2008: B.E. in Information Security, University of Science and Technology of China, Hefei, Anhui, China

PUBLICATIONS

Book Chapters

- **Mao Ye**, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A Survey on Human Motion Analysis from Depth Data. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, Lecture Notes in Computer Science*, Volume 8200, 2013, pp 149-187

Journal Articles

- **Mao Ye**, Huamin Wang, Nianchen Deng, Xubo Yang and Ruigang Yang. Virtual Try-On Using a Single Commodity Depth Camera. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2014 Apr;20(4):550-9.

- Anne Schmitz, **Mao Ye**, Robert Shapiro, Ruigang Yang, and Brian Noehren. Accuracy and repeatability of joint angles measured using a single camera markerless motion capture system. *Journal of Biomechanics*, Vol 47, Issue 2, 22 January 2014, pp 587-591

- Yuhua Xu, **Mao Ye**, Zunhua Tian, Xiaohu Zhang. Locally Adaptive Combining Color and Depth for Human Body Contour Tracking Using Level Set Method. *Institution of Engineering and Technology, IET Computer Vision*, January 2014

- Hui Lin*, Jizhou Gao* (*joint first authors), Yu Zhou, Guiliang Lu, **Mao Ye**, Chenxi Zhang, Ligang Liu, and Ruigang Yang, Semantic Decomposition and Reconstruction of Residential Scenes from LiDAR Data, *ACM Transaction on Graphics (TOG) - Proceedings of ACM SIGGRAPH*, Volume 32, Issue 4, 2013.

Refereed Conference Proceedings

- **Mao Ye** and Ruigang Yang. Real-time Simultaneous Pose and Shape Estimation for Articulated Objects with a Single Depth Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014

- Chenxi Zhang, **Mao Ye** and Ruigang Yang. Data-driven Flower Petal Modeling with Botany Priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014

- Qing Zhang, Bo Fu, **Mao Ye** and Ruigang Yang. Quality Dynamic Human Body Modeling Using a Single Low-cost Depth Camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014

- **Mao Ye**, Cha Zhang and Ruigang Yang. Video Enhancement of People Wearing Polarized Glasses: Darkening Reversal and Reflection Reduction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013

- Qing Zhang, **Mao Ye**, Ruigang Yang, Yasuyuki Matsushita, Bennett Wilburn and Huimin Yu. Edge-Preserving Photometric Stereo via Depth Fusion. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012

- Bo Fu, **Mao Ye**, Ruigang Yang and Cha Zhang. See-Through Image Enhancement Through Sensor Fusion. In *International Conference on Multimedia & Expo (ICME)*, 2012

- **Mao Ye**, Xianwang Wang, Ruigang Yang, Liu Ren and Marc Pollefeys. Accurate 3D Pose Estimation from a Single Depth Image. In *International Conference on Computer Vision (ICCV)*, 2011

- H. P. Kuo, K. H. Tan, **M. Ye**, R. L. Cobene, X. Li, L. Hubby,A. M. Bratkovsky, B. Culbertson. Glasses-Free Projection Continuous 3D Displays. In *the 19th International Display Workshops (IDW)*, 2011

- Matt Steele, **Mao Ye**, and Ruigang Yang. Color Calibration of Multi-Projector Displays through Automatic optimization of Hardware Settings. In *PROCAMS*, 2009