EGOCENTRIC RECONSTRUCTION OF HUMAN BODIES FOR REAL-TIME MOBILE TELEPRESENCE

YoungWoon Cha

A dissertation submitted to the faculty at the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2021

Approved by:

Henry Fuchs

Gary Bishop

Jan-Michael Frahm

Shahram Izadi

Shahriar Nirjon

ABSTRACT

YoungWoon Cha: Egocentric Reconstruction of Human Bodies for Real-time Mobile Telepresence
(Under the direction of Henry Fuchs)

A mobile 3D acquisition system has the potential to make telepresence significantly more convenient, available to users anywhere, anytime, without relying on any instrumented environments. Such a system can be implemented using egocentric reconstruction methods, which rely only on wearable sensors, such as head-worn cameras and body-worn inertial measurement units. Prior egocentric reconstruction methods suffer from incomplete body visibility as well as insufficient sensor data.

This dissertation investigates an egocentric 3D capture system relying only on sensors embedded in commonly worn items such as eyeglasses, wristwatches, and shoes. It introduces three advances in egocentric reconstruction of human bodies. (1) A parametric-model-based reconstruction method that overcomes incomplete body surface visibility by estimating the user's body pose and facial expression, and using the results to re-target a high-fidelity pre-scanned model of the user. (2) A learning-based visual-inertial body motion reconstruction system that relies only on eyeglasses-mounted cameras and a few body-worn inertial sensors. This approach overcomes the challenges of self-occlusion and outside-of-camera motions, and allows for unobtrusive real-time 3D capture of the user. (3) A physically plausible reconstruction method based on rigid body dynamics, which reduces motion jitter and prevents interpenetrations between the reconstructed user's model and the objects in the environment such as the ground, walls, and furniture.

This dissertation includes experimental results demonstrating the real-time, mobile reconstruction of human bodies in indoor and outdoor scenes, relying only on wearable sensors embedded in commonly-worn objects and overcoming the sparse observation challenges of egocentric reconstruction.

The potential usefulness of this approach is demonstrated in a telepresence scenario featuring physical therapy training.

"To my Mom and Dad".

ACKNOWLEDGEMENTS

I would like to thank many individuals and groups for supporting and inspiring me during my Ph.D. study. First and foremost, I would like to express my deep gratitude and respect to my advisor, Prof. Henry Fuchs. I have many unforgettable moments when he helped me so that I could overcome difficulties not only in research but also in personal matters. He helped me keep motivated to be a researcher and I learned a lot from his passion and guidance in research.

I would like to thank my committee members: Dr. Gary Bishop, Dr. Jan-Michael Frahm, Dr. Shahram Izadi, and Dr. Shahriar Nirjon, for their advice and support. With their broad insights and experienced mentoring, I could go forward in a right way and earn helpful knowledge for research.

I would also like to thank my mentors. I am grateful to Dr. Adrian Ilie for always being my first reviewer and working on papers with me until the very last minute, as well as for his constant support. Andrei State was my best office mate, with unforgettable kindness, helping on papers, artistic designs, and advice. I also thank Jim Mahaney, my superhero, for his extensive assistance with the physical set-up of the prototypes and experimental captures indoors and outdoors.

Discussions with colleagues have helped inform my research. I thank former group members: Kishore Rathinavel, Rohan Chabra, Praneeth Chakravarthula, David Dunn, and Mingsong Dou.

My thanks for permission to record surgical procedure cases go to Eric Wallen, MD. I also thank collaborators from Ximmerse for the headset design. I will miss the staff members in our Computer Science Department.

I would like to thank funding and institutional support for my research by National Science Foundation (NSF) Grants CNS-1405847, IIS-1423059, IIS-1622515, IIS-1718313, and CMMI-1840131, by a grant from CISCO Systems, and by the BeingTogether Centre, a collaboration between Nanyang Technological University (NTU) Singapore and University of North Carolina (UNC) at Chapel Hill,

supported by NTU, UNC and the Singapore National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

Finally, I recognize sacrifices of my family and dedicate this dissertation to the memory of my father.

TABLE OF CONTENTS

LIST OF TABLES xii

LIST OF FIGURES xiii

LIST OF ABBREVIATIONS xvi

CHAPTER 1: INTRODUCTION 1

1.1 Motivation 1

1.2 Thesis Statement 4

1.3 Contributions 4

1.4 Organization 5

CHAPTER 2: BACKGROUND 7

2.1 Human Pose Representation 7

2.2 Kinematics 9

2.3 Parametric Body Model 11

2.4 Inertial Sensors 14

2.5 Motion Capture 17

CHAPTER 3: IMMERSIVE LEARNING EXPERIENCES FOR SURGICAL PROCEDURES 19

3.1 Introduction 20

3.2 Related Work 21

3.3 Method 22

3.3.1 Capture and Dynamic Scene Reconstruction 24

3.3.2 Scene Annotation and Playback Control 28

3.3.3 Head-Mounted Display Visualization 28
3.4 Results 29
3.5 Conclusion and Future Work 31
CHAPTER 4: MOBILE 3D RECONSTRUCTION USING ONLY HEAD-WORN CAMERAS 33
4.1 Introduction 34
4.2 Related Work 36
4.2.1 Static 3D Reconstruction 36
4.2.2 Dynamic Object Reconstruction 37
4.2.3 Dynamic Scene Reconstruction from Depth Sensors 38
4.2.4 Egocentric Motion Capture 38
4.3 System Overview 39
4.4 Mobile Headset Prototype 41
4.5 Digital Human Pre-scan 43
4.6 Video-based Body Pose Reconstruction 44
4.6.1 2D Human Body Joint Detection 44
4.6.2 3D Human Pose Sequence Estimation 47
4.6.3 CNN Training and Testing 48
4.6.4 Body Motion Re-targeting 50
4.7 Audio/Video-based Face Reconstruction 53
4.7.1 Video-based Face Reconstruction 53
4.7.2 Audio Enhancement for Face Reconstruction 57
4.7.3 Combining Video and Audio 59
4.7.4 Facial Motion Re-targeting 60
4.8 Device Tracking and Environment Reconstruction 62
4.9 Integration 63
4.10 Results 65

4.10.1 Results for Body Visibility Simulation using head-worn egocentric cameras 65
4.10.2 Results for Body Pose Estimation 67
4.10.3 Results for Face Reconstruction 70
4.10.4 Application: Virtual Tour 72
4.11 Conclusion and Future Work 73
CHAPTER 5: MOBILE HUMAN MOTION RECONSTRUCTION USING ONLY EYEGLASSES-MOUNTED CAMERAS AND A FEW BODY-WORN INERTIAL SENSORS 75
5.1 Introduction 75
5.2 Related Work 79
5.2.1 Body Reconstruction 79
5.2.2 Visual Pose Estimation 80
5.2.3 Visual Egocentric Pose Estimation 80
5.2.4 Inertial Pose Estimation 81
5.3 Wearable Capture and Egocentric Dataset 83
5.3.1 Eyeglasses and IMUs Prototype 83
5.3.2 Egocentric Visual+Inertial Human Pose Dataset 83
5.4 Egocentric Reconstruction Method 86
5.4.1 3D Body Representation 87
5.4.2 Visibility-Aware 3D Joint Detection Network 89
5.4.3 Temporally, Multi-view Consistent Joint Estimation 92
5.4.4 Visual-Inertial Alignment 94
5.4.5 Temporal Visual-Inertial Orientation Network 97
5.4.6 Deformable Body Model Fitting 100
5.5 Results and Evaluation 100
5.6 Applications 104
5.7 Conclusion and Future Work 111

CHAPTER 6: PHYSICALLY PLAUSIBLE EGOCENTRIC MOTION RECONSTRUCTION 113
6.1 Introduction . . . 113
6.2 Related Work . . . 114
6.3 Method . . . 115
6.3.1 Deformable Physics Character . . . 115
6.3.2 Physics Character Control . . . 121
6.4 Results . . . 122
6.5 Conclusion and Future Work . . . 128
CHAPTER 7: DISCUSSION AND CONCLUSION . . . 130
7.1 Summary . . . 130
7.2 Discussion . . . 131
7.3 Future Work . . . 132
7.4 Conclusion . . . 134
REFERENCES . . . 135

LIST OF TABLES

Table 4.1 – Egocentric Human Pose Dataset 65

Table 4.2 – 2D And 3D Joint Estimation Errors 69

Table 5.1 – Egocentric Visual+Inertial Human Pose Dataset (Ego-VIP dataset) 85

Table 5.2 – Online IMU Rotation Offset Calibration Algorithm 96

Table 5.3 – Performance Of Monocular HG3D 101

Table 5.4 – Quantitative Evaluation On Ego-VIP Dataset As Average Joint Position Errors . . . 102

Table 5.5 – Quantitative Evaluation On Ego-VIP dataset As Orientation Errors 102

Table 5.6 – Per-Joint Average Position Errors 103

Table 5.7 – Per-Bone Average Orientation Errors 103

Table 6.1 – Comparison Of Motion Jitter On Ego-VIP Dataset 124

Table 6.2 – Comparison Of Ground Penetration On Ego-VIP Dataset 125

LIST OF FIGURES

Figure 1.1 – Smart AR Glasses Systems 2

Figure 1.2 – Mobile Social Applications 3

Figure 2.1 – Body Representation 8

Figure 2.2 – Skeleton Representation in Kinematics 9

Figure 2.3 – Body Model Representation 12

Figure 2.4 – Motion Capture in Capture Studio 16

Figure 2.5 – Egocentric Motion Generation 16

Figure 3.1 – Immersive Experience of 3D reconstruction 20

Figure 3.2 – System Pipeline 23

Figure 3.3 – Recording configuration 24

Figure 3.4 – Dynamic scene generation 25

Figure 3.5 – Segmentation of Dynamic Elements 27

Figure 3.6 – Interacting with Immersive Reconstruction 28

Figure 3.7 – Reconstructed Immersive Environment 29

Figure 3.8 – Reconstructed Immersive Environment of Mock-up Room 30

Figure 4.1 – The Head-worn Egocentric Capture System 34

Figure 4.2 – The Eight Views From A Single Time-point Of Capture On The Prototype Device 35

Figure 4.3 – Functional Overview Of the System 40

Figure 4.4 – The Prototype Device 42

Figure 4.5 – Example Images From The Pair Of Downward-Facing Body Cameras On The Headset Device 46

Figure 4.6 – Images From The Six External Cameras And Two Top-down Body Cameras On The Headset Device 49

Figure 4.7 – Body Pose Re-Targeting 51

Figure 4.8 – Audio/Video-based Face Reconstruction Pipeline 54

Figure 4.9 – Two Video/Audio-based Fitting Results 58

Figure 4.10 – Face Re-Targeting Result 60

Figure 4.11 – Integration Result 64

Figure 4.12 – Environment And Body Part Visibility Simulation For Head-worn Egocentric Camera Modeling 66

Figure 4.13 – Example 2D And 3D Pose Estimation Results On The Validation Dataset 68

Figure 4.14 – Example 2D And 3D Pose Estimation Results For The Outdoor And Indoor Video Tour Scenes 68

Figure 4.15 – 2D Face Landmark Detection And 3D Facial Fitting 70

Figure 4.16 – Four Frames From The Indoor Section Of Virtual Tour 71

Figure 4.17 – Outdoor Virtual Tour 72

Figure 4.18 – 2D And 3D Pose Estimation Result 73

Figure 5.1 – Mobile, Egocentric Real-Time Body Motion Capture System Using Only Eyeglasses-Mounted Cameras And A Few Body-Worn Inertial Sensors ... 77

Figure 5.2 – Headset Capture Prototype 82

Figure 5.3 – Egocentric Visual+Inertial Human Pose Dataset (Ego-VIP dataset) 84

Figure 5.4 – 3D Reconstruction Pipeline 86

Figure 5.5 – Bone Representation 87

Figure 5.6 – Network Structure For The 3D Joint Detector 89

Figure 5.7 – Consistent 3D joints 93

Figure 5.8 – Coordinate Frame Transformations 94

Figure 5.9 – Temporal Visual-Inertial Orientation Network Architecture 99

Figure 5.10 – Qualitative Evaluation 1 In Ego-VIP Dataset 105
Figure 5.11 – Qualitative Evaluation 2 In Ego-VIP Dataset 106
Figure 5.12 – Qualitative Evaluation 3 In Ego-VIP Dataset 107
Figure 5.13 – Qualitative Evaluation 4 In Ego-VIP Dataset 108
Figure 5.14 – Qualitative Evaluation 5 In Ego-VIP Dataset 109
Figure 5.15 – Interactive Physical Therapy Application In VR 110
Figure 5.16 – Selected Frames In Real-Time Demo 110

Figure 6.1 – Physically Plausible Egocentric Reconstruction Pipeline 116
Figure 6.2 – Real-Time Shape Deformation Of Physics Character For Male Body Model 118
Figure 6.3 – Real-Time Shape Deformation Of Physics Character For Female Body Model . . 119
Figure 6.4 – Physics Character Overlaid With Body Model 120
Figure 6.5 – The Motions of The Physics Character Based On Rigid Body Dynamics 123
Figure 6.6 – The Motions Of The Physics Character In The Simplified Physics Environment 126
Figure 6.7 – Interaction With Objects In The Scene 127

LIST OF ABBREVIATIONS

AR	Augmented Reality
CNN	Convolutional Neural Network
EgoVIP	Egocentric Visual+Inertial Poser; the method introduced in Chapter 5
Ego-VIP dataset	Egocentric Visual+Inertial Human Pose Dataset introduced in Chapter 5
FK	Forward Kinematics
FoV	Field of View
IK	Inverse Kinematics
IMU	Inertial Measurement Unit
LBS	Linear Blend Skinning
MVS	Multi View Stereo
NLP	Natural Language Processing
PCA	Principal Component Analysis
PD-Controller	Proportional-Derivative Controller
PhysEgo	Physically Plausible Egocentric Poser; the method introduced in Chapter 6
RNN	Recurrent Neural Network
SfM	Structure from Motion
VR	Virtual Reality
VSLAM	Visual Simultaneous Localization and Mapping

CHAPTER 1: INTRODUCTION

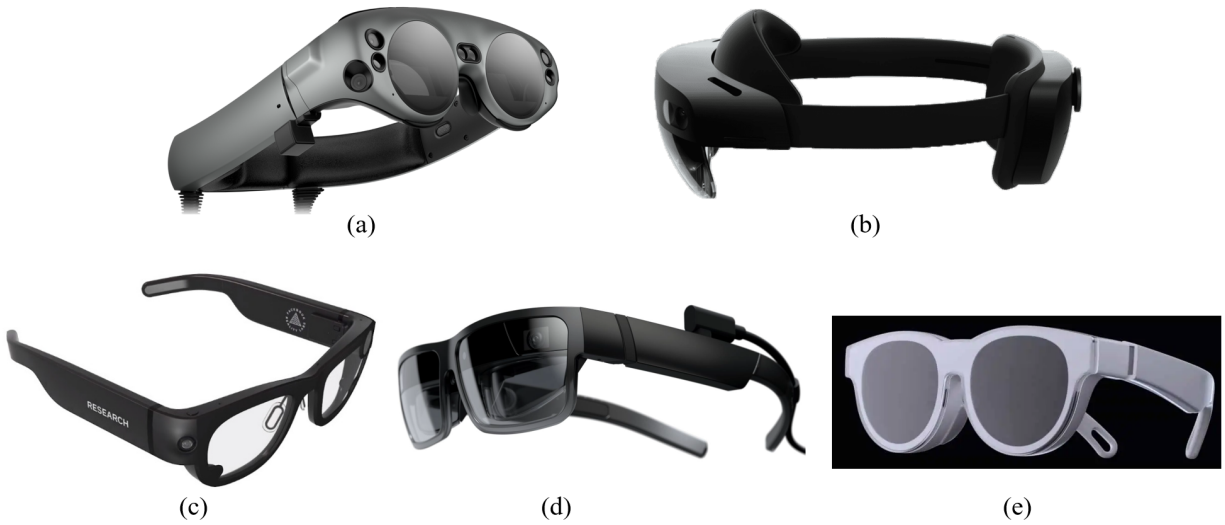
1.1 Motivation

3D Telepresence Systems: 3D telepresence systems may have an important impact in future remote social applications. Such a system consists of three main components: 3D capture, sharing 3D content, and visualization on a virtual reality (VR) / augmented reality (AR) display. The 3D capture system reconstructs 3D content such as the human subject, the environment, or objects in the scene. Depending on the capture target, different methods are employed to resolve the particular constraints of the target. Many topics are still in active or unsolved research, such as lighting, topological changes, and physical plausibility.

This dissertation introduces methods for a real-time 3D capture system for human bodies that captures 3D content anytime and anywhere, without relying on any instrumented environments. This mobile 3D capture system enables remote social interactions, which may dramatically amplify human capabilities in workflows, increasing the productivity of workers in their daily jobs by remotely participating when they are not in the same place at the same time.

Next Generation of AR Glasses: 3D capture of user experiences is likely to become a common feature of head-worn devices in the future. Today's ubiquitous mobile phones and AR systems such as the ones in the top row in Figure 1.1 may eventually evolve into the form factor of conventional eyeglasses, with transparent see-through and wide field-of-view (FoV) capabilities, to be worn all day like ordinary eyeglasses. Several companies are conducting active research to release such products in a few years, such as the ones shown in the bottom row in Figure 1.1.

[1]Magic Leap 1 (2018), Microsoft Hololens 2 (2019), Facebook Reality Labs Project Aria (2020), Lenovo ThinkReality A3 (2021), Samsung Glasses Lite Concept (2021)



(a) (b) (c) (d) (e)

Figure 1.1: Current Smart AR Glasses (top row): (a) Magic Leap 1. (b) Microsoft Hololens 2. Next Generation of Smart AR Glasses (bottom row): (c) Facebook Reality Labs Project Aria. (d) Lenovo ThinkReality A3. (e) Samsung Glasses Lite Concept. Next-generation AR glasses systems are developing toward the form factor of conventional eyeglasses to be worn like ordinary eyeglasses. [1]

Mobile Telepresence Applications: A *Real-time 3D Mobile Telepresence System* would enable shared presence and virtual touring to occur in any indoor or outdoor location, with no reliance on any instrumentation other than that in the user's worn sensors for capture. Illustrative scenarios for the benefit of 3D mobile telepresence systems are shown in Figure 1.2. Imagine a real-time 3D mobile telepresence system is capturing and sharing the user's pose. It could assist the user on a variety of tasks, such as helping a physical therapy patient with exercises. The user records the exercise motions wearing the capture device, and later, the user plays back the recorded performance through an AR display. The virtual assistant appears next to the wearer and monitors the patient's status with feedback. It could help with instant medical treatment instead of physical presence of a distant expert. In addition, more than just guiding a worker through information presented in (2D) manuals, step-by-step, the virtual presence could also monitor progress, verify the correctness of the steps, point out errors, and alert the worker when it may be appropriate to seek specialized outside help, perhaps via 3D mobile telepresence systems. The 3D mobile telepresence advanced technology, if physically unobtrusive, would enable significantly enhanced productivity for both the local worker and the remote expert.

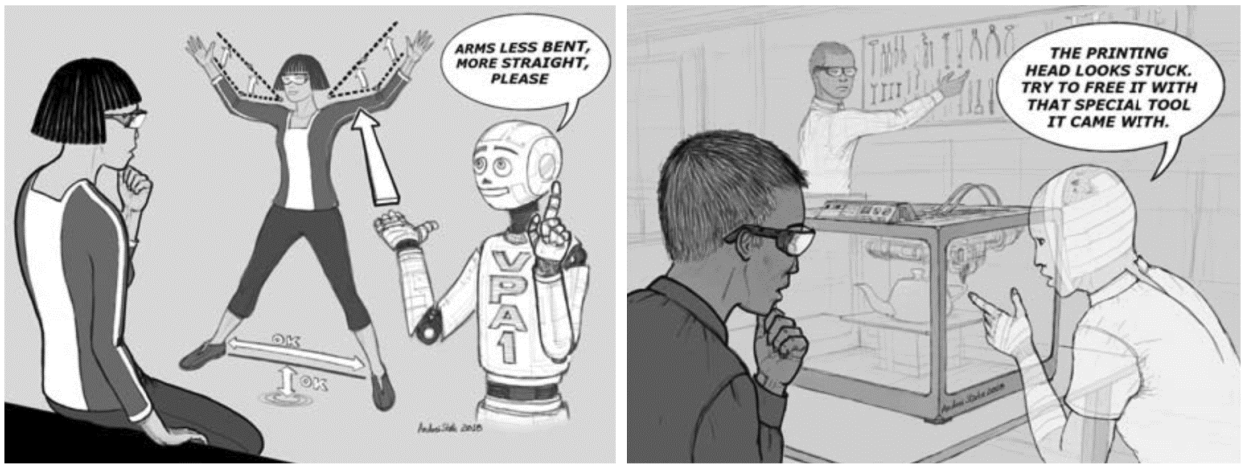


Figure 1.2: Mobile social applications: Physical therapy by virtual personal assistant (left). Virtual assistance by remote worker (Right). Drawings by Andrei State.

Requirements for Mobile Telepresence System: To realize the capabilities of real-time 3D mobile telepresence systems, the following requirements should be fulfilled and overcome the fundamental limitations of current systems: 1) The reconstruction methods must operate in arbitrary, uninstrumented environments. Current outside-looking-in camera-based methods constrain their application within a limited space, and thus are not suited in 3D mobile telepresence scenarios. 2) For personal 3D capture, self-contained AR systems to be widely accepted, displays and sensors must be unobtrusively embedded in commonly worn items such as eyeglasses, wristwatches, and shoes. It is critical that they integrate unobtrusively within an existing eyeglasses form factor to ensure everyday wearability by the user. Current motion capture systems require tens of sensors, a number unlikely to be accepted for general use, even with miniaturization.

Challenges in Egocentric Reconstruction: It is challenging to estimate the body pose from unusual near-head viewpoints. Learning-based egocentric self-capture methods that enable mobile 3D capture using only head-worn cameras have been proposed recently to overcome the problem (Cha et al., 2018; Xu et al., 2019; Tome et al., 2019). However, the fundamental problem of incomplete body visibility, that body parts are frequently occluded or outside of egocentric views, has not been yet addressed. Also, the existing head-worn prototypes are bulky and uncomfortable to wear.

Inertial Measurement Units (IMUs) are adequate as lightweight body-worn sensors, but the calibration and the measurement noise over time should be taken into account in the pose estimation. Recent IMU-based pose estimation approaches have shown promise (von Marcard et al., 2017; Huang et al., 2018) based only on sparse IMUs. However, acceptable accuracy has yet to be achieved using only insufficient sensor data.

This dissertation describes a wearable 3D acquisition system for 3D mobile telepresence to capture its user's body using only eyeglass-mounted cameras and a few IMUs, and introduces advances in egocentric reconstruction overcoming the sparse observation challenges. It also includes experimental results demonstrating the real-time, self-contained, mobile reconstruction of human bodies in indoor and outdoor scenes. The potential usefulness of the approach is demonstrated in a telepresence scenario featuring physical therapy training.

1.2 Thesis Statement

The combined use of a parametric body model, head-worn cameras, and body-worn inertial sensors enables a model-based full-body reconstruction approach only from egocentric input, providing consistent body pose and shape estimation and overcoming the challenge of sparse observations such as incomplete body visibility and insufficient sensor data.

1.3 Contributions

The results in this dissertation make several significant contributions that advance egocentric human body reconstruction for real-time mobile 3D capture systems. These contributions include:

- **Parametric-Model-Based Egocentric Reconstruction:** A method in Chapter 4 that overcomes incomplete body surface visibility from egocentric head-worn views by estimating the user's body pose and facial expression only from partial information of body parts and using the full-body estimation to re-target a high-fidelity pre-scanned model of the user. The

method demonstrates face/body/environment reconstruction indoors and outdoors only from head-worn cameras.

- **Learning-based Egocentric Visual+Inertial Human Pose Estimation:** A method in Chapter 5 that relies only on (2) unobtrusively eyeglasses-mounted cameras and a reduced number (4) of body-worn inertial sensors for widespread acceptability. It overcomes the challenges of self-occlusion, outside-of-camera motions, and non-instrumented body parts by learning the visibility-awareness of joints and the temporal correlations between instrumented and non-instrumented body parts. The approach allows for real-time ($30hz$) 3D capture of fast movements of the user indoors and outdoors.

- **Physically Plausible Egocentric Motion Reconstruction:** A method in Chapter 6, based on rigid body dynamics-based pose estimation, reduces physically implausible motion jitter and interpenetrations between the reconstructed user's model and the objects in the environment such as the ground, walls, and furniture.

Chapter 5 also describes a real-time, standalone, proof-of-concept prototype in an eyeglasses form factor for mobile 3D capture and an egocentric human motion dataset that includes multiple views with joint visibility information as well as inertial measurements. The collected egocentric human motion dataset is made publicly available to contribute to the community of learning-based egocentric reconstruction. (EgoVIP Dataset, 2021)

1.4 Organization

Chapter 2 provides an introduction to human performance capture, and Chapters 3-5 cover related work individually. Chapter 3 describes a depth camera-based, room-sized scene reconstruction of surgical procedures and introduces the inspiration for egocentric reconstruction. Chapter 4 explains the parametric model-based face, body, and environment reconstruction using only head-worn cameras. Chapter 5 describes learning-based egocentric human pose estimation using only eyeglasses-mounted cameras and sparse body-worn inertial sensors. Chapter 6 discusses

the physically plausible human motion reconstruction based on rigid body dynamics. Chapter 7 summarizes limitations and future work, ending with a conclusion.

CHAPTER 2: BACKGROUND

Given its complexity and wide range of challenges, this dissertation is related to a variety of existing research in the areas of human performance capture. This chapter briefly reviews closely-related topics in human performance capture. In Section 2.1, human pose representations are introduced. Forward and inverse kinematics are described in Section 2.2 and parametric body model representation is introduced in Section 2.3. Inertial sensors and motion capture approaches are discussed in Sections 2.4 and 2.5 respectively.

2.1 Human Pose Representation

Human pose estimation from imagery is a long-lasting active research area in Computer Vision and Computer Graphics literature. A human body pose can be represented by joint locations and their orientations. Depending on the problem, pose estimation can be divided into three categories: 2D joint location detection, 3D joint location detection, and 3D joint orientation estimation.

A joint structure is defined using a hierarchy as a rooted tree. In Figure 2.1a, joint $\#8$ (hip center) can be defined as the root among the 25 joint nodes and their parent-child relationships are denoted as lines. The state of a 3D joint can be expressed by its location and rotation in 3D space. A set of all 3D joint states indicates a particular human body pose.

Recent Convolutional Neural Network (CNN)-based approaches that detect joints from images have shown significant improvements in real-time accuracy. Such methods represent a human body pose as a set of joint locations in 2D (Wei et al., 2016; Cao et al., 2019) or in 3D (Mehta et al., 2017b, 2018). The missing joint rotations are separately estimated for a complete body pose description using external methods such as the inverse kinematics algorithms in Section 2.2. These vision-based approaches, however, suffer from occlusion of joints, resulting in significantly lower

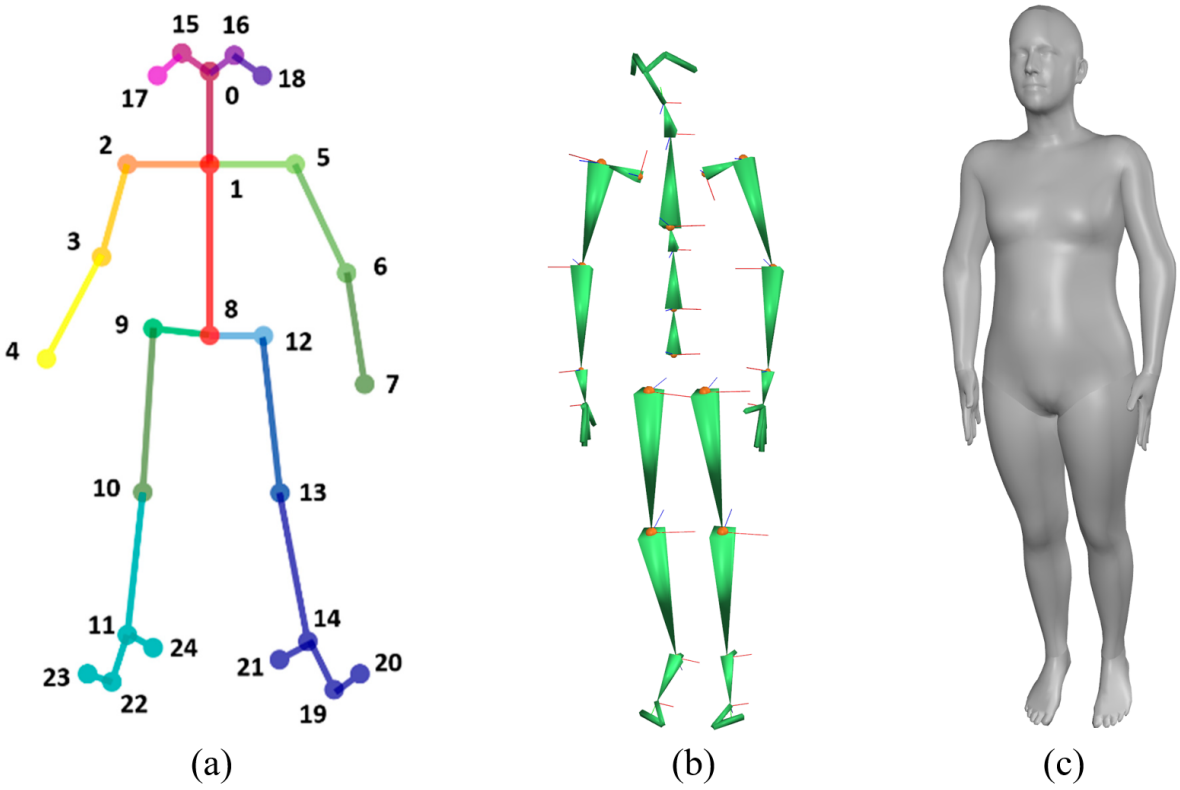


Figure 2.1: Body Representation. (a) Joints of Openpose Body25 (Cao et al., 2019). (b) Skeleton and (c) Mesh of SMPL Body Model (Loper et al., 2015).

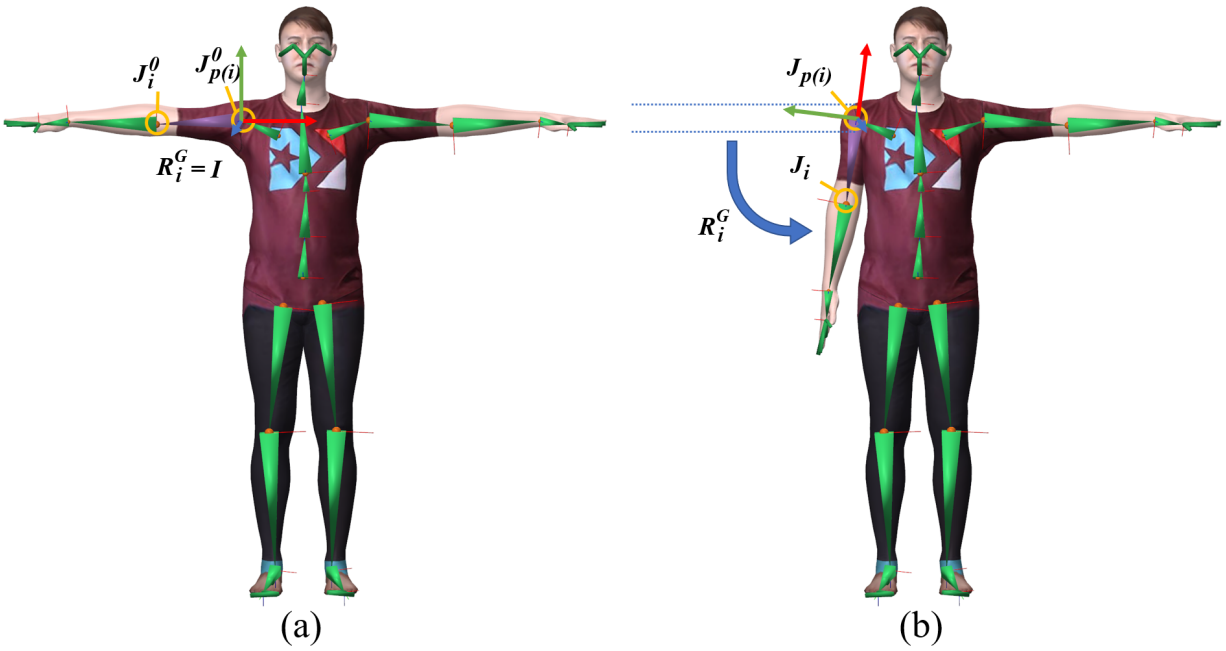


Figure 2.2: Skeleton Representation in Kinematics. (a) The skeleton in a rest pose. The bone for right upper arm B_i consists of base $J^0_{p(i)}$ (shoulder) and tip J^0_i (elbow) joints and can be represented by a joint rotation R^G_i in global space. (b) The pose after deforming the right upper arm with R^G_i. The location of tip J_i (elbow) joint is transformed according to the joint rotation R^G_i.

accuracy, and temporal inconsistency (Cheng et al., 2019). Ensuring full-body visibility is crucial in these methods for consistent human body representation over time.

2.2 Kinematics

Kinematics describes motions of a skeleton that consists of joints in an articulated rigid body. By deforming the skeleton using the joint orientation states, a human motion can be described as a sequence of skeletal deformations over time.

A skeleton is defined in advance, and consists of joint positions in a rest pose $\mathbf{J}^0$ and joint coordinate frames. The rest pose indicates all joint rotations as the identity matrix. The joint coordinate frame is also called a bind pose matrix in computer animation. An example skeleton representation is illustrated in Figure 2.2. A bone consists of base and tip joints (parent $J^0_{p(i)}$ and child J^0_i respectively) and can be represented by a rotation matrix of joint R^L_i in the local coordinate

frame and their relative distance $J_i^0 - J_{p(i)}^0$ as the local coordinate origin in the rest pose. A bone transformation in local coordinate space $B_i^L \in \mathbb{R}^{4\times4}$ can be defined as,

$$B_i^L = \begin{bmatrix} R_i^L & J_i^0 - J_{p(i)}^0 \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.1}$$

A bone transformation in global space $B_i^G \in \mathbb{R}^{4\times4}$ is defined similarly as,

$$B_i^G = \begin{bmatrix} R_i^G & J_{p(i)} \\ \mathbf{0} & 1 \end{bmatrix} \tag{2.2}$$

where R_i^G represents a joint rotation in global space and $J_{p(i)}$ is the deformed base joint location in global space.

Forward Kinematics (FK) is a process of calculating joint locations in global space $\mathbf{J}$ using the joint rotation matrices in local space $\mathbf{R}^L$. It can be expressed using B_i^L and B_i^G as,

$$B_i^G = B_{p(i)}^G \cdot B_i^L \tag{2.3}$$

Equation 2.3 is computed from the root bone to the leaf bones as they are defined in the joint structure. Depending on the problem, use of joint rotation $\mathbf{R}^L$ in local space or $\mathbf{R}^G$ in global space can be chosen. Their change of space can be similarly done with the joint hierarchy order as,

$$R_i^L = (R_{p(i)}^G)^T \cdot R_i^G \tag{2.4}$$

Traditional Inverse Kinematics (IK) algorithms estimate a set of 1D joint angles $\boldsymbol{\theta}$ to reach out the end effector (tip of the 1D joint) positions s to the target positions t. The error in position is denoted as $e = t - s$. The Jacobian matrix $\mathbb{J}$, the change in target positions w.r.t. joint angles, is defined as, (Buss, 2004)

$$\mathbb{J} = \left(\frac{\partial t_i}{\partial \theta_j}\right)_{i,j} = (r_j \times (t_i - s_{p(j)}))_{i,j} \tag{2.5}$$

where r_j is the rotational axis, $s_{p(j)}$ is the base position for joint j, and $\times$ denotes the cross product operator. The angular derivative $\dot{\boldsymbol{\theta}}$ of joints are estimated by solving the differential IK problem:

$$\dot{\boldsymbol{\theta}} = \mathbb{J}^{\#}\dot{e} \tag{2.6}$$

where $\mathbb{J}^{\#}$ is the pseudo-inverse of the Jacobian matrix. Equation 2.6 is solved iteratively until convergence.

Although traditional IK methods are efficient, they are not suitable for 3D rotational joints (spherical joints) used in most human body models. To use the IK algorithms, the 3D rotations must be transformed into ordered three Euler angles such as EulerXYZ, which is prone to suffer from the gimbal lock problem, the loss of one degree of freedom to move.

The gimbal lock problem can be avoided by restricting the Euler angle space, such as constraining $y \in [-\pi/2, \pi/2]$ in EulerXYZ. Joint angle limits can be exploited during the iteration in Equation 2.6. When the angle is approaching its limit, the update is canceled and other joints contribute more in movements toward the target positions (Drexler and Harmati, 2012). However, this results in slower convergence when the angle is close to the limit.

The IK problem can also be solved directly using 3D rotational joints. Aristidou and Lasenby (2011) estimate only desired 3D joint locations, and then the joint rotations are calculated from changes in the joint positions. This approach is more desirable for 3D rotational joints so that the joint orientations do not need to be transformed to Euler angles and thus do not exhibit the gimbal lock problem.

2.3 Parametric Body Model

A body model can describe surfaces of subjects who have different shapes and poses. A parametric body model approximates particular body surfaces by deforming them according to a set of shape and pose parameters. A human performance can be captured by estimating the parameters

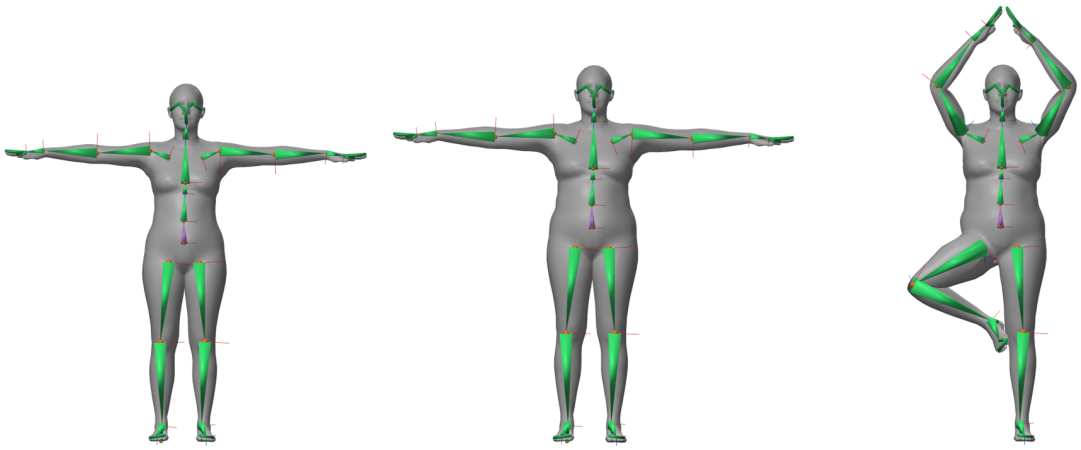


Figure 2.3: Body Model Representation. (a) The mean shape vertices $\bar{M}$ in the rest pose. (b) The shaped mesh $M_s = \bar{M} + \mathcal{B}_s(\beta)$ in the rest pose by using the linear blend shape space $\mathcal{B}_s$ specified by the shape parameters β. (c) The posed mesh $M_t = \mathcal{W}(\bar{M} + \mathcal{B}_s(\beta), \mathbf{R})$ by using the joint orientations $\mathbf{R}$ (the pose parameters) and the skinning weights $\mathcal{W}$ for Linear Blend Skinning.

of the body model. The deformed model using the estimated parameters represents the captured performance.

The SMPL body model in Loper et al. (2015) represents the body shape using $S = 10$ shape parameters β and $23 \cdot 3 + 3 = 72$ pose parameters $\boldsymbol{\gamma}$ with $K = 23$ joints and 3 parameters for the root orientation in 3D space. The triangular mesh $\mathcal{M}$ consists of $N = 6,480$ vertices and is deformed $\mathcal{M}(\beta, \boldsymbol{\gamma})$ by the specified parameters. As an example, the body skeleton and the corresponding body mesh for a male are shown in Figure 2.1b-c respectively, and the ones for a female are shown in Figure 2.3. SMPL has three different gender models; male, female, and neutral. It is assumed that the gender is known in advance.

The mean shape $\bar{M} \in \mathbb{R}^{3N}$ represents the default vertex positions in the rest pose. When the pose parameters are all zeros $\boldsymbol{\gamma} = \mathbf{0}$, it represents the rest pose (default initial pose) as shown in Figure 2.3a. The shape of the body model can be deformed by using the linear blend shape space $\mathcal{B}_s(\beta) \in \mathbb{R}^{3N \times S}$ specified by β as shown in Figure 2.3b as,

$$M_s = \bar{M} + \mathcal{B}_s(\beta) \tag{2.7}$$

where M_s represents the shaped vertices. $\mathcal{B}_s$ is the singular vectors estimated by Principal Component Analysis (PCA) to reduce the dimensionality in shape space, and thus β represents the singular values as described in (Loper et al., 2015). The shaped joint locations in rest pose $\mathbf{J}_s^0$ also can be acquired by using the joint regressor $\mathcal{J} \in \mathbb{R}^{3K \times 3N}$ as,

$$\mathbf{J}_s^0 = \mathcal{J}(M_s) \tag{2.8}$$

The pose parameter $\gamma \in \boldsymbol{\gamma}$ is represented as a rotation vector in local space converted from an angle-axis representation as $\gamma = \theta \cdot u$, where θ is an angle and u is an axis. Depending on the problem, rotation matrices can be used instead, which can be computed from the rotation vectors. A cross-product matrix U can be defined from the axis $u = (u_x, u_y, u_z)$ as,

$$U = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \tag{2.9}$$

Using Rodrigues' rotation formula, the angle-axis $\gamma = \theta \cdot u$ can be transformed to the rotation matrix R as,

$$R = I + sin(\theta)U + (1 - cos(\theta))U^2 \tag{2.10}$$

where I is the identity.

Using the shaped joint locations in rest pose $\mathbf{J}_s^0$ and the joint rotation matrices $\mathbf{R}^L$ in local space from the pose rotation vectors $\boldsymbol{\gamma}$, the forward kinematics of a SMPL skeleton $\mathbf{B}^G = FK(\mathbf{J}_s^0, \mathbf{R}^L)$ can be performed using Equation 2.1, Equation 2.2, and Equation 2.3 in Section 2.2.

The shaped mesh M_s can be deformed using joint orientations $\mathbf{R}^G$ (in $\mathbf{B}^G$) in global space in Equation 2.2 and the skinning weights $\mathcal{W} \in \mathbb{R}^{N \times K}$ as shown in Figure 2.3c as,

$$M_t = \mathcal{W}(M_s, \mathbf{R}^G) \tag{2.11}$$

where M_t represents the posed mesh. Linear Blend Skinning (LBS) for each vertex $v_i^s \in M_s$ is performed for the deformation $\mathbf{R}^G$ as,

$$v_i^t = \sum_j \mathcal{W}_{i,j} \cdot R_j^G \cdot v_i^s \tag{2.12}$$

where $v_i^t \in M_t$. The skinning weight matrix $\mathcal{W}$ is sparse, so real-time performance is achieved by rearranging the weights in descending order for each vertex and using at most n weights. In this dissertation, $n = 4$ is used.

Use of pose-dependent blend shapes further deforms the body shape with respect to the pose parameters. In this dissertation, the pose blend shape terms in the original paper (Loper et al., 2015) are left out to maintain real-time performance.

Using the body model, the joint positions in rest pose $\mathbf{J}^0$ in Section 2.2 refer to the shaped joint locations $\mathbf{J}_s^0$ as,

$$\mathbf{J}_s^0(\beta) = \mathcal{J}(\bar{M} + \mathcal{B}_s(\beta)) \tag{2.13}$$

In this dissertation, a joint orientation refers to a rotation matrix in local joint coordinate frame $R^L \in \mathbf{R}^L$ in Equation 2.1, instead of an angle-axis representation. The parametric body model deformation is summarized as,

$$M(\beta, FK(\mathbf{R}^L)) = \mathcal{W}(\bar{M} + \mathcal{B}_s(\beta), \mathbf{R}^G) \tag{2.14}$$

2.4 Inertial Sensors

Inertial measurement units (IMUs) can be used for human pose estimation by just wearing them on particular parts of the body. The sensors are rigidly attached and moving along with the body parts that move, which enables fast motions to be captured. As an example, the inertial sensors worn on the parts of the body are shown in Figure 2.4a-b. The 8 sensors are worn on the upper and

lower bones of both arms and legs. In this subsection, the problems of inertial sensors and sensor calibration methods are discussed.

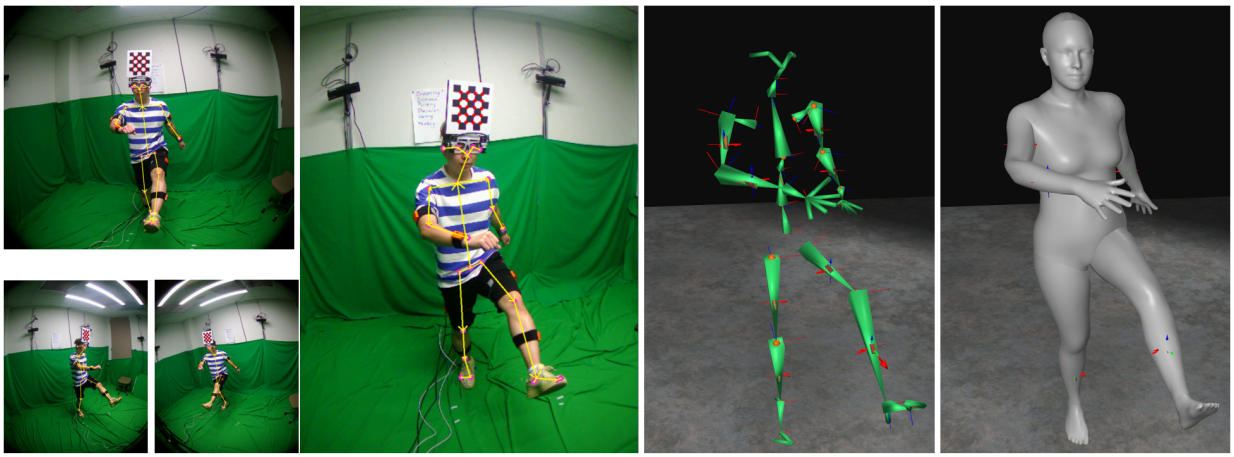
IMU devices are equipped with gyroscopes, accelerometers, and magnetometers, which can measure 3D orientations and 3D accelerations over time at a high frame rate (von Marcard et al., 2017). The output measurements are internally filtered using a built-in Kalman Filter and the measurements in device coordinates can be sent to a master PC wirelessly (Xsens Mtw Awinda, 2015). Depending on the device specification, the orientation can be represented by rotation vectors, Euler angles, or quaternions. It is assumed that the 3D orientation output is already converted to a rotation matrix.

Although IMUs are convenient to use, the sensor coordinate system can be easily disturbed. The device coordinate system is defined by the up direction measured by the accelerometer and the north direction measured by the magnetometers. The third axis is defined by the cross product of the up and the north directions. The measured north direction is inaccurate especially indoors since any metal objects nearby disturb the magnetometers and thus the measured north direction can change over time. This variability causes the sensor measurements to drift over time. Also, the orientations and accelerations measured by the gyroscope and the accelerometer respectively are noisy even after Kalman filtering. These sensor noise and drift problems should be taken into account when the measurements are used for pose estimation.

IMUs need to be calibrated in the beginning and adjusted at run-time as well to maintain consistent measurements over time. The sensors need to be stabilized in the beginning, so recordings start with a designated stationary pose for a few seconds. When using multiple sensors, the measured north directions for each device are not identical due to noise, so heading reset is performed to cancel out the noisy north directions as,

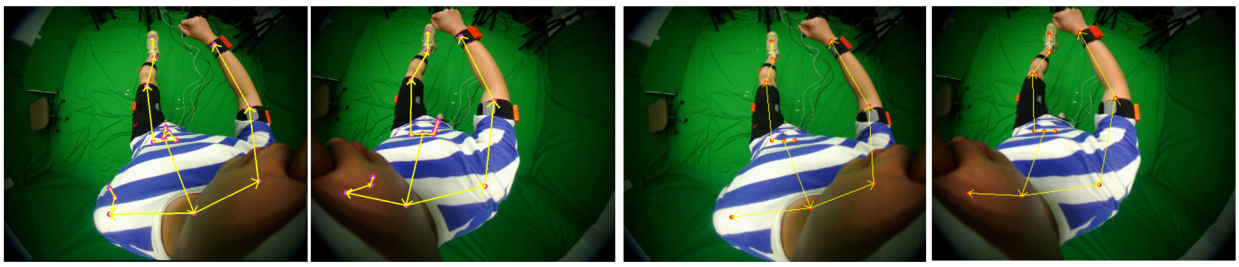
$$R_t = (R_0)^T \cdot \tilde{R}_t \tag{2.15}$$

where $\tilde{R}_t$ is the measured raw orientation at time t. R_0 is the averaged rotation during the calibration step in the beginning. The heading reset for acceleration is defined similarly as,



(a) (b) (c) (d)

Figure 2.4: Motion Capture in Capture Studio. The human pose is estimated from both 4 external cameras and 8 body-worn inertial sensors. (a,b) Fixed external camera views overlaid with the estimated joints. Deformed (c) skeleton and (d) mesh of SMPL Body Model (Loper et al., 2015) using the estimated pose.



(a) (b) (c) (d)

Figure 2.5: Egocentric Motion Generation. (a,b) Joints transferred from the pose estimation using both external cameras and inertial sensors in Figure 2.4, overlaid onto head-worn camera views. (c,d) Egocentric joints with visibility applied, invisible arm and leg joints are removed.

$$a_t = (R_0)^T \cdot \tilde{a}_t \tag{2.16}$$

where $\tilde{a}_t$ and is the raw acceleration at time t. In this dissertation, the orientation and the acceleration measurements at t are referred to R_t and a_t respectively, with the heading reset already applied.

Since the initial pose for the beginning calibration is known, the inertial sensor measurements can be maintained in the global coordinate space as,

$$R_t^G = R_t \cdot R^E \tag{2.17}$$

$$a_t^G = R^E \cdot a_t \tag{2.18}$$

where R^E is the given external rotation for sensor coordinate transform into the global coordinate space. A sensor calibration in the beginning refers to both the heading reset in Equation 2.15 and the coordinate system transform in Equation 2.17. A run-time sensor adjustment method will be discussed in subsection 5.4.4.

2.5 Motion Capture

A motion capture system records a sequence of poses of the subject over time. The pose can be measured using cameras, inertial sensors, markers, and so on. The captured motions can be directly used for character animation or as a dataset for other applications such as training a neural network. This subsection introduces the human motion capture using multiple cameras and IMUs used in Chapters 5-7.

The human motion capture in a capture studio for generating an egocentric human pose dataset is shown in Figure 2.4 and Figure 2.5. There are 4 fixed outside-looking-in cameras. The user is wearing 8 inertial sensors to measure orientations of upper and lower limb motions. The user is also wearing a headset equipped with 2 downward-looking body cameras for capturing egocentric

images, and a rigidly attached checkerboard to aid in conversions between the various coordinate systems.

The 3D joint locations are detected using the 4 fixed external cameras that are calibrated in advance. 2D joint locations are detected using the method in Cao et al. (2019), followed by triangulation of the 2D joints in the 4 external camera images using the camera calibration matrices. The estimated 3D joint locations are shown in Figure 2.4a-b by projecting the joints onto each external camera image.

The limb bone orientations are measured by the 8 inertial sensors. Other non-instrumented bone orientations are estimated using the IK algorithm from Aristidou and Lasenby (2011). The estimated joint locations and orientations form a complete body pose representation.

The shape parameters of SMPL body model are estimated using Equation 5.5. Using the estimated shape parameters and the joint orientations, the SMPL body model can be deformed using Equation 2.7 and Equation 2.11. The deformed skeleton and the mesh are shown in Figure 2.4 c-d.

The captured pose is used to generate a pose in the downward egocentric camera space. The 3D pose of the head-worn checkerboard is estimated by the detection in the four external cameras. Since the downward cameras and the checkerboard are pre-calibrated, the head-worn camera poses in global space are estimated using the checkerboard pose. Using the body camera poses in global space, the 3D joint locations can be transformed into the egocentric body camera spaces. The estimated joints in the egocentric spaces are shown in Figure 2.5 a-b. Unlike the external camera views, some body parts can be occluded or outside of the egocentric camera FoVs. These invisible joints are excluded from the transferred joints, as shown in Figure 2.5c-d.

The large-scale recordings of the visibility-applied egocentric joints and the corresponding inertial sensor measurements form the egocentric motion dataset for training and evaluating the method in Chapter 5.

CHAPTER 3: IMMERSIVE LEARNING EXPERIENCES FOR SURGICAL PROCEDURES

This chapter introduces a system for creating immersive learning environments for surgical procedures by applying depth camera-based, room-sized 3D capture and dynamic reconstruction methods to reconstruct the actions and events during the procedure. The reconstruction can be annotated in space and time to provide more information about the scene to users. The resulting 3D-plus-time reconstruction can be immersively experienced later; equipped with a VR display, a user can walk around the reconstruction of the procedure room while controlling the playback of the recorded surgical procedure. Experimental results demonstrate the potential usefulness of the system in applications such as training medical students and nurses. This chapter also introduces the inspiration for egocentric reconstruction described in the following chapters.

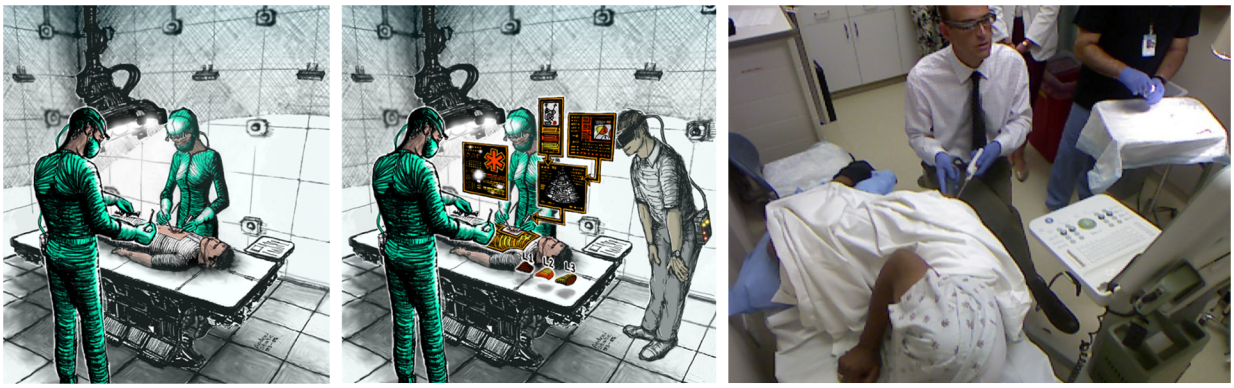
This chapter is mainly based on "Immersive Learning Experiences for Surgical Procedures", Young-Woon Cha, Mingsong Dou, Rohan Chabra, Federico Menozzi, Andrei State, Eric Wallen, MD, and Henry Fuchs, published in Studies in health technology and informatics, Proceedings of Medicine Meets Virtual Reality / NextMed (MMVR), April, 2016. [1]

This chapter is also partially based on "Optimizing Placement of Commodity Depth Cameras for Known 3D Dynamic Scene Capture", Rohan Chabra, Adrian Ilie, Nicholas Rewkowski, Young-Woon Cha, and Henry Fuchs, published in IEEE Virtual Reality (VR), March, 2017. [2]

The author contributed to the teamwork in Chabra et al. (2017) for the mock-up recording and its reconstruction shown in Section 3.4.

[1] Cha et al. (2016)

[2] Chabra et al. (2017)



(a) (b) (c)

Figure 3.1: (a) 3D capture during medical procedure. (b) Immersive experience of the 3D reconstruction. Drawings by Andrei State. (c) Prostate biopsy procedure.

3.1 Introduction

This chapter introduces a system for 3D-plus-time recording of activities, such as surgical procedures, through the room-sized 3D capture and reconstruction methods, for applications such as immersive environments for medical training. In the initial prototype system shown here, a prostate biopsy procedure is captured at a UNC Urology clinic (Figure 3.1). The system performs dynamic reconstruction for all persons present: a patient, a physician, a nurse assistant, and an observer. The small procedure room was instrumented with three Kinect color+depth cameras in three of its corners. Because of the setup's limited coverage and the frequent occlusion events caused by the participants, the reconstruction results contain spatial gaps and other inaccuracies.

Yet despite its shortcomings, compared with being physically present at the procedure, the virtual presence provided by the prototype system has several advantages: a student experiencing the immersive reconstruction can freely move to any desired viewing location, including locations that might have interfered with the procedure as it was being executed; the reconstruction can be annotated in space and time with information that facilitates insight and accumulation of knowledge–annotations can be added post-reconstruction by the physician who performed the procedure, or by other competent personnel; finally, the student may pause, rewind, or temporally

scan through the procedure at variable speed forward or backward in time, or even ”single-step” through it.

The remainder of this chapter is organized as follows. After reviewing relevant previous work in Section 3.2, the introduced framework is detailed in Section 3.3, which includes descriptions of the dynamic scene reconstruction, annotation, playback, and visualization based on a head-mounted display (HMD). The experimental results are discussed in Section 3.4. This chapter concludes and summarizes possible improvements in Section 3.5.

3.2 Related Work

The modern medical simulator systems in Parvati et al. (2011); Alexandrova et al. (2012); Liu (2014) investigate animatable simulators on immersive virtual environments to provide better understanding to users using 3D visualizations than fixed 2D video streams. Limited perspectives are provided to users for immersive visualization. The studies by Ebert et al. (2014); Ferracani et al. (2014); Lin et al. (2013) investigate walk-around VR systems using HMDs, though these approaches still use predefined meshes or manually reconstruct virtual scenes using 3D graphics tools for real-world scenarios.

The immersive environments can be generated directly from recorded images using 3D reconstruction methods for more realistic visualizations. In a controlled setting, the approach by Welch et al. (2005) proposes the 3D reconstruction of environments from images with HMD visualization. The system by Kurillo et al. (2009) shows reconstruction of objects combined with predefined environments using real-time stereo matching. In the system shown here, the entire immersive environment is fully reconstructed from captured images.

The immersive 3D virtual reality (VR) system introduced here is similar to previously described telepresence systems such as the one described in Fuchs et al. (2014), and enables users to experience immersive 3D environments through a combination of 3D scanning and immersive display. 3D scanning methods for dynamic scenes such as the one by Dou and Fuchs (2014) reconstruct a sequence of surfaces by updating changes in the scene over time.

Geometric change detection methods such as the ones in Taneja et al. (2011); Ulusoy and Mundy (2014) estimate the changed areas in the scene by modeling static backgrounds. The dynamic 3D scene is updated by re-scanning the changing regions while leaving other regions untouched.

Recent work shows that using AR systems can also enhance surgical procedures with 3D tracking and display technologies. Rose et al. (2019) introduce an egocentric AR system-based method for surgical procedure training. Using the marker-based tracking system from an AR headgear (Microsoft HoloLens 1, 2016) worn by the user, the user is able to perform a simulated surgical task based on the guidance provided by the AR platform. Desselle et al. (2020) show the usefulness of using an AR headset-based system that directly overlays 3D imagery on the physical procedure scenes instead of 2D computer displays, resulting in effective aids for surgeons during the procedure.

3.3 Method

In this section, the approach for generating the immersive learning environments is described in detail. The system pipeline is illustrated in Figure 3.2. First, synchronized multiple-viewpoint RGB-depth image sequences are captured during the procedure using Microsoft Kinect depth cameras. Second, the entire procedure is reconstructed as a sequence of 3D surface meshes over time, using the method described in Dou and Fuchs (2014). Third, the sequence of 3D surfaces can be manually annotated by adding timed 3D text labels in appropriate locations to describe and explain the activities. After this processing, a user wearing a tracked HMD can examine the reconstructed, annotated immersive environment at leisure and repeatedly, as described above. By updating the eye positions provided by the HMD tracker in real-time, the visualization subsystem presents a walkable, immersive environment from the user's perspective. The user controls the playback of the reconstruction with a remote hand-held controller such as a wireless mouse. In the following subsections, each step is elaborated.

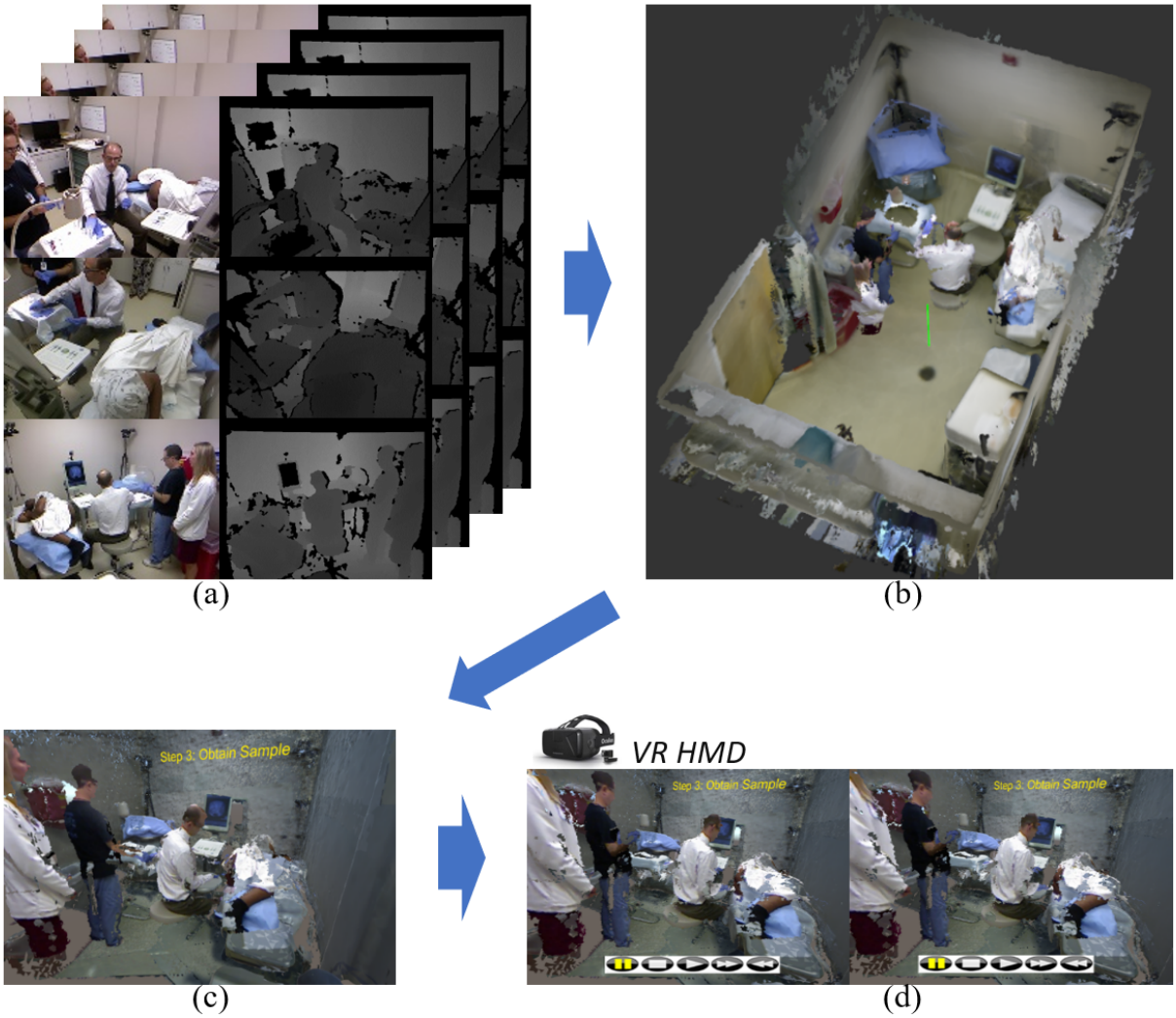


Figure 3.2: System Pipeline: (a) Recording: Multiple-view RGB+depth image sequences are captured during the procedure. (b) 3D reconstruction: The 3D scene is reconstructed using the sequences. (c) Annotation: The reconstruction is annotated with 3D text for playback. (d) Immersive Experience: A user walks around the reconstruction using a head-mounted display.

Figure 3.3: Recording configuration. (a): reconstructed 3D procedure room using depth images from a single moving hand-held camera. (b) and (c): fixed wall-mounted Kinect depth cameras that capture moving objects during the procedure.

3.3.1 Capture and Dynamic Scene Reconstruction

This subsection describes how surgical procedure scenes are reconstructed as a sequence of surface meshes: $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_T\}$. To capture dynamically changing indoor environments over time, the static background (e.g., the room where the procedure takes place) is captured in advance, and dynamically changing objects are separately acquired to handle changes in the surface mesh (Dou and Fuchs, 2014).

The static background, denoted as $\mathbf{M}_0$, is pre-scanned with a single moving camera (Figure 3.3a). An extended version of KinectFusion (Newcombe et al., 2011) is utilized for a room-sized scene reconstruction that incorporates plane matching to improve reconstructions of features in walls, ceiling, and floor (Dou et al., 2012).

The moving objects (typically, people and instruments) are captured over time by fixed depth cameras mounted in the corners of the room (Figure 3.3). The depth cameras are pre-calibrated to a global coordinate system (Dou and Fuchs, 2014); One of the cameras, C^1, is located at the origin $[\mathbf{I}_{3\times3}|\mathbf{0}_{3\times1}]$ of the global coordinate system, and other cameras, C^i, are at their respective poses $[\mathbf{R}^i_{3\times3}|\mathbf{T}^i_{3\times1}]$ relative to C^1. Let $\mathbf{V}_t = \mathbf{V}^1_t \cup ... \cup \mathbf{V}^N_t$ be a set of colored 3D vertices extracted from RGB-depth images of the N calibrated and synchronized cameras at time t.

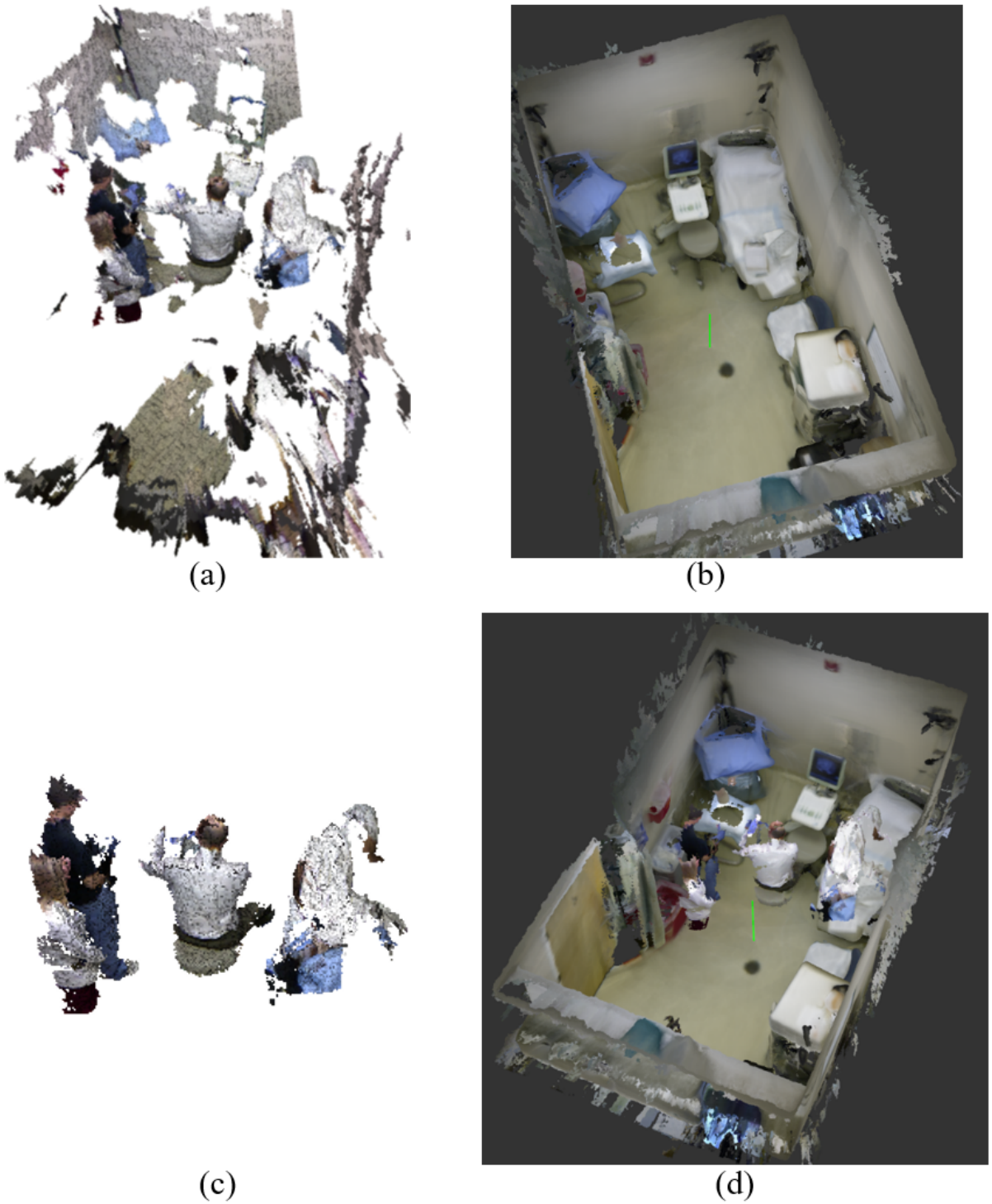


Figure 3.4: Dynamic scene generation. (a): 3D surface meshes from three Kinects. (b): Pre-scanned 3D surface mesh of procedure room. (c): Segmented surface mesh from (a). (d): Combined mesh consisting of (b) and (c).

The pre-scan $\mathbf{M}_0$ of the static background (Figure 3.4b) is also aligned to the global coordinates of the camera cluster. To achieve that, the pose estimation based on SIFT feature matching is initially employed; then the alignment is refined through ICP registration between $\mathbf{M}_0$ and the initially (at time step 0) acquired live geometry set $\mathbf{V}_0$ (Dou and Fuchs, 2014).

At each subsequent time step, the acquired live geometry set $\mathbf{V}_t$ as shown in Figure 3.4a is segmented to detect foreground (i.e., non-background) data $\mathbf{F}_t \equiv \{v|v \in \mathbf{V}_t \text{ and } v \notin \mathbf{M}_0\}$ (Figure 3.4c) by comparing $\mathbf{V}_t$ (Figure 3.4a) with $\mathbf{M}_0$ (Figure 3.4b). The reconstructed surface mesh at frame t is defined as $\mathbf{M}_t \equiv \mathbf{F}_t \cup \mathbf{M}_0$ (Figure 3.4d).

The foreground vertices $\mathbf{F}_t^i$ at each camera i are estimated from $\mathbf{V}_t^i$ via superpixel-based background subtraction. Figure 3.5 shows an example of such foreground segmentation. The static background model $\mathbf{B}_0^i$ at each camera i is estimated from a set of depth images captured just before the procedure (Figure 3.5b). The $\mathbf{V}_t^i$ is labeled as $l(v \in \mathbf{V}_t) \in \{0 = background, 1 = foreground\}$ by subtracting $\mathbf{B}_0^i$ from the depth image $\mathbf{D}_t^i$. The color image $\mathbf{I}_t^i$ is segmented as a set of superpixels $\mathbf{S}$ using SLIC (Achanta et al., 2012) by merging local pixels based on the color similarity. (Figure 3.5e). The superpixel $S_i \in \mathbf{S}$ includes a set of vertices $v_s \in S_i$, and is labeled as $l(S_i)$ by voting $l(v_s)$. Superpixel-level connected components are extracted based on the similarity of depth values between adjacent superpixels (Figure 3.5e-f). The foreground vertices $\mathbf{F}_t^i = \{v|v \in \mathbf{V}_t^i, v \in S_i \text{ and } l(S_i) = 1\}$.

The $\mathbf{F}_t = \mathbf{F}_t^1 \cup ... \cup \mathbf{F}_t^N$ represent the colored 3D points that differ from the static background mesh $\mathbf{M}_0$. The $\mathbf{F}_t$ are meshed using the marching cubes algorithm (Lorensen and Cline, 1987) followed by a volumetric fusion pass (Curless and Levoy, 1996). This forms the dynamic surface at time t as shown in Figure 3.4c. The complete 3D surface mesh becomes $\mathbf{M}_t = \mathbf{F}_t \cup \mathbf{M}_0$ shown in Figure 3.4d. At visualization time, the sequence of dynamic 3D surface meshes $\mathbf{M} = \{\mathbf{M}_1, ..., \mathbf{M}_T\}$ are rendered to the user's HMD in real-time.

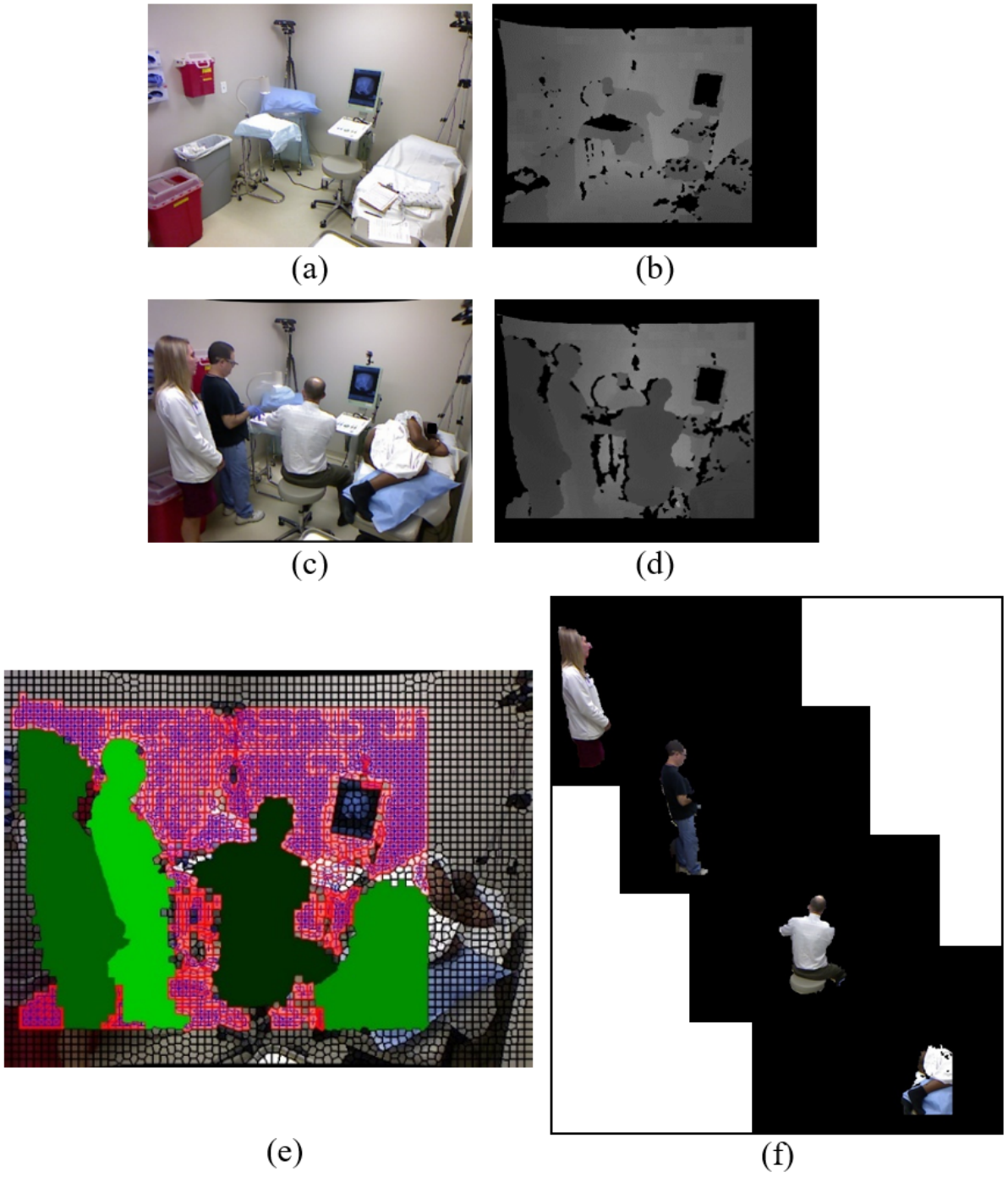


Figure 3.5: Segmentation of dynamic elements. (a) and (b) are a pair of color and depth images of the empty procedure room. (c) and (d) show the RGB-depth image at time t. From (b) and (d), changed parts (green) are segmented from background (purple) using superpixel-based foreground detection in (e). (f) shows the separated segments in (e).

(a) (b) (c)

Figure 3.6: Interacting with the immersive reconstruction. (a): The user examines the reconstruction through a head-mounted display. The user controls the playback of the scene with a wireless hand-held controller. (b) and (c): User's views at two different times. The VCR controls are shown at the bottom. Annotations are visible as yellow text on the wall.

3.3.2 Scene Annotation and Playback Control

In addition to the sequence of dynamic 3D surface meshes $\mathbf{M}$, the user is able to view additional descriptions about the scene and to control the playback of the sequence as mentioned.

To insert the annotations, a subset of the frames $\mathbf{M}_s \equiv \{\mathbf{M}_{t_1}, ..., \mathbf{M}_{t_2}\}$ where $t_1 < t_2$ are manually enhanced with 3D text labels placed in specific locations in M_s. An example of such annotation is shown in Figure 3.6. The text in this example is positioned on the wall in 3D space and provides information about the surgery step occurring during this period.

During playback, the user can quickly move to a specific time period in the recording using the virtual playback controller and a wireless mouse (Figure 3.6b and Figure 3.6c). The controller includes play, pause, stop, fast forward, and rewind buttons.

3.3.3 Head-Mounted Display Visualization

Figure 3.7 shows a user walking through the reconstructed procedure room. Using the 3D positions of the user's eyes and the HMD viewing direction supplied by the HMD tracker, the immersive, annotated environment is rendered stereoscopically, distortion-corrected and displayed in the user's HMD.

To enable the user to walk along the floor in the reconstructed procedure room as he or she walks in the real world, the coordinates between the reconstructed scene and of the HMD tracker

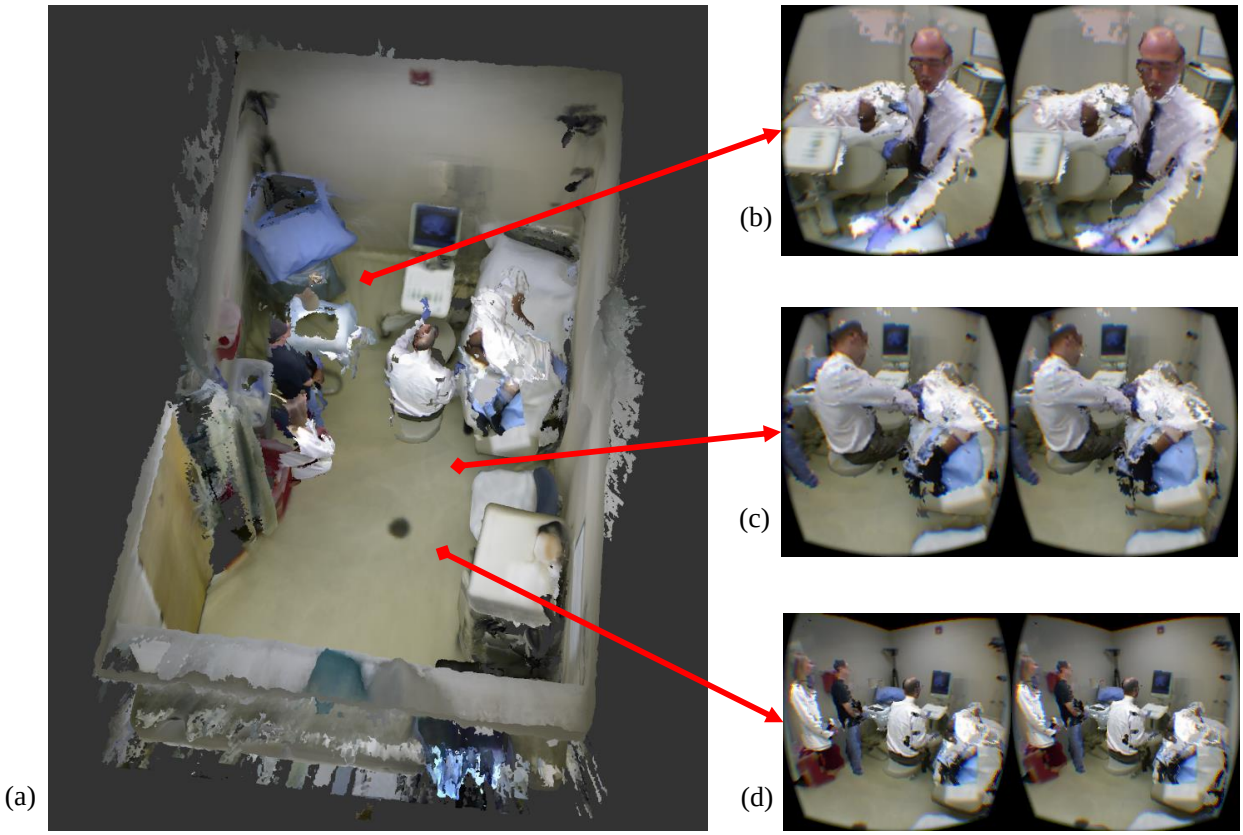


Figure 3.7: Reconstructed immersive environment. (a): top view with sample user locations (red). (b-d): corresponding views inside HMD. (These are "screen shots" provided by the Oculus SDK, approximations to the images sent to the HMD screen.)

must be aligned. To accomplish this, the reconstructed mesh is manually transformed to align the floor plane ($Y = 0$) in the mesh with the floor plane ($Y = \alpha$) in the user's room.

The mesh is transformed in advance so that the floor in the mesh is located at $Y = 0$ in the coordinate. The XZ plane of the HMD tracker is aligned with the ground regardless of camera orientation. The floor planes in the reconstruction and in the real world are aligned by manually adjusting the Y coordinate of the mesh.

3.4 Results

In the recording, four people were present during the procedure shown: a patient, a physician, a nurse assistant, and an observer. To record the scene, four calibrated Microsoft Kinect depth cameras were used; one mobile unit for the pre-scan of the static background, and three fixed wall-mounted

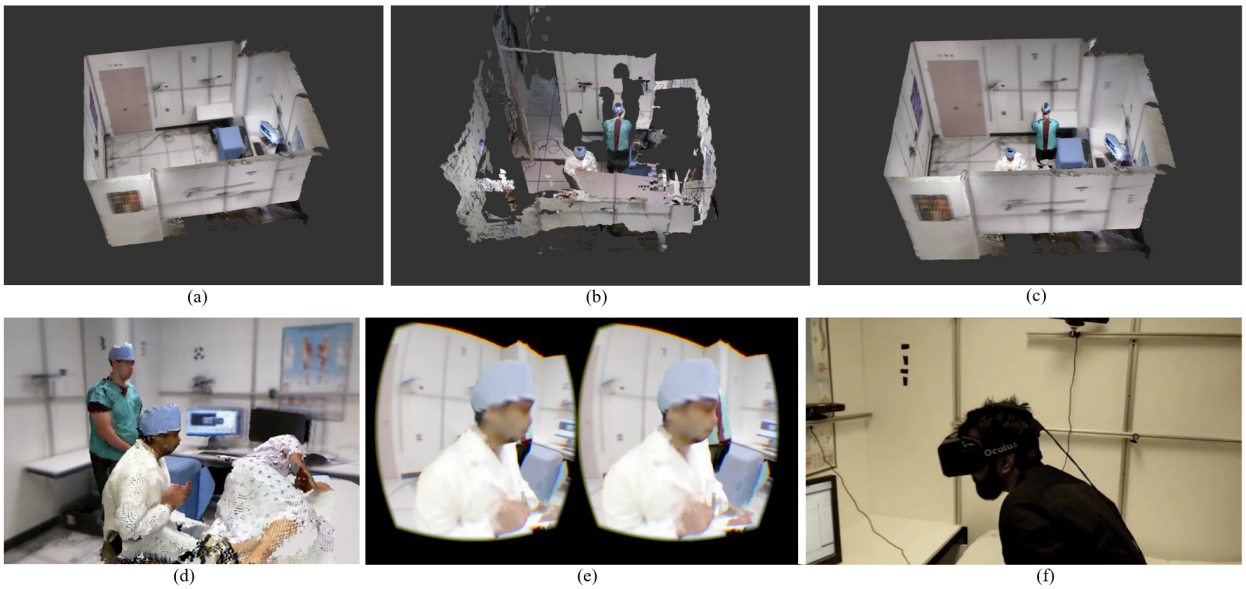


Figure 3.8: Reconstructed immersive environment of mock-up room. (a): Pre-scanned 3D reconstruction of mock-up room. (b): 3D surface meshes from nine Kinects at time t. (c): Combined reconstruction consisting of (a) and (b). (d): Front view of (c). (e): Stereo views of (d) inside HMD. (f): External view of the user.

units for dynamic procedure capture (Dou and Fuchs, 2014). The multiple Kinect recording setup provided approximately 25 color and depth images per second at 640×480 resolution. The three fixed depth cameras were synchronized manually. The reconstruction and visualization systems used Intel Xeon E5-2630V3 Octa-core 2.4GHz with 64GB memory.

The pre-scanned room shown in Figure 3.3a was reconstructed from 401 RGB-depth images captured by a single hand-held Kinect depth camera. The dimensions of the room are approximately $2.5m \times 4.5m \times 3m$ (width, length, and height). In this first experiment, $1,841$ consecutive multiple-view RGB-depth images were sampled, equivalent to a playback running time of approximately 1.5 minutes. At viewing time, an Oculus Rift DK2 HMD was used and the scene is rendered using the Unity 5 Integration provided by Oculus VR.

Figure 3.7 demonstrates the walkaround capability within the reconstructed immersive environment. Figure 3.7a shows the reconstructed room (including dynamic objects such as people and instruments), which the user can observe from his own, freely selectable position. Three sample views are depicted in Figure 3.7b-d and show the distortion-compensated imagery presented within the Oculus HMD.

The user is able to direct his/her gaze at and approach any spatial regions of the scene he/she is interested in, without restrictions.

Figure 3.6 illustrates the playback functions while walking around the reconstruction. The user holds a wireless mouse and can click the buttons on the virtual playback controller. This feature makes it easy to find and replay the interesting time snippets.

To improve the reconstruction quality, a mock-up room of similar dimensions to the operating room was set up in our lab space, with more number of (9) depth cameras (Chabra et al., 2017). The reconstructions in the mock-up room are shown in Figure 3.8. Using 9 depth cameras in the mock-up room significantly improves the reconstruction compared to using only 3 depth cameras in the procedure room. However, some holes and gaps still remain in the reconstruction of dynamic objects due to occlusions by the participants as shown in Figure 3.8b.

3.5 Conclusion and Future Work

This chapter introduced a system for creating immersive learning environments for surgical procedures by applying 3D capture and dynamic reconstruction methods to such procedures. The resulting dynamic geometry can be annotated post-reconstruction to enhance educational utility. I expect that such immersively experienced, annotated procedures can be useful for beginning medical students and nurses in particular, as it can supplement preparation for their initial patient treatment encounters. In the long term, ubiquitous deployment and continuous operation of such acquisition and reconstruction technology can help to make it possible to re-experience difficult or unusual cases, helping medical personnel develop skills for interventions that occur infrequently.

The main limitation of the system is the spatial gaps caused by the limited coverage and the frequent occlusions by the participants. To improve the surface quality of the dynamic scene elements, non-rigid registration methods by Dou et al. (2015); Zollhöfer et al. (2014); Newcombe et al. (2015) can be utilized to continually track and integrate the surfaces of moving objects. Incorporating color-based multi-view segmentation can also help improve the surface quality in dynamic scene reconstruction (Djelouah et al., 2013). To improve the overall system, deploying a

larger number of cameras, including higher resolution cameras, can help reduce artifacts caused by occlusion or reconstruction failures. However, this high-cost instrumentation extension still does not guarantee solving the inherent problem of random occlusions.

From this work, I realized that the contributions from cameras and/or depth scanners worn by the attending personnel could observe the most important parts of the reconstructed geometry, since they represent the focus of attention of the medical personnel at the time of the procedure. This observation inspired the use of egocentric, head-mounted cameras, worn by the participants. The following chapter will discuss the head-worn camera based 3D capture system and the egocentric reconstruction methods.

CHAPTER 4: MOBILE 3D RECONSTRUCTION USING ONLY HEAD-WORN CAMERAS

This chapter presents a parametric model-based face, body, and environment reconstruction method that does not rely on any instrumented environment but only on head-worn cameras worn by the user for future fully mobile 3D capture systems. This method overcomes incomplete body surface visibility from egocentric head-worn views by estimating the user's body pose and facial expression only from partial information of body parts and uses the full-body estimation to re-target a high-fidelity pre-scanned model of the user. The experimental results demonstrate that the self-sufficient, head-worn capture system in this chapter is capable of reconstructing the wearer's movements and their surrounding environment in both indoor and outdoor situations without any additional views.

This chapter is primarily based on "Towards Fully Mobile 3D Face, Body, and Environment Capture Using Only Head-worn Cameras", Young-Woon Cha[§], True Price[§], Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, Zihe Qin, Hyounghun Kim, Zhaoqi Su, Yebin Liu, Adrian Ilie, Andrei State, Zhenlin Xu, Jan-Michael Frahm, and Henry Fuchs, published in IEEE Transactions on Visualization and Computer Graphics (TVCG), Vol. 24, November, 2018 (Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR) 2018). [1] [2]

Sections 4.6.1-3, 4.7.1-3, and 4.8 are mainly contributed by the coauthors. Other sections are mostly contributed by the author.

---

[1] (Cha et al., 2018)

[2] [§]These authors contributed equally to the paper.

Figure 4.1: The head-worn egocentric capture system in this chapter is capable of reconstructing the wearer and their surrounding environment in 3D. Left: Hardware prototype. Center: An individual using the device. Right: Dynamic reconstruction of the user's body pose and static environment, obtained solely from the prototype's headset-mounted cameras.

4.1 Introduction

We envision a future in which passive 3D capture of user experiences is a feature of commonplace head-worn devices. In this future, AR systems have shrunk to the form factor of conventional eyeglasses and so can be worn all day just like ordinary eyeglasses. In order to enable a self-contained 3D capture system, we wish to augment such eyeglasses with a multiplicity of inward- and outward-looking miniature cameras. These cameras form an egocentric reconstruction system that (1) captures its wearer's 3D pose, face, body, and limbs, and (2) maps the 3D structure of its surroundings. The resulting dynamic scene can (3) be displayed to other users, using AR/VR systems to create a shared, immersive 3D experience. Such self-contained, head-worn systems can enable shared presence and virtual touring to occur in any indoor or outdoor location, with no reliance on any instrumentation other than that in the user's headgear.

In this chapter, a prototype system is introduced for demonstrating the egocentric capture and reconstruction as shown in Figure 4.1. Example camera views on the device are shown in Figure 4.2. The main challenge of reconstruction from such head-worn cameras is the sparse visibility of body parts, which leads to large gaps in self-reconstruction. To address this problem, a deformable-model-based approach is introduced to complete the unobserved parts of the wearer: as the generic parametric model still has gaps in the user's appearance, e.g., in clothing, texture, and other detailed characteristics such as hair, such surface details are transferred from a pre-scan of the

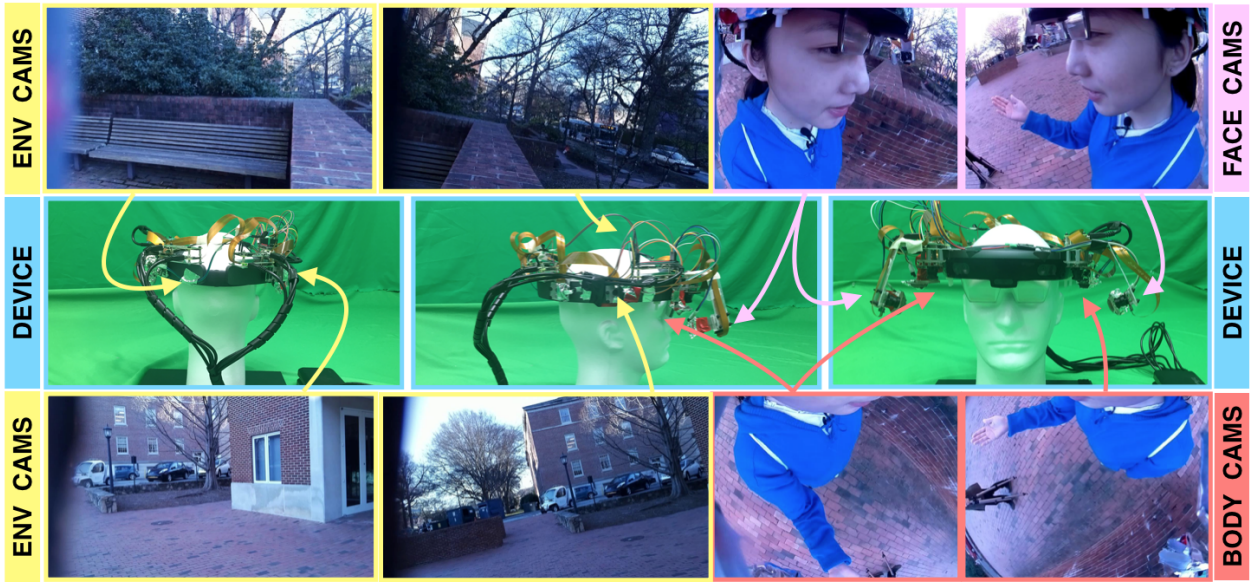


Figure 4.2: The eight views from a single time-point of capture on the prototype device. Outward-looking environment cameras (yellow) are placed on the side and rear of the device. Face-oriented cameras (pink) are placed on short arms on either side of the device. Downward-facing body cameras (orange) are located on both sides of the user's forehead. The top row shows the left rear external, left side external, right face, and left face views. The bottom row shows the right rear external, right side external, right body, and left body views.

full body of the user. When such systems are miniaturized, personalized, and worn for long periods of time, we expect that they can automatically and gradually acquire detailed full-body information of their users and their wardrobes.

The reconstruction approach is a user-oriented model-based self-reconstruction pipeline that combines parametric body and face models. The model-based incomplete reconstruction is re-targeted to a high-quality pre-scan of the user in a coarse-to-fine manner. The deformable models have two types of parameters: shape-related and pose-related. The shape parameters of the body and face are estimated in preprocessing stage by fitting the models to the pre-scan. The pose-related parameters, body pose and facial expression, are detected at run-time using CNN-based pose estimation and through audio- and video-based facial expression estimation, respectively.

The system demonstrates full scene reconstruction, including the user's moving body with audio and their surrounding environment. The environment is reconstructed using structure-from-motion with outward-looking cameras. The trajectory of the user's head is determined using multiple

calibrated cameras, which allows the system to localize the reconstructed user within the environment over time. The unified capture can be immersively experienced in a VR system.

The remainder of this chapter is organized as follows: The related work is discussed in Section 4.2. The overall self-reconstruction pipeline is discussed in Section 4.3. The egocentric capture prototype is described in Section 4.4. Section 4.5 describes the pre-scanning process. Sections 4.6 and 4.7 address CNN-based body pose estimation and CNN-based facial expression estimation from both audio and video. The environment and head pose estimation techniques are detailed in Section 4.8. After integration considerations in Section 4.9, experimental results are shown in Section 4.10, followed by limitations and future work in Section 4.11.

4.2 Related Work

As our society grows ever more connected digitally, individuals are increasingly interested in maintaining a connection with reality when communicating their experiences and ideas with others across the globe. Indeed, modern video (e.g., YouTube) and televideo (e.g., FaceTime or Cisco TelePresence) content-sharing systems are used daily by hundreds of millions of people because they come the closest to relaying a veridical human experience. However, while such systems have grown in popularity as substitutes for witnessing events firsthand or having face-to-face meetings, these technologies fall short of delivering an actual sense of shared physical presence. The 360° videos (e.g., Google Jump (2015)) offer more immersive video experiences but limit the viewer to a fixed position of observation. Prototype 3D capture and telepresence systems such as Microsoft Research's Holoportation (Orts-Escolano et al., 2016) have likewise demonstrated promising steps towards shared 3D presence, but require substantial, expensive, instrumented areas. However, such instrumentation hampers the ability of an everyday user to capture 3D directly.

4.2.1 Static 3D Reconstruction

Static 3D Reconstruction of an environment from photos and videos has been a long-standing research thrust in computer vision. 3D reconstruction algorithms include structure from motion

(Snavely et al., 2006; Agarwal et al., 2011; Heinly et al., 2015; Schonberger and Frahm, 2016) combined with stereo vision (Scharstein and Szeliski, 2002), simultaneous localization and mapping (SLAM) (Engel et al., 2014; Mur-Artal et al., 2015; Wang et al., 2017; Tateno et al., 2017; Engel et al., 2017), multi-view vision (Seitz et al., 2006; Schönberger et al., 2016), and depth-camera-based algorithms (Newcombe et al., 2011; Izadi et al., 2011), and can be used to reconstruct only static scenes. The system in section 4.3 is built on the progress made by the body of work in these areas to obtain its environment reconstruction and to track the user within the environment. Moreover, the approaches are extended to leverage the constraints provided by the multi-camera setup in the system.

4.2.2 Dynamic Object Reconstruction

Dynamic Object Reconstruction has long been an active research area. Most approaches rely on moderate surface deformations or known object shape for reconstructing a 3D model using a video of the object (Tong et al., 2012; Hirshberg et al., 2012; Weiss et al., 2011; Li et al., 2013; Zeng et al., 2013; Liao et al., 2009). Alternatively, motion capture systems (De Aguiar et al., 2008; Gall et al., 2009; Ballan and Cortelazzo, 2008; Vlasic et al., 2008; Wu et al., 2013; Ye et al., 2012; Starck and Hilton, 2007; De Aguiar et al., 2007) deliver reliable reconstructions of human bodies from a sequence of color and/or depth videos. These approaches require a pre-scanned body model or template, an instrumented environment, and complicated skinning and rigging preprocessing. These factors prevent their application to reconstructing general shapes in unconstrained environments, which is mandatory for a mobile 3D capture system.

There has also been a keen interest in parametric body models for reconstruction and tracking. Allen et al. (2003) leveraged high-resolution range scans to develop a parametric body shape model. The SCAPE model (Anguelov et al., 2005) advanced this approach to not only parameterize body shape but also encode pose deformation. Chen et al. (2013) further extended the SCAPE model by introducing parameters to explain the deformation from clothing. Their model deformed the overall person model non-rigidly by applying the composite transformations of the poses, the shapes, and

the clothing for each triangle independently. Loper et al. (2015) proposed the SMPL model, which provides more realistic deformations and achieves a more accurate representation of the effects of joint motion. The parametric body models can be utilized to estimate human shapes in conjunction with visual pose estimation (Bogo et al., 2016; Xu et al., 2018). However, these approaches require the visibility of all joints on external camera views to fit full-body shapes and pose based on the observations.

Another work for template-free dynamic surface fusion (Dou et al., 2015; Newcombe et al., 2015; Dou et al., 2016; Orts-Escolano et al., 2016) has shown promising results for object-level and human reconstruction in outside-in capture scenarios for instrumented environments. However, these methods are not suited to work with passively captured data from a mobile system, which requires reconstruction methods that operate in arbitrary environments without external instrumentation.

4.2.3 Dynamic Scene Reconstruction from Depth Sensors

There has been significant interest in dynamic scene reconstruction from depth sensors. For example, Maimone and Fuchs (2011, 2012) constructed a real-time 3D capture system using a dozen Kinects. This method adapts the volumetric fusion of Curless and Levoy (1996) to dynamic objects (i.e., people) while incorporating depth and color information. More recent room-size dynamic object reconstruction (Dou and Fuchs, 2014) combines pre-scanning of the static scene parts, data accumulation for dynamic objects, and rigid and nonrigid tracking. However, these approaches rely on successful depth image capture using structured light, which typically fails outdoors. The system in this chapter targets both outdoor and indoor use and hence cannot use structured light sensors.

4.2.4 Egocentric Motion Capture

Egocentric, body-worn cameras have been used for 3D pose estimation of certain parts of the body such as facial expressions via helmet-mounted cameras (Olszewski et al., 2016) or finger motions via wrist-worn sensors (Kim et al., 2012). Shiratori et al. (2011) determined full-body

motions based on 16 body-worn cameras with poses estimated through structure-from-motion, assuming a static environment. Chan et al. (2015) and Jiang and Grauman (2017) proposed learning-based approaches to predict full-body poses from a chest-worn camera view to infer invisible poses with limited accuracy. Zhang et al. (2014) used a single outside-in depth camera combined with foot-worn sensors for full-body pose estimation. All of the above approaches only perform skeleton-based motion capture and do not reconstruct the 3D surface of the wearer solely from the body-worn cameras.

Rhodin et al. (2016) employed two head-mounted fisheye cameras to estimate the full-body skeleton pose. The large field of view allowed the cameras to observe most of the body and to integrate with approaches based on outside-in cameras. However, their system required the head-mounted cameras to be placed on long telescopic arms reaching significantly outward in front of the wearer. This obtrusive setup enabled them to perform a stereo-based body reconstruction at the cost of usability. In contrast, the system presented in this chapter leverages cameras close to the body, trading a full-body stereo view for broad usability.

Significant improvements have been made using learning-based approaches to deal with the unusual viewpoints. Recent methods based on a single head-worn camera view (Xu et al., 2019; Tome et al., 2019) have used less-obtrusively mounted cameras to arrive at pose estimation improvements. However, the form factors employed are still too obtrusive for wide acceptability. Our 3d capture system using an eyeglasses form factor, with its challenges and approaches, will be discussed in Chapter 5.

4.3 System Overview

An overview of the mobile capture pipeline is shown in Figure 4.3. From a computational perspective, the inputs to the system are individual views from synchronized head-worn cameras, and the output is a posed 3D model of the wearer placed into a reconstructed 3D model of the surrounding environment. Body poses and facial expressions are captured entirely from the on-device camera views, as is the 3D environment model. For visualization, a pre-computed

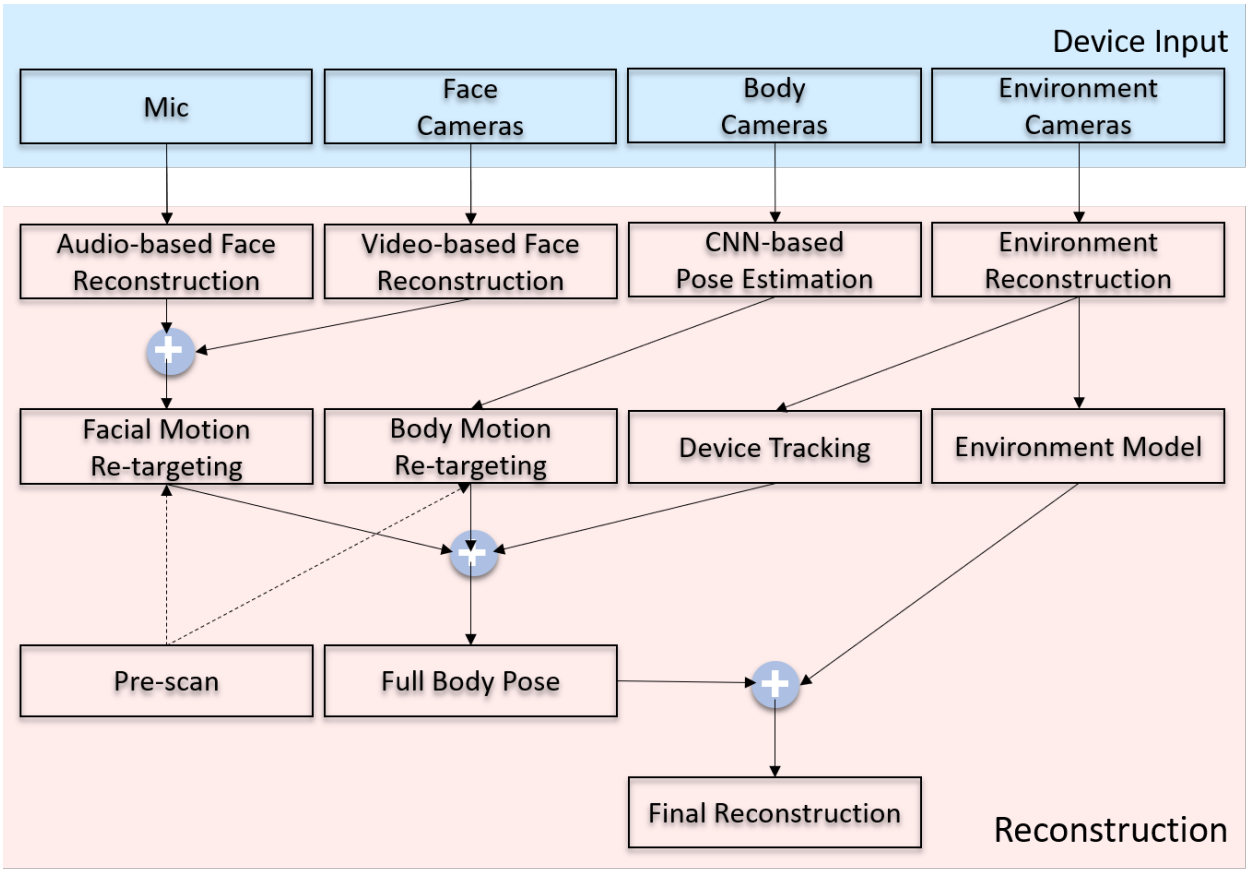


Figure 4.3: Functional overview of the system, with HoloLens-mounted (Microsoft HoloLens 1, 2016) capture components at top and offline reconstruction processing pipeline at bottom.

digital human representation ("pre-scan") of the user is posed according to estimated face and body parameters.

Details about the head-worn camera configuration are provided in Section 4.4, and the pre-scan acquisition process is described in Section 4.5. The reconstruction approach consists of three processing pipelines, each of which takes in separate camera imagery: body pose estimation, consisting of skeleton joint detection and 3D triangulation (Section 4.6); face reconstruction from video and audio data (Section 4.7); and environment reconstruction, which encompasses both reconstructing the 3D scene and tracking the motion of the user as they move within their surroundings (Section 4.8).

The individual reconstruction results are combined (see Section 4.9 for results). The body pose and face expressions are applied as parametric deformations of their associated pre-scan models; these adjusted face and body pre-scans are then combined to create the momentary digital human representation of the user. This representation is then placed into the scene based on the tracked location of the user within their environment, and the placed (animated) model can be rendered in the context of the reconstructed static scene around the user. The resulting dynamic 3D model can then be utilized for a variety of applications, such as virtual tours (see Subsection 4.10.4).

4.4 Mobile Headset Prototype

In our vision for ubiquitous AR/VR systems of the future, an individual will be able to fully capture themselves and their 3D surroundings solely from a lightweight pair of eyeglasses fitted with miniature cameras. We anticipate that these devices, possibly combined with a small backpack computer for processing, will have functionalities for both general capture (e.g., self-created VR content analogous to current online video services) and telepresence (i.e., real-time 3D ego-capture, coupled with AR displays). In this work, a prototype headset is developed to demonstrate the various camera configurations and reconstruction approaches that such a device would employ.

The prototype 3D capture unit has been outfitted with 8 Pi V2 miniature cameras (Figure 4.2 and Figure 4.4). These cameras are divided into three categories based on their function: four

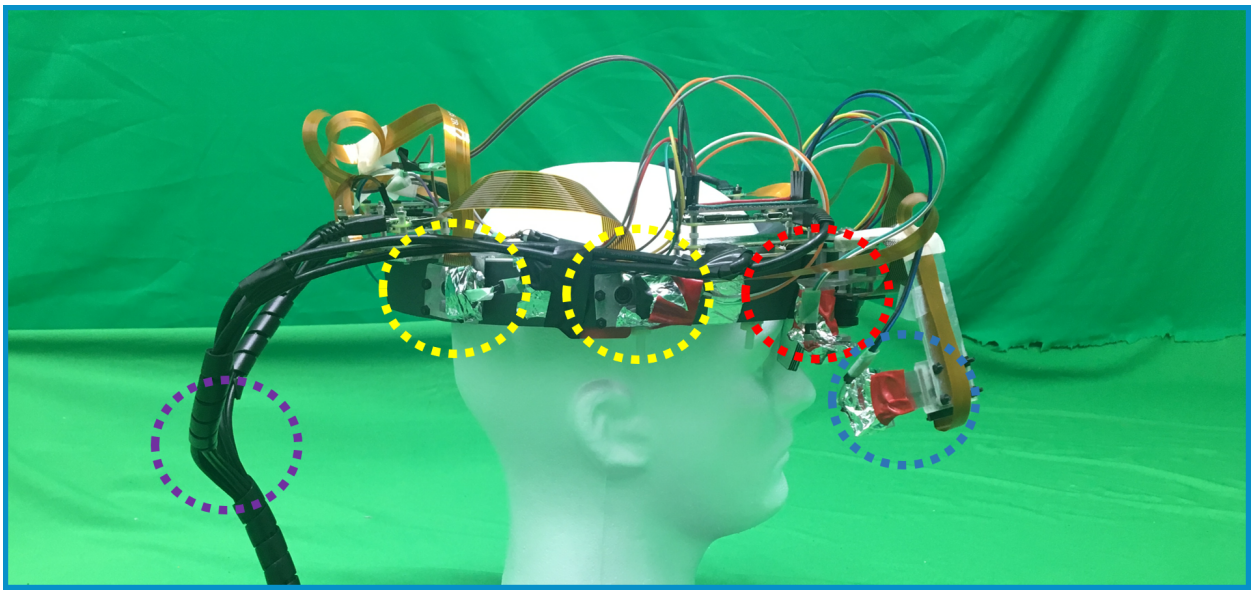


Figure 4.4: The prototype device is equipped with 8 miniature cameras, each paired with an LED for synchronization. The camera-LED pairs are directly mounted on a Microsoft Hololens: 2 downward-facing body cameras (red), 2 face-oriented cameras (blue), 4 outward-looking environment cameras (yellow). The cameras on the headset run on miniature computers powered by portable battery banks (purple).

outward-facing cameras capture the environment and track the device's motion, two downward-facing cameras capture the user's body, and two face-oriented cameras capture the wearer's facial expression. The cameras on the headset run individually on Raspberry Pi Zero miniature computers powered by portable battery banks worn in a backpack. The external views are captured using 70° diagonal FoV cameras and are located on the sides and back of the headset. The face and body cameras have 160° diagonal FoV lenses; the body cameras are placed slightly in front of the wearer's forehead, and the face cameras are placed on slightly extended mounts $\sim$9cm from the user's face. We expect that future systems will be able to reduce the outside-in distance of the face cameras even further, to the point where the cameras are mounted directly next to the lenses of the eyeglass frame.

The cameras are synchronized offline using LED blinking (Bapat et al., 2016) and capture at $25fps$. (These were design decisions for the prototype; hardware synchronization and faster frame rates are possible in principle.) Anticipating future AR integration capabilities, the camera system is mounted on a Microsoft HoloLens headset (Microsoft HoloLens 1, 2016); however, the HoloLens'

onboard display or capture technologies is currently not used. Also note that the capture scenario involves online capture and offline 3D reconstruction – in this work, the motivation is to demonstrate the technologies involved in performing automated, hands-free, use-anywhere 3D capture.

System Calibration. In addition to frame-level camera synchronization, it is assumed that the intrinsic and relative extrinsic camera parameters for the device are known before capture. This calibration involves estimating the relative rotations between the cameras, the absolute distances between the cameras' centers of projection, and the position of the rig in relation to the wearer's head. Camera intrinsics were computed using standard checkerboard-based camera calibration. To capture the relative camera poses, a small, well-textured scene was set up and the headset was moved/rotated by hand (without anyone wearing it) while capturing imagery from the cameras. Then this synchronized multi-camera sequence was reconstructed using structure-from-motion (SfM) (Schonberger and Frahm, 2016) with a bundle adjustment that estimates a global pose for the device at each time instant while enforcing static relative poses for the cameras in the cluster. Since SfM reconstructions are inherently scale-independent, the absolute scale of the headset was recovered by manually comparing the sizes of reconstructed objects with known real-world measurements. The location of the rig with respect to the wearer was then established by computing the midpoint of the two side external cameras and aligning it with the approximate midpoint of the wearer's temples.

4.5 Digital Human Pre-scan

The egocentric system integrates motion capture and environment reconstruction. For visualization, however, it is impossible to create a complete model of the wearer from the headset views because the headset captures only partial views of the user's face and parts of their body, resulting in an incomplete digital human representation. Instead, an off-device 3D scan ("pre-scan") of the user that fully captures their body shape and clothing is obtained. The system localizes body skeleton joints in the two downward-facing views, and parameters for the user's facial expression are computed from the two (non-overlapping) face-oriented views. The pre-scan is deformed to match the skeleton

and face parameters and then placed in the 3D environment based on the estimated device pose. Details about the skeletal rigging and skinning of the pre-scan are provided in Subsection 4.6.4.

To obtain the pre-scan, a textured mesh of the entire body, a 3D scanning software (ItSeez3D, 2014) is used. The user stands still with their arms extended while another individual moves a small RGB+D camera unit around them to capture the body surface and texture.

In the future, we anticipate that pre-scan acquisitions could be completed entirely on-device, with the wearer capturing their appearance by, e.g., placing the device on a table and walking in front of it, or by wearing the device and standing or turning in front of a mirror. Such on-device processing would not only increase the ease of use, but would also enable on-the-fly representations of new individuals or allow updates of the clothing or appearance of the same individual.

4.6 Video-based Body Pose Reconstruction

Body pose estimation solely from head-worn cameras is a challenging task. The most closely-related system, EgoCap (Rhodin et al., 2016), uses two head-worn fisheye cameras on an extended 'V'-shaped rig. However, they extend 20-30 cm away from the user's head, which is prohibitive for convenient, portable use. The egocentric system in this chapter is unique in that it is targeted to locate commodity cameras directly on the compact headgear; this generally results in very restricted viewpoints that provide less reliable measurements for body pose estimation, particularly for the legs, which are far from the cameras and often occluded. To overcome the difficulties in capturing body pose, the system leverages deep convolutional neural networks (CNNs) to perform body part detection in the individual downward-facing views, as well as an additional recurrent neural network (RNN) module to obtain a final skeleton-based human pose estimation.

4.6.1 2D Human Body Joint Detection

To solve the initial problem of detecting the device wearer in the downward-facing views, an extended convolutional pose machine (CPM) network (Wei et al., 2016; Cao et al., 2019) is employed to detect 2D joint positions in each image independently. CPM incorporates a convolutional neural

network into the pose machine framework (Ramakrishna et al., 2014), which enhances image feature extraction (in this case, 2D joint locations) by leveraging inference on image-dependent spatial models. CPM is built upon an end-to-end, multi-stage deep network that enables the learning of both joint appearances and spatial relationships in input imagery. Beyond traditional cascaded networks, CPM is also an interactive sequence framework, with each stage considering the context of previous stages in order to derive an overall set of joint positions for a given image.

A pose machine consists of a hierarchy of 2D joint predictors $g_t(\mathbf{f}_t(x), \psi_t(j, \mathbf{b}_{t-1}))$ that output joint-specific belief values for all positions x in the image domain, for each stage t in the hierarchy. $\mathbf{f}_t(x)$ represents a stage-specific feature embedding for the input image, and $\psi_t(\cdot)$ maps the existing volume of belief values $\mathbf{b}_{t-1}$ for all joints across the image into a specific context mapping for joint j. Given the input image, the first stage $g_0(\cdot)$ is an image-space classifier that produces a joint-probability volume $\mathbf{b}_0 = \left\{b_t^j(X_j = x)\right\}_{j \in 0 \ldots J}$, where X_j is a random variable relating the position of joint j. Later stages $g_t(\cdot)$ update the belief for assigning a location to each part:

$$g_t(\mathbf{f_t}(x), \psi_t(j, \mathbf{b}_{t-1})) \mapsto \mathbf{b_t}. \tag{4.1}$$

The final 2D joint predictions are retrieved as the most probable locations for each X_j after the final belief values are predicted.

The prediction and image feature computation modules of a pose machine can be replaced by a deep convolutional architecture, allowing for both image and contextual feature representations to be learned directly from data. The CPM contains multiple stages of a fully convolutional network cascaded to characterize both the local features of the input image and the global features across larger receptive fields. By chaining prediction stages, the receptive fields at the output layer of the network are large enough to allow the learning of potentially complex and long-range correlations between body parts.

The cost function minimized at each stage of the CPM is an l_2 distance between the predicted and ideal belief map for each joint:

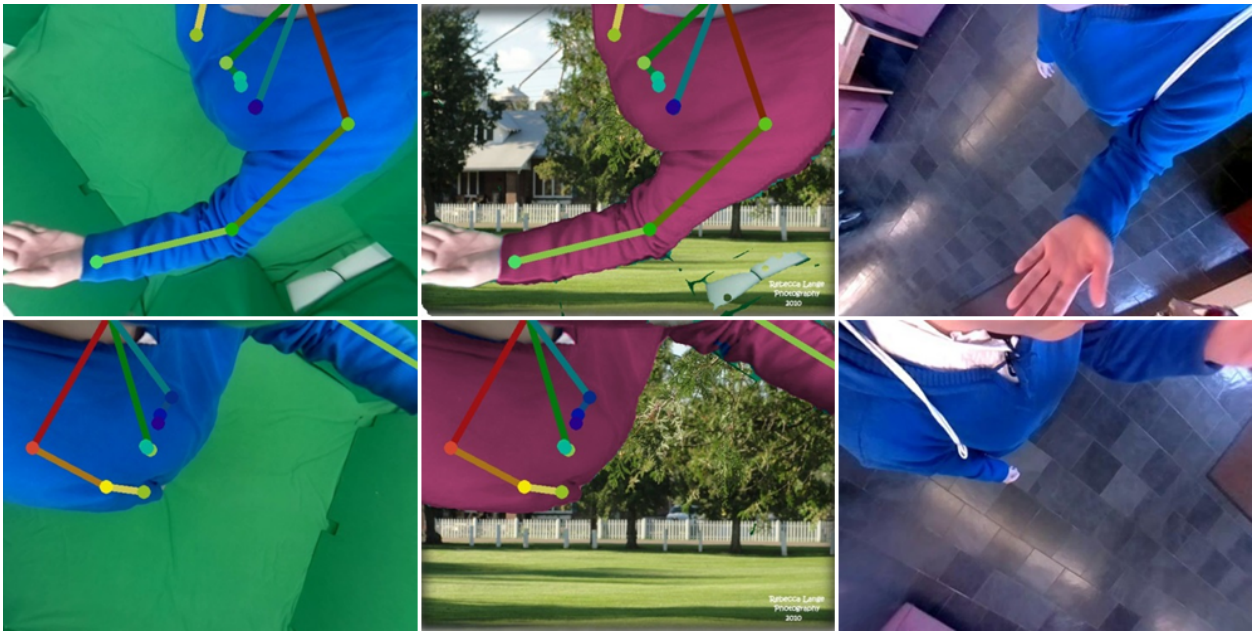


Figure 4.5: Example images from the pair of downward-facing body cameras on the headset device. Left: Training images captured in the green-screen room. Middle: Training images augmented by shirt recoloring and background replacement. Right: Images from the hallway demo. The top and bottom rows show images from the left and right body cameras, respectively. The colored skeleton depicts the projection of the ground-truth 3D joint positions into the individual views in the original captured and augmented training images.

$$\ell_t = \sum_{j=1}^{J} \sum_{x} \|b_t^j(x) - b_*^j(x)\|_2^2, \tag{4.2}$$

where $b_*^j(X_j = x)$ represents the ideal belief map for joint j. The overall objective for the full architecture is obtained by adding losses over all T stages and is given by

$$\mathcal{F} = \sum_{t=0}^{T-1} \ell_t. \tag{4.3}$$

As seen in the views of the downward body cameras shown in Figure 4.5, the detectable joints are defined as the shoulders, elbows, wrists, hips, and knees. Ankles are generally not visible from the near-body views – for instance, each foot is independently visible for only $\sim$33% of a gait cycle. – so instead they are modeled in 3D using motion priors (see Section 4.9).

The joint positions are predicted via a custom-trained CPM for the egocentric input views (see Subsection 4.6.3). This 2D detection is trained separately from the subsequent 3D pose estimation

network. The original images are padded to allow predicting the position of joints that are located outside the images. This padding enables the fully convolutional network to learn correlations between (and predict 2D locations for) all joints, whether or not they are actually visible in the input views.

4.6.2 3D Human Pose Sequence Estimation

Given the 2D detection result, a 3D pose sequence module is employed to predict the 3D skeleton joint positions over time. This module leverages an RNN to capture long-term motion trajectories for all observable joints. Compared to general neural networks, RNNs are able to scale to much longer temporal sequences and are practical for sequence-based specialization, such as video processing. This is because in RNNs, each member of the output is a function of the previous member of output, with all outputs being produced by the same update rule. Thus, the temporal motion information between frames can be effectively incorporated into the 3D pose prediction.

For the recurrent 3D human pose network, a sequence of 2D positions (x_t^j, y_t^j) in the images of each of the two body-camera views and their corresponding probabilities p_t^j are taken as the network input $X = [X_1, X_2, ..., X_t]^T$, where $X_t = [(x_t^1, y_t^1, p_t^1), \ldots]$ at time step t. (Note that t here refers to the temporal domain of the capture sequence and j refers to the joints over both views.) For training, points and probabilities are generated by random Gaussian perturbations of the ground truth 2D joint position. At run-time, they are generated using the trained CPM.

The network consists of three fully connected layers (512, 1024, and 1024 neurons, respectively), one recurrent layer (2048 hidden states), and finally two fully connected output layers (1024 and 30 neurons) that unilaterally predict all 3D joint positions for a given time step t.

$$h_t = \sigma(W_{h_1} h_{t-1} + W_{h_2} f_i(X_t) + b_h) \tag{4.4}$$

$$Y_t = f_o(h_t) \tag{4.5}$$

where f_i is the function applied on the input before the recurrent part; h_t is the recurrent layer's hidden state at step t; W_{h_1}, W_{h_2}, and b_h are the weights and bias; σ is a non-linear function; and f_o is the function applied after the recurrent layer to obtain the output 3D positions Y_t at time step t.

The sum of l_2 distances between the ground truth 3D positions and the predictions are minimized as:

$$E = \sum_t \|Y_t - Y_t^*\|_2^2, \tag{4.6}$$

where Y_t^* consists of the ground truth 3D body joint positions at frame t. Incorporating the previous 3D pose prediction at each stage allows the network to compute the pose predictions robustly.

4.6.3 CNN Training and Testing

Training Dataset Capture. The key challenge for training the body pose estimation network lies in obtaining ground truth data for the 2D and 3D joint positions. To solve this problem, a data capture setup was constructed for outside-in markerless motion capture, including calibrated headset tracking and background subtraction for data augmentation.

The videos for the training dataset and the ground truth positions of 3D human body joints are obtained using a calibrated set of synchronized external cameras. The training setup consists of a mid-size room with the outside-looking-in cameras placed near the walls. The user wearing the headset device is standing in the middle of the capture space. Figure 4.6 shows an example set of camera images captured at the same time.

In each external view, a pre-trained OpenPose CPM network (Wei et al., 2016; Cao et al., 2019) is applied to detect 2D joint positions. Using pre-computed camera poses in the room, each joint can be triangulated in 3D over time. The 6-DoF poses of the downward-facing cameras are also tracked using a checkerboard pattern mounted on the device. The relationship between the checkerboard and the device cameras is calculated using hand-eye calibration (Shah et al., 2012), and the pose of the device within the capture space is determined by recovering the pose of the checkerboard from

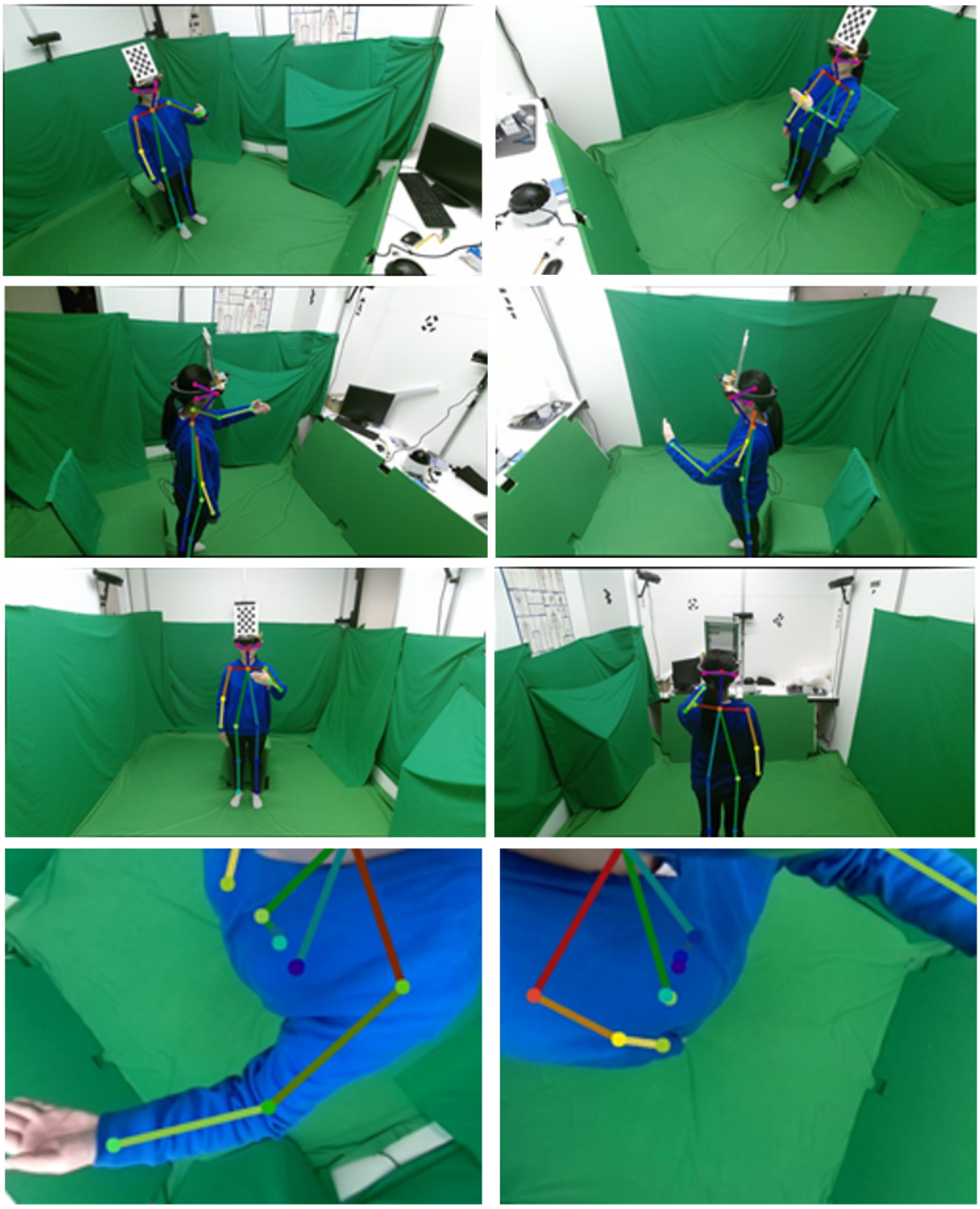


Figure 4.6: Images from the six external cameras (first three rows) and two top-down body cameras (the last row) on the headset device used for capturing the ground-truth body pose dataset. The colored skeleton depicts the ground-truth 3D joint positions.

the external views. Given the triangulated 3D joint positions and the pose of the device, ground truth 3D joint positions are obtained by simply applying the scene-to-device transformation, and 2D joint positions for each camera are then determined via projection using the camera intrinsics.

Network Training. Using data from the capture environment, a new CPM network is trained for the downward-facing views and an RNN to predict the 3D human pose sequence. The Caffe deep-learning framework (Jia et al., 2014) is used to train both networks. To enhance the generality of the CPM, the surrounding room was made into a "green-screen" environment, and the capture subject was given a blue sweater to wear during training. The training data was then augmented by replacing the green surfaces with random floor/object textures and the blue shirt with randomly adjusted hues. The input images were further augmented using flips, rotations, and translations. In order to obtain sufficient samples for training the 3D pose RNN, fast-motion speeds are simulated by interpolating poses between the frames, and the captured frame sequence is also subsampled into many shorter frame sub-sequences.

Network Execution. At run-time, the CPM is used to hypothesize the most likely 2D joint positions for the input downward-facing imagery. The 3D RNN then takes these points, along with their probabilities, and outputs a hypothesis for the 3D position of each joint relative to the left downward-facing camera. The 3D joint result is post-processed using a Kalman filter and basic exponential smoothing, which allows to robustly account for sporadic mis-predictions of the 3D joint position. The end result is a smoothed skeletal motion capture sequence of the user across time.

4.6.4 Body Motion Re-targeting

Rigged parametric body models (Allen et al., 2003; Anguelov et al., 2005; Chen et al., 2013; Loper et al., 2015) can be exploited to deform the pre-scan model constrained by the 3D joint positions output by the RNN in the previous subsection. The re-targeting approach employs the Simplified-SCAPE parametric body model (Pishchulin et al., 2017) using linear blend skinning for computational efficiency. During pre-processing, the parametric body model is fit to the pre-scan

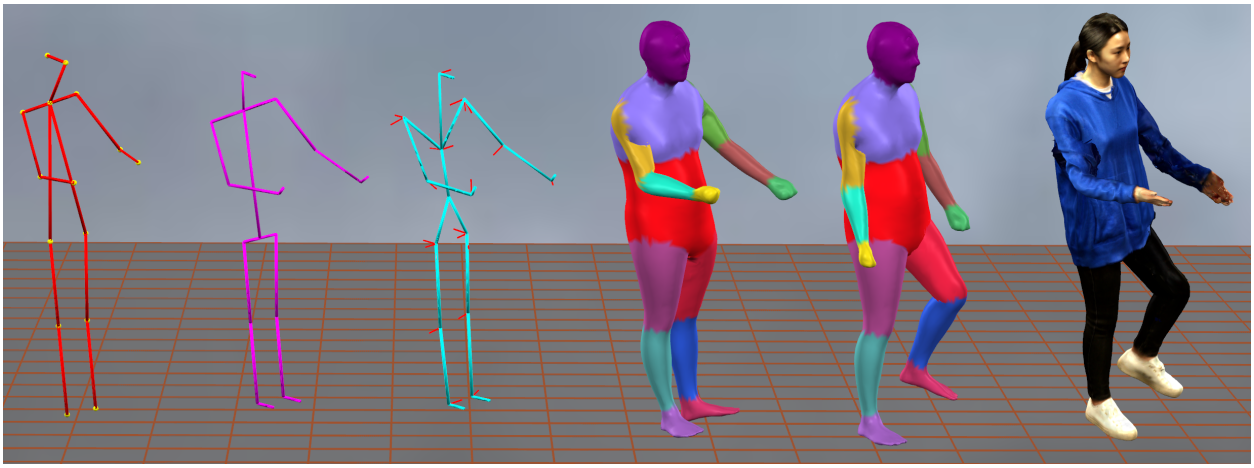


Figure 4.7: Body pose re-targeting. From left to right: 1) Detected joint positions. 2) Bone length adjusted joint positions with hand/foot orientation constraints. 3) Rotational skeleton of the model. 4) Deformed body model in which joint angles are estimated by fitting the model skeleton (3) to the canonical positions (2) using joint-limit-constrained inverse kinematics. 5) Walking motion prior. 6) Final textured pre-scan model with the blended pose.

model for automatic rigging. The predicted 3D posture at each frame is applied to the rigged pre-scan at run-time.

The body model $\mathbf{M}(\theta, \beta)$, which is represented in homogeneous coordinates, is specified by the joint configuration θ and shape parameters β of PCA space $\mathbf{S} \in \mathbb{R}^{4|V|\times|\beta|}$, and is deformed from the mean body shape $\widehat{\mathbf{M}}$:

$$\mathbf{M}(\theta, \beta) = \mathbf{R}(\theta)\widehat{\mathbf{M}} + \mathbf{R}(\theta)\mathbf{S}(\beta). \tag{4.7}$$

$\mathbf{R} \in \mathbb{R}^{4|V|\times 4|V|}$ is the block diagonal matrix of per-vertex joint transformations. $\mathbf{M}(\theta, \beta)$ is fit to the pre-scan $\mathbf{T}$ to estimate the vertex correspondences by minimizing the following energy w.r.t θ and β:

$$\mathbf{E}_{\mathbf{M}}(\theta, \beta) = \sum_{i=1}^{|V|} ||v_i(\mathbf{M}(\theta, \beta)) - \mathrm{NN}_i(\mathbf{T})||_F^2, \tag{4.8}$$

where $||\cdot||_F$ is the Frobenius matrix norm. Each vertex $v_i(\mathbf{M}(\theta, \beta))$ of the model is fit to its closest compatible nearest neighbor vertex $\mathrm{NN}_i(\mathbf{T})$. More details regarding the optimization are given in Pishchulin et al. (2017). Using Equation 4.8, the preprocessing shape parameters β_0 with

bone lengths and pose parameters θ_0 are determined for the association between the model and the pre-scan. β_0 and bone lengths are fixed for the entire run-time sequence.

The skeletal joint placements Θ_T of the pre-scan $\mathbf{T}$ are transferred from the fit joints $\Theta_M(\theta_0, \beta_0)$ of $\mathbf{M}$. Based on the vertex correspondences from Equation 4.8, the skin weights $w(v_i) = \{w_1(v_i), ..., w_{|\theta|}(v_i)\}$ of each model vertex are also transferred to $\text{NN}_i(\mathbf{T})$. The skin weights of remaining pre-scan vertices are interpolated from nearby $\text{NN}_i(\mathbf{T})$. From the transferred joint structure and skinning weights, the captured skeletal animation can be accordingly applied to the pre-scan.

At run-time, the pose parameters θ_t at time t of the body model $\mathbf{M}(\theta_0, \beta_0)$ are estimated from the 3D joint positions output by the RNN. Specifically, the joint positions form a positional skeleton using a pre-defined joint structure and pre-defined joint correspondences between the model skeleton and the positional skeleton. The joint angles θ_t are estimated from this positional skeleton using joint-limit-constrained IK (Drexler and Harmati, 2012). To fit the model skeleton to the positional skeleton, bone lengths of the positional skeleton are adjusted to match the model skeleton. The rigid-body transformation from the model to the positional skeleton is estimated by minimizing point-to-point distances of spine and hip joint pairs. The remaining joint angles are estimated using the constrained IK.

The angular derivative $\dot{\theta}$ of joints are estimated by solving the differential IK:

$$\dot{\theta} = \mathbf{J}^{\#}\dot{x} \tag{4.9}$$

where $\dot{x}$ is the change in corresponding joint positions, and $\mathbf{J}^{\#}$ is the pseudo-inverse of Jacobian matrix. The joint angle limit is constrained by transforming the angle derivative $\dot{\theta}$ to the transformed space $\dot{z}$. When $z_t = z_{t-1} + \dot{z}_t$ converges to the joint limit, it regains manipulability by enforcing z_i to move in the other direction (Drexler and Harmati, 2012). This guarantees that $\theta_i = T(z_i)$ is always a valid joint angle. The elbow and knee joint limits are used to prevent anatomically implausible poses.

The foot and hand orientations are not included in the positional skeleton, however, which can result in an IK result that arbitrarily twists the arms and legs. To prevent this, dummy joints are added at each end effector (hands, feet, and head) to constrain them to valid orientations in IK. The torso normal direction is set according to these dummy joints. Figure 4.7 shows the joint fitting result with the joint limits and the orientation constraints.

From the estimated pre-pose θ_0, and current pose θ_t, the pre-scan $\mathbf{T}$ is deformed as:

$$\widehat{\mathbf{T}} = \theta_t \, \theta_0^{-1} \, \mathbf{T}, \tag{4.10}$$

where θ_0^{-1} is the inverse joint transformation of θ_0, which moves the pre-scan to the neutral pose of $\mathbf{M}$, allowing the current pose θ_t to be applied directly.

4.7 Audio/Video-based Face Reconstruction

To obtain a high-quality 3D model of the user's face, a similar pipeline is adopted to the body-modeling approach as shown in Figure 4.8. In the prototype system, two on-device cameras are used to capture each side of the user's face. This is in contrast to most work on face reconstruction that utilizes a single frontal view for face capture. The goal of the setup is to have the cameras capture adequate views of the face without being obtrusive. Similar to prior live face capture systems, facial landmarks are detected in the individual views to fit a 3D deformable face model that incorporates both face shape and expression. The reconstruction quality can be further enhanced by transferring facial expressions (Sumner and Popović, 2004) from the deformable face model to a high-quality user model. To compensate for the limited visibility of the face, an audio-driven deep neural network is employed to enhance the facial expression estimation.

4.7.1 Video-based Face Reconstruction

The video-based face reconstruction pipeline takes as input two synchronized images from the downward-facing cameras, as well as a pre-scan model of the user's face. For each captured

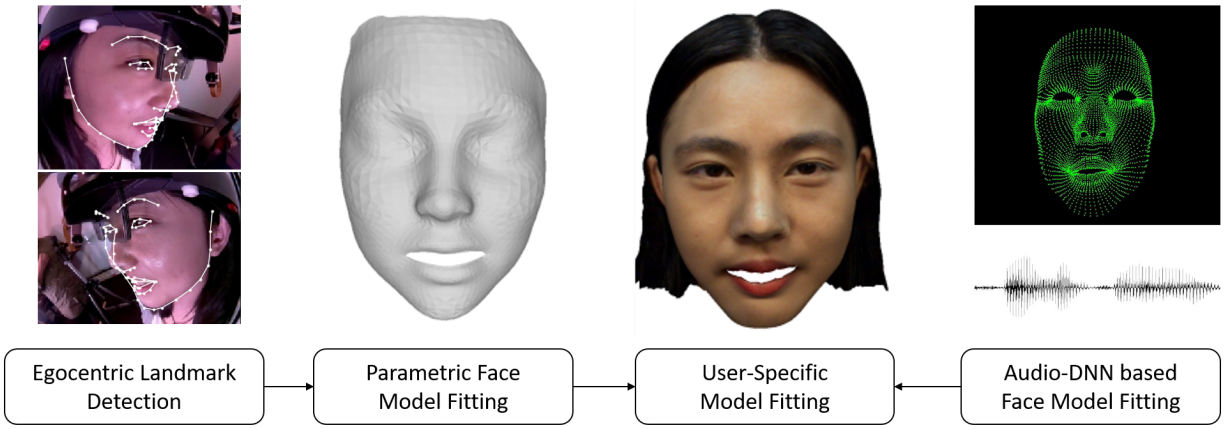


Figure 4.8: Audio/Video-based Face Reconstruction Pipeline.

time instant, 2D landmarks are detected in the two images. Then a deformation of the pre-scan is computed by minimizing the reprojection error between the face model's fiducial 3D landmarks and their corresponding 2D detections.

Pre-scan Fitting. As input to the capture process, a morphable model is fit to the high-quality face pre-scan. In general, the face model has three sets of parameters: the pose T (global rotation and translation in relation to the left camera), shape parameters α_s, and expression parameters α_e. First, 68 3D landmarks are manually labeled in both the pre-scan and the model, and then the face pose T is computed through rescaling and fitting these correspondences. Following Cao et al. (2014b), it is assumed that the pre-scan has a neutral expression α_{e_0} and the shape coefficients α_s are estimated by minimizing

$$E_{fPre} = \omega_{lm} E_{lm} + \omega_d E_d + \omega_{reg} E_{reg}, \tag{4.11}$$

where the first term E_{lm} penalizes errors in the 3D landmark alignments, the second term E_d relates to dense vertex matching between model vertices and their nearest neighbor vertices in the pre-scan, and the final term E_{reg} regularizes the PCA coefficients α_s. The full method and objectives used for shape parameter optimization are described in Cao et al. (2014b). In the formulation, $\omega_{lm} = 1$, $\omega_d = 2$, and $\omega_{reg} = 1$ are used respectively.

Detecting 2D Landmarks. To compute the face model parameters for the user at a given time instant, 2D facial landmark detection is performed first from the side images. The problem of 2D facial landmark localization for frontal face images has largely been solved (Cao et al., 2014c,a; Xiong and De la Torre, 2013; Zhu et al., 2016; Bulat and Tzimiropoulos, 2017). However, these methods fail for the profile and oblique views that occur in the egocentric side image views. Bulat and Tzimiropoulos (2017) have shown good performance on significantly non-frontal 2D and 3D face alignment in difficult illumination conditions; however, it is found that this method could not detect landmarks in most of the egocentric images. This neural network is fine-tuned with new data captured from the egocentric viewpoints and provided a rough bounding box to the face detector, which greatly improved the detection accuracy. Because the face cameras are fixed in the prototype headset, determining a reliable bounding box for the face is straightforward. Ground-truth landmark positions were obtained by applying the detector to a separate front-facing external view, computing the 3D landmark positions using the approach from Bulat and Tzimiropoulos (2017), and projecting these points into the face-oriented views using the checkerboard tracking method of Subsection 4.6.3.

3D Model Fitting. Once the detected 2D facial landmarks are obtained, the low-quality face mesh is deformed to fit the two side camera images by minimizing the reprojection errors of the model's corresponding 3D landmarks. Specifically, for a given time instant, the pose T, shape α_s, and expression α_e of the morphable model are optimized to fit the detected 2D landmarks. In practice, the shape and pose of the face are nearly constant in relation to the viewing cameras; however, it is found that the egocentric facial capture results improved slightly by optimizing these values on a per-frame basis.

For each frame, the optimization iteratively minimizes a separate cost function for each parameter type (pose, shape, and expression). The pose cost function is the sum of errors for the left and right cameras (indexed as 1 and 2):

$$E_{pose} = \sum_{i \in L_1} ||y_i - \Pi_1 \left(TV_i\right)||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_2 \left(MTV_j\right)||_2^2, \tag{4.12}$$

where y_i is the i-th detected 2D landmark, V_i is the corresponding labeled vertex, Π_c denotes the projection function of camera c, M is the relative transformation matrix between the two face-oriented cameras, and L_c denotes the set of visible landmarks in camera c. T is thus optimized by minimizing the reprojection errors between each y_i and the projection of its 3D correspondence V_i.

With a fixed M, we found that the pose solution sometimes converged to a local minimum, which led to inaccurate shape and expression parameters. Thus, the M constraint is relaxed by computing a face pose for each camera separately, and added a term to limit their transformation matrix to be as close as T_r as possible:

$$E'_{pose} = \sum_{i \in L_1} ||y_i - \Pi_1(T_1 V_i)||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_2(T_2 V_j)||_2^2 + ||M' - M||_2^2, \tag{4.13}$$

where T_1 and T_2 are camera-specific face pose estimates, and $M' = T_2 T_1^{-1}$. After optimization, $T := T_1$ is set.

Having computed the pose matrix T, the shape and then expression parameters are independently optimized. For the shape parameters, which are initialized according to the pre-scan, the cost function is

$$E_{shape} = w_l E_l + w_{sparse} E_{sparse} + w_{sym} E_{sym} + w_{smooth} E_{smooth}. \tag{4.14}$$

The first term is similar to the pose cost function, minimizing reprojection error of the corresponding 2D and 3D landmarks:

$$E_l = \sum_{i \in L_1} ||y_i - \Pi_K T(\bar{V} + A_s \alpha_s)_i||_2^2 + \sum_{j \in L_2} ||y_j - \Pi_K T_r T(\bar{V} + A_s \alpha_s)_j||_2^2, \tag{4.15}$$

where $\bar{V}$ is the base shape of the morphable model, and A_s is the model's shape basis matrix. The subscript i denotes the i^{th} deformed vertex.

The second term is a regularizing term to constrain the number of active shape parameters:

$$E_{sparse} = \sum_{i=1}^{N_s} |\alpha_s^i|, \tag{4.16}$$

where N_s is the total number of shape parameters.

The third term enforces vertical symmetry for each left face landmark i with a corresponding right face landmark j:

$$E_{sym} = \sum_{(i,j)} \left| (\bar{V} + A_s\alpha_s)_i - (\bar{V} + A_s\alpha_s)_j \right|_y^2, \tag{4.17}$$

where $|\cdot|_y$ is the distance measured only in the y direction.

The final term smoothes the parameters for consecutive frames:

$$E_{smooth} = \sum_{i=1}^{N_s} ||\alpha_s^t - 2 \cdot \alpha_s^{t-1} + \alpha_s^{t-2}||_2^2 \tag{4.18}$$

where α_s^t denotes the frame index for the current frame.

Once the shape parameters have been estimated, the fitting of expression parameters α_e is repeated using the same terms as Equation 4.14, and incorporating the shape estimate for Equation 4.15 and Equation 4.17. $w_l = 1, w_{sp} = 4, w_{sy} = 1, w_{sm} = 1.5$ are selected for the shape cost function and $w_l = 1, w_{sp} = 6, w_{sy} = 1, w_{sm} = 0.8$ are for the expression cost function, with α_e initialized to zero each frame. Figure 4.15 shows the results of 3D face fitting. For each frame, the fitted parametric model is transferred back to the high-quality pre-scanned user model using the approach from Sumner and Popović (2004).

4.7.2 Audio Enhancement for Face Reconstruction

Full-face reconstruction relying solely on egocentric views is challenging due to the oblique viewing angles. For example, the model expression parameters are highly influenced by small errors in the landmark detections for the mouth, yet the mouth is only partially visible in each

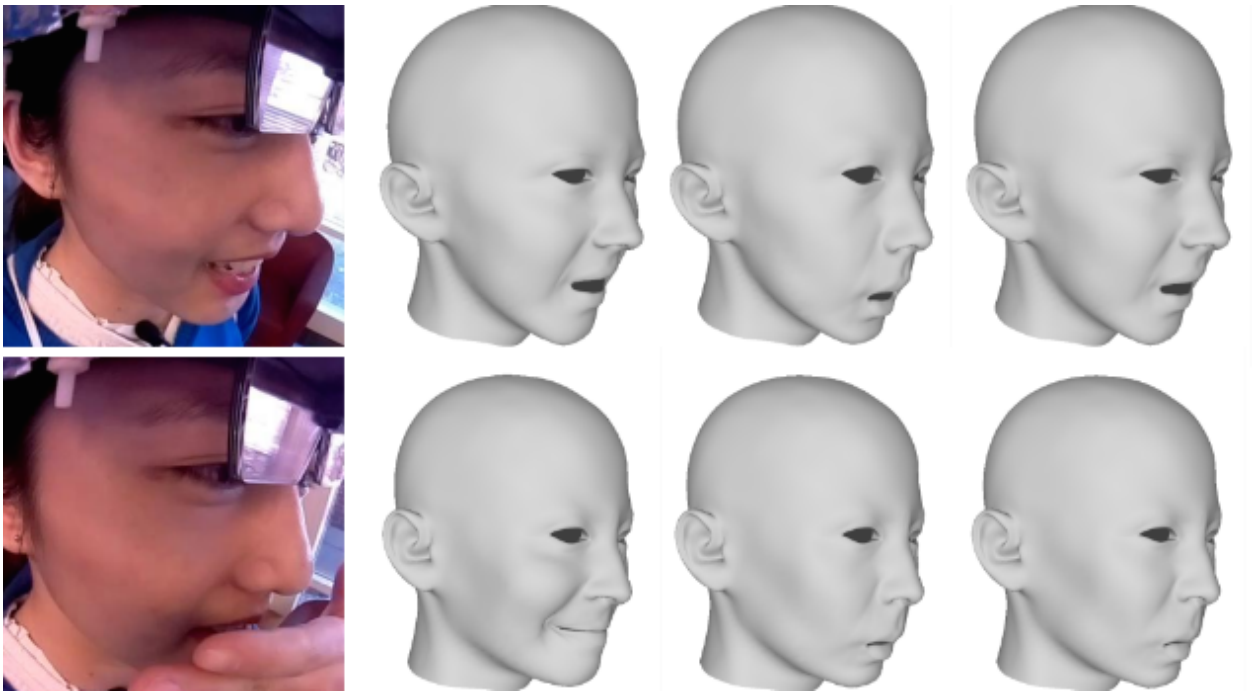


Figure 4.9: Two video/audio-based fitting results. The first column shows the original image captured by the right-side camera, and the second and third columns respectively show reconstruction results using only video or audio. The last column shows the final result of combining video and audio. The top row shows a result where the face is unoccluded; in this case, the combined result closely matches the video-only result. The second row demonstrates the contribution of audio-based capture when the face is partially occluded.

view. Moreover, video-based reconstruction is hindered if the face is (partially) occluded (e.g., see the bottom example in Figure 4.9). These problems can be addressed by augmenting the face reconstruction with geometry derived from the captured audio. Liu et al. (2015) presented a real-time facial tracking and animation approach that uses audio data to augment reconstruction from a single RGB-D camera. This neural network-based approach is adapted for the egocentric scenario.

Network Training. First, the video-based expression parameters α_e are computed as ground truth from front-facing videos with audio. Then, the corresponding audio features are extracted following Karras et al. (2017). For every video frame, a $520ms$ audio window is used; it consists of 64 overlapping audio frames, each $16ms$ in length. Audio features consisting of 32 Linear Predictive Coding (LPC) coefficients are calculated for every audio frame. Thus, the input features for each audio window is a 64×32 image, which serves as the input to the neural network.

A modified VGG-16 network architecture from Simonyan and Zisserman (2015) is used. The last 6 convolutional layers and 2 pooling layers are dropped for small-sized input signals, and the output size of the last fully-connected layer is set to 16, which corresponds to the first 16 expression coefficients. A weight ω_e^a is also inferred for every time instant, representing the confidence of the audio result, as follows. Silent frames in the data are first detected by checking the 600 ms window around each time instant. If all converted wave values in the window are below a threshold, it is called a silent frame. Non-silent frames are assigned a "full-audio" weight $\omega_e^a = 1$, and for silent frames, ω_e^a is determined by the length of time to the nearest non-silent frame.

4.7.3 Combining Video and Audio

Similar to Liu et al. (2015), the audio-estimated expression parameters α_e^a and the video parameters α_e^v are combined to compute the final frame parameters α_e:

$$A_e\alpha_e = WA_e\alpha_e^a + (I - W)A_e\alpha_e^v, \tag{4.19}$$

Figure 4.10: Face re-targeting result. The two left images show the input deformed face model and the re-targeted face part of the pre-scan, respectively. The face vertices of the body model are replaced by their deformed counterparts, as shown in the right image.

where $W \in \mathbb{R}^{3N \times 3N}$ is a diagonal weighting matrix, and N is the number of vertices in the morphable model. Differently from Liu et al. (2015), a weight map around the mouth landmarks is computed and multiply it with weights inferred from the audio neural network ω_e^a as the final weights of every vertex. During the combination step, occlusion of the mouth is also considered. If the landmarks detection result has a large difference between two consecutive frames around the mouth, the video-based mouth weights are negated, relying strictly on audio for that region. The result of combining video and audio is shown in Figure 4.9.

4.7.4 Facial Motion Re-targeting

During capture, pose, shape, and expression coefficients of the 3D morphable face model are estimated as in the previous subsection. For visualization, a method is required to deform the pre-scan face mesh according to this transformation. Because the face part of the pre-scan is not rigged, a deformation transfer (Sumner and Popović, 2004) from the face model to the pre-scan is employed. It minimizes the differences of the corresponding triangle deformations between the face-model mesh and the pre-scan mesh (first and second images in Figure 4.10, respectively).

Let S be the face-model mesh after shape-based alignment to the pre-scan; denote its 3D vertices as $\{s_1 \ldots, s_n\}$ and its triangles as $\{(a_1, b_1, c_1), \ldots, (a_m, b_m, c_m)\}$, where the (a_j, b_j, c_j) indexes

three vertices. Let $\widetilde{\mathbf{S}}$ denote the deformed face-model mesh using the estimated coefficients for a given frame; it has vertices $\{\widetilde{s}_i\}$ and the same triangles as $\mathbf{S}$.

As outlined in Sumner and Popović (2004), the affine transformation for a triangle j in $\mathbf{S}$ to its corresponding triangle in $\widetilde{\mathbf{S}}$ can be defined as $\mathbf{Q_j} = \widetilde{\mathbf{E}}_\mathbf{j}\mathbf{E_j}^{-1}$. Here, $\mathbf{E_j} \in \mathbb{R}^{3\times3}$ is the *edge matrix* for triangle j, defined as

$$\mathbf{E_j} = [(s_{b_j} - s_{a_j}) \;\; (s_{c_j} - s_{a_j}) \;\; n_j], \tag{4.20}$$

where n_j is the unit normal for the triangle. $\widetilde{\mathbf{E}}_\mathbf{j}$ is similarly defined.

Now, the pre-scan mesh $\mathbf{T}$ is deformed into a new mesh $\widetilde{\mathbf{T}}$ in a manner similar to the transformation of $\mathbf{S}$ into $\widetilde{\mathbf{S}}$. Assume, for the moment, that for each triangle j in $\mathbf{S}$, the corresponding triangle ℓ in $\mathbf{T}$ is known. (It will be explained how to obtain these correspondences below.) Using deformation transfer, it is optimized for the vertices $\{\tilde{t}_k\}$ of $\widetilde{\mathbf{T}}$ by encouraging the affine transformations $\{\mathbf{Q}'_\ell\}$ of the triangles of $\mathbf{T}$ to match their counterparts $\{\mathbf{Q_j}\}$ of $\mathbf{S}$:

$$\min_{\widetilde{t}} \sum_{(j,\ell)\in\mathbf{C}} ||\mathbf{Q_j} - \mathbf{Q}'_\ell||_F^2, \tag{4.21}$$

where $\mathbf{C}$ is the set of triangle correspondences, and $||\cdot||_F$ denotes the Frobenius matrix norm.

Computing triangle correspondences. The correspondences between the triangles of $\mathbf{S}$ and $\mathbf{T}$ are computed in a pre-processing step that first aligns the 3D landmarks $\mathbf{S}$ with $\mathbf{T}$ while encouraging smoothness of the triangle deformations of $\mathbf{S}$. Once this alignment is achieved, triangle correspondences are obtained based on nearest neighbors. The landmark correspondences in this section are the same as those used for the initial landmark-based model fitting.

Specifically, consider aligning the landmarks of $\mathbf{S}$ and $\mathbf{T}$ by deforming the vertices of $\mathbf{S}$. In a slight abuse of earlier notation, the vertices of $\widetilde{\mathbf{S}}$ are optimized so that they match $\mathbf{T}$:

$$E_{lm}(\{\widetilde{s}_i\}) = \sum_{(i,k)\in\mathbf{L}} ||\widetilde{s}_i - t_k||_2^2, \tag{4.22}$$

where $\mathbf{L}$ is the set of corresponding landmark-vertex-index pairs for the two meshes.

Equation 4.22 needs to be regularized to ensure smooth triangle deformations of $\mathbf{S}$ into $\widetilde{\mathbf{S}}$. To do this, the neighborhoods of the triangles in $\mathbf{S}$ are considered, such that their deformation is similar:

$$E_{ne}(\{\widetilde{s}_i\}) = \sum_{j=1}^{m} \sum_{r \in \text{adj}(j)} ||\mathbf{Q_j} - \mathbf{Q_r}||_F^2, \tag{4.23}$$

where adj(j) denotes the set of triangles sharing an edge with triangle j in $\mathbf{S}$, and m is the total number of triangles in $\mathbf{S}$.

Additionally, to avoid over-fitting, the presence of strong deformations is penalized for each triangle:

$$E_{id}(\{\widetilde{s}_i\}) = \sum_{j=1}^{m} ||\mathbf{Q_j} - \mathbf{I}||_F^2, \tag{4.24}$$

where $\mathbf{I}$ is the identity transformation.

The final cost function for the fit is the sum of Equation 4.22-Equation 4.24:

$$E(\{\widetilde{s}_i\}) = E_{lm}(\{\widetilde{s}_i\}) + E_{ne}(\{\widetilde{s}_i\}) + E_{id}(\{\widetilde{s}_i\}) \tag{4.25}$$

Figure 4.10 shows an example of the re-targeting result for the face.

4.8 Device Tracking and Environment Reconstruction

The headset device is fitted with four outward-facing cameras that serve to track the motion of the wearer within their environment while simultaneously reconstructing their surroundings. This reconstruction capability is an important component for the overall capture scenario: the wearer's environment provides context for remote observers and greatly contributes to their sense of "being there." While device tracking is ultimately necessary for the system as a *motion capture* unit, the external reconstruction endows the device with the ability for *content capture*.

In the prototype system, environment capture is performed using four synchronized views on the sides and back of the wearer's head, and processing is performed offline. From this multi-view imagery, the motion of the camera rig is estimated simultaneously with reconstructing the environment using COLMAP for incremental SfM (Schonberger and Frahm, 2016) and multi-view stereo (MVS) (Schönberger et al., 2016). The process of SfM has three stages: feature extraction for individual images, feature matching between image pairs, and reconstruction. During reconstruction, images are iteratively registered to each other based on their feature correspondences; here, registration involves computing the rotation and translation of the image relative to the environment, as well as 3D scene points for the individual image features. Since the camera rig is pre-calibrated for both intrinsics and local extrinsics, a to-scale registration of the cameras to the scene can be obtained via SfM in an unsupervised fashion. Given these camera registrations, MVS is used to estimate a dense (pixel-wise) depth map for each image, and then depth-map fusion and subsequent surface meshing (Kazhdan and Hoppe, 2013) are employed to obtain the final environment model.

The outcome of this offline processing is a textured 3D mesh depicting the user's environment, as well as information about where the user was standing and where they were looking relative to the environment at each time-point in the capture. When visualizing the capture in, e.g., virtual reality, this information is directly used to place the animated reconstructed body model within the virtual environment.

4.9 Integration

The resulting face, body, and environment reconstructions are integrated to compose the entire scene (Figure 4.11). First, the face vertices in the body pre-scan are replaced using Equation 4.21. Then, the pre-scan is deformed using Equation 4.10. The deformed pre-scan $\widehat{\mathbf{T}}_{\text{local}}$ in model space is localized to the environment coordinates using the estimated headset pose $C_t \in \mathbb{R}^{4\times4}$ at time t.

$$\widehat{\mathbf{T}}_{\text{global}} = C_t \begin{bmatrix} \mathbf{R}_M^{-1} & \mathbf{R}_M^{-1}\mathbf{J}_{\text{head}} \\ \mathbf{0} & 1 \end{bmatrix} \widehat{\mathbf{T}}_{\text{local}}, \tag{4.26}$$

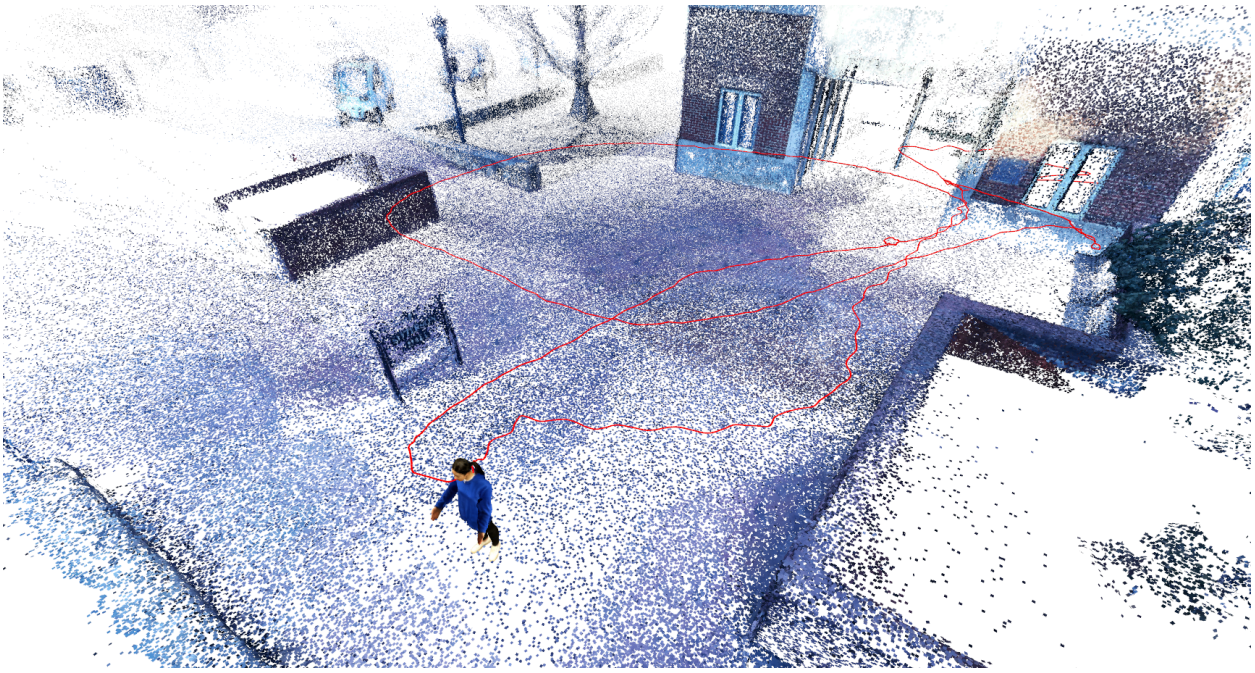


Figure 4.11: Integration result. The deformed pre-scan is placed into the reconstructed environment using headset tracking. The entire path of the tracked headset is shown in red.

where $\mathbf{R}_M$ is the rotation of the body model estimated during the skeleton alignment in Subsection 4.6.4 and $\mathbf{J}_{\text{head}}$ is the head joint position. $[\mathbf{R}_M^{-1}|\mathbf{R}_M^{-1}\mathbf{J}_{\text{head}}]$ reorients the pre-scan to its head joint at the origin in local space.

Because the feet are often occluded in the downward-facing views, the leg motions of the wearer are rarely detected. The feet are modeled in 3D as located on the ground, exactly below the knees in Subsection 4.6.2. To compensate for this, a motion prior is added to the pre-scan deformation based on the norm of average velocity $V_t = ||d/\Delta_t||$ of head-track displacement d. Specifically, a separate walking motion pose sequence $\{\theta_{\text{walk},t}\}$ is captured, including two full strides of an individual. This step sequence is looped continuously throughout the capture sequence. For a given frame t, the refined pose $\widehat{\theta}_t$ is estimated as,

$$\widehat{\theta}_t = \alpha_t\, \theta_{\text{walk},\,t} + (1 - \alpha_t)\, \theta_t\,;\ \alpha_t = \min(V_t, 1). \tag{4.27}$$

Table 4.1: Egocentric Human Pose Dataset, in number of synchronized frames. The train data consists of 6 sequences collected using 6 external cameras in a capture studio described in Subsection 4.6.3

	Train Data Size	Test Data Size	Indoor Data Size	Outdoor Data Size
Frames	32,896	3,010	1,760	1,250

The blended pose $\widehat{\theta}_t$ is controlled by velocity V_t. When the user moves quickly, the influence of the walking motion increases. When the user stops walking, the motion becomes negligible. Figure 4.7 shows a result of the pose blending.

4.10 Results

In this section, the results for the body pose and facial expression estimation pipelines are presented. Additionally, a possible use case for the system in this chapter is showcased: virtual tours of a remote place (indoors and outdoors), with the wearer of the device acting as a tour guide.

None of the existing datasets were directly suitable for training and evaluation using the prototype headset introduced in this chapter since they lacked egocentric video data with similar viewpoints for both face and body observations. To evaluate the body pose and facial expression estimation approaches discussed in this chapter, we collected 6 sequences using 6 external cameras for training (32k frames) and 2 sequences for evaluation (3k frames) with users wearing the prototype headset. The ground truth full-body 3D joints were acquired using multiple fixed cameras in a capture studio as described in Subsection 4.6.3. The summary of the dataset is shown in Table 4.1.

4.10.1 Results for Body Visibility Simulation using head-worn egocentric cameras

To explore the head-worn egocentric camera placement, a room-sized environment was simulated with static objects such a whiteboard, a desk, and chairs. A simulated user was animated over a 60-second sequence to perform actions such as sitting on a chair, getting up, and writing on the whiteboard. The egocentric cameras were modeled in a similar configuration as the physical

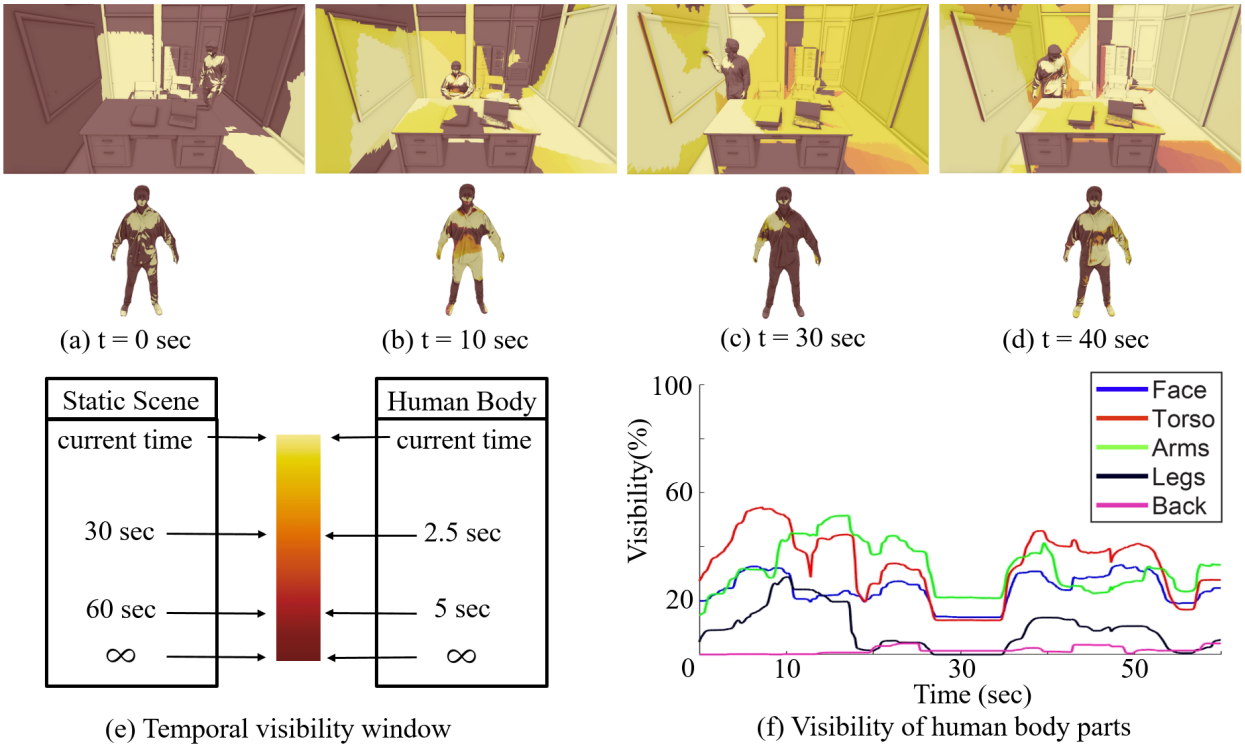


Figure 4.12: Environment and body part visibility simulation for head-worn egocentric camera modeling. (a-d) Temporal visibility heat maps using only head-worn cameras for the static scene (top) and user's body (bottom). (e) Color coding heat map for (a-d). Surfaces are colored according to how recently they were visible to one of the head-worn cameras. (f) Time plot of visibility percentages for several parts of the simulated user's body.

prototype, with each simulated camera's horizontal field of view set to $90°$. The method introduced by Chabra et al. (2017) was used to model temporal visibility of a surface in the simulation:

$$v_t = \begin{cases} 1 & \text{if } s \text{ is visible from at least one camera at time } t \\ 1 - \frac{\Delta t}{\tau} & \text{if } s \text{ is hidden for a time } \Delta t < \tau \\ 0 & \text{otherwise} \end{cases} \tag{4.28}$$

In the analysis, the temporal visibility threshold interval τ is set to 5 seconds for dynamic objects and to 60 seconds for static objects. The resulting temporal visibility v_t is shown in Figure 4.12 as heat maps at 4 different time instants, with the brightest color representing polygons that were visible most recently. The percentage of visible polygons over time is also shown for the virtual person's body. The noticeable drop in visibility around time $t = 30$ corresponds to the interval during which the person was writing on the whiteboard, remaining relatively motionless.

The simulation results indicate that with the specified egocentric camera arrangement, most of the dynamic scene is visible to at least one camera within reasonable visibility threshold intervals, which provides confidence that the reconstruction approach can successfully reconstruct a near-static environment. However, the results of this simulation led us to use larger FoV lenses for body and face capture in the physical prototype (120 degrees horizontal) than in the simulation (90 degrees horizontal), and smaller FoV lenses for environment capture (62 instead of 90 degrees).

4.10.2 Results for Body Pose Estimation

In addition to qualitatively evaluating the body pose estimation on demo data, qualitative and quantitative analyses are provided on a validation dataset that was captured in the same environment as the training dataset in Table 4.1, independently but using similar motions. Figure 4.13 shows results for the 2D joint detection and 3D pose estimation on two typical poses from the validation dataset: walking and sitting while gesticulating. Qualitatively, the results exhibit satisfactory alignment with the ground truth. In Figure 4.14, qualitative results are shown for the 2D joint

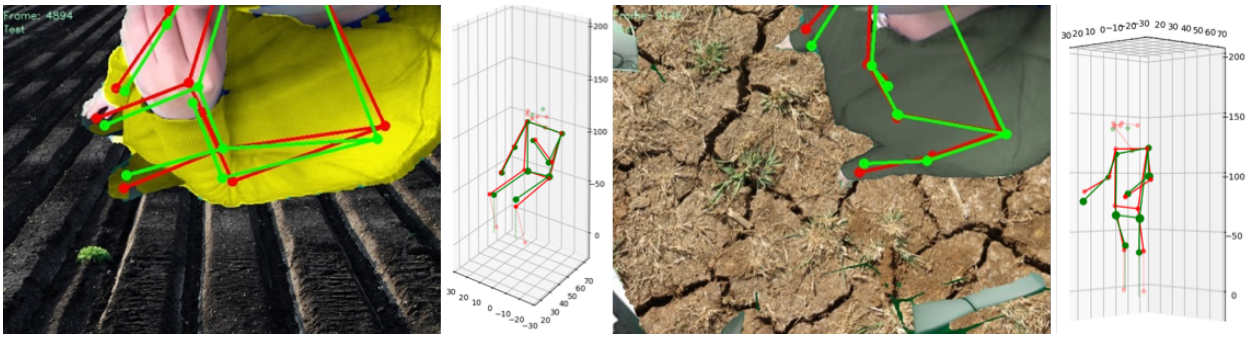


Figure 4.13: Example 2D and 3D pose estimation results on the validation dataset. Red points and lines show the ground-truth joints positions and skeleton in the images, while those in green are the prediction results. Left: Sitting pose. Right: Walking pose. In each image, the background and shirt color have been synthetically augmented.

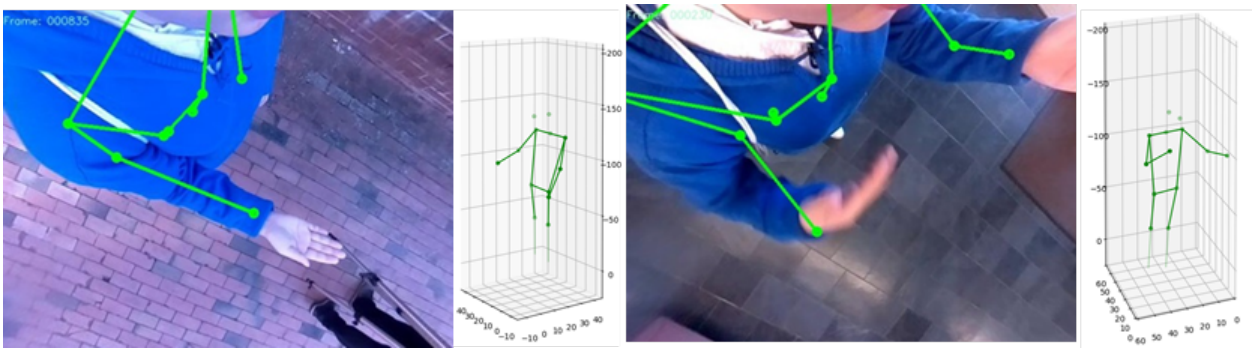


Figure 4.14: Example 2D and 3D pose estimation results for the outdoor (left) and indoor (right) video tour scenes. Green points and lines show the predicted skeleton. Note that the mis-predicted right arm in the right image is corrected in 3D using the introduced RNN.

Table 4.2: 2D and 3D joint estimation errors for the method introduced in this chapter. 2D: Mean and standard deviation pixel errors for detected 2D joints. 3D: Mean 3D distance (in cm) between the ground-truth and predicted joint positions for two-view triangulation from the body cameras (Tri.) and the introduced recurrent approach (RNN). Notation: Shoulder (S), elbow (E), wrist (W), hip (H), and knee (K). R/L: Right/left joint.

		RS	RE	RW	LS	LE	LW	RH	RK	LH	LK
2D (px)	Avg	11	5.9	7.1	11	7.9	8.5	4.5	7.1	4.5	5.9
	Std	12	7.4	12	12	8.5	14	4.5	11	3.8	9.1
3D (cm)	Tri.	5.9	3.4	4.0	6.2	3.7	6.2	3.7	6.1	3.6	5.9
	RNN	3.7	2.9	3.3	4.3	3.0	4.7	2.1	4.0	2.0	3.9

detection and 3D pose estimation on the demo test dataset in both outdoor and indoor scenes in Table 4.1. The indoor result shows an example that a reasonable 3D pose can be obtained despite imperfect 2D detections (right arm in the right image).

Table 4.2 provides a quantitative analysis for the validation dataset, including 2D errors in joint detection and 3D errors in joint position estimation. For 2D detections, mean and standard deviation errors in pixels are reported. The input images are 640×480 px. Skewed error distributions are generally observed with the majority of detections closer to the ground truth than the mean. Sporadic large detection errors arise from false-positive maxima in the belief maps output by the CPM. These detection errors are typically corrected during the subsequent 3D prediction and motion smoothing. Regarding 3D skeleton errors, the performance of two methods is evaluated: 1) simple two-view triangulation using the known relative calibration of the downward-facing views, and 2) the introduced RNN approach. The RNN approach has lower positional error for all joints, with average validation errors between 2cm and 4.7cm in Table 4.2. The result compares favorably to EgoCap (Rhodin et al., 2016), the existing system most similar to the system presented in this chapter, for which average 3D joint position errors of 7 ± 1cm were reported. These averages roughly follow the general visibility of the joints in each view, with the hips and elbows having the lowest errors.

Figure 4.15: 2D face landmark detection and 3D facial fitting. White points in the first column show the 66 2D landmarks of the indoor image (top) and the outdoor image (bottom) respectively. The second column shows the mesh fitting visualization with all mesh vertices (green) projected into the images.

4.10.3 Results for Face Reconstruction

The top row of Figure 4.15 shows the face landmark detection and model fitting results for both indoor and outdoor illuminations. The face model has 66 total landmarks. Due to the limited visibility in each view, the 10 midline landmarks and the 28 additional landmarks are detected for each half of the face.

To quantitatively evaluate the face reconstruction result, the distances are computed between all 66 2D and projected 3D landmarks for the complete set of frames in the indoor and outdoor virtual tour capture data in Table 4.1. The alternating pose, shape, and expression optimization run for 5 iterations. Over all frames for both views, the RMS error decreases from an average initial value of 16.38 px to an average final value of 3.57 px. To visualize the 3D fitting, the projection

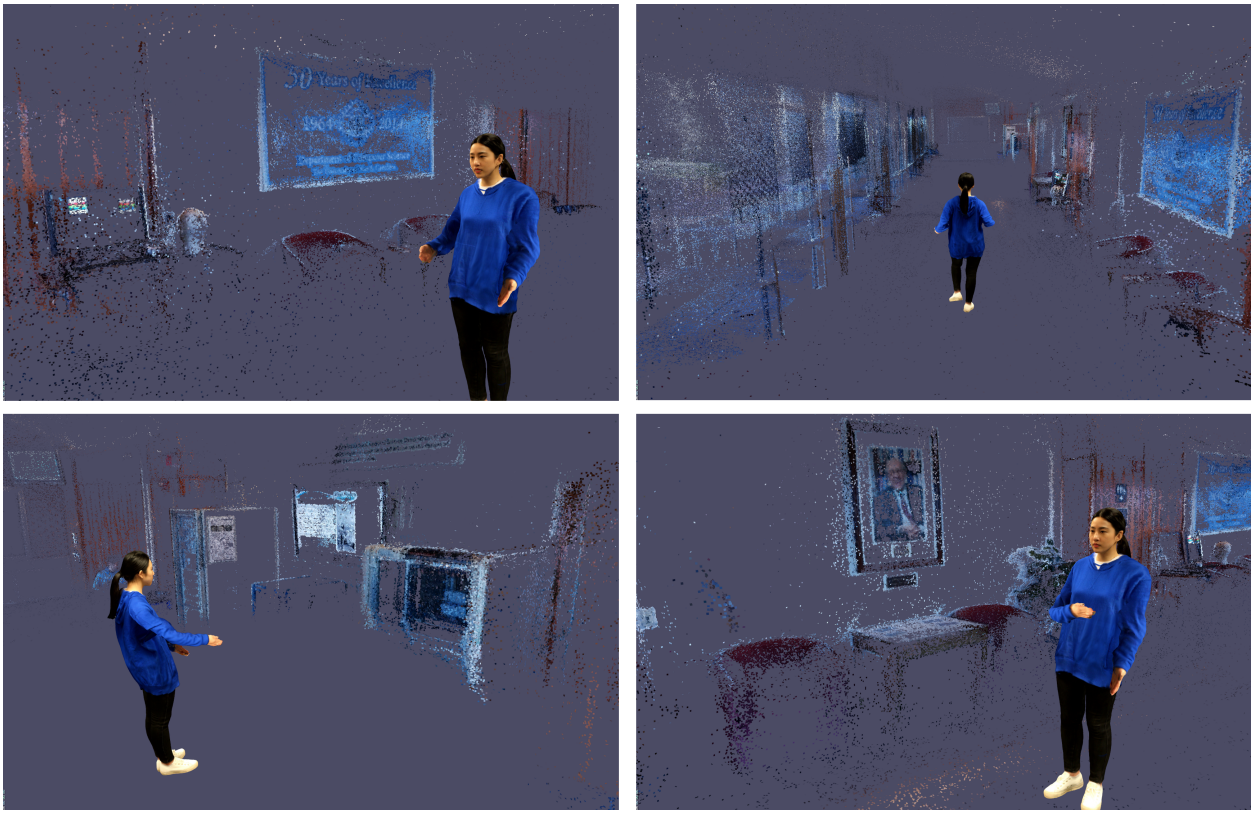


Figure 4.16: Four frames from the indoor section of virtual tour.

of the corresponding mesh onto the input imagery is shown in Figure 4.15. It is observed that the projection fits the entire face accurately, including the neck and the ears.

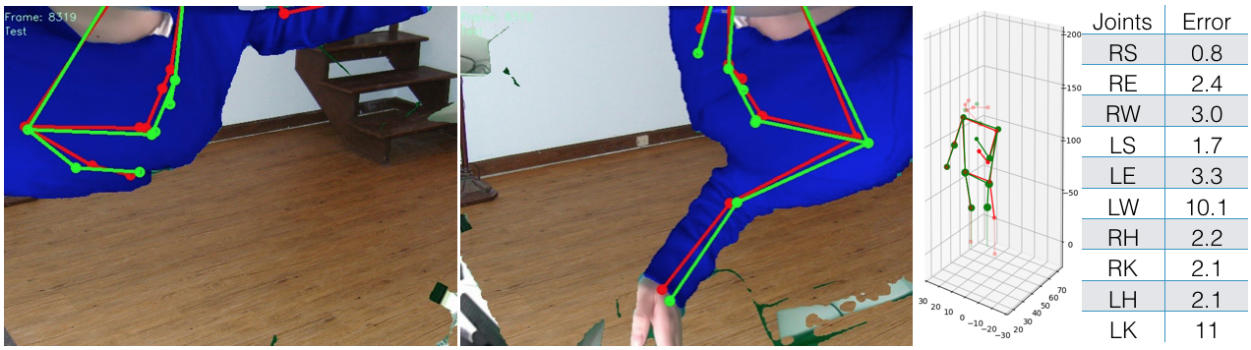
Figure 4.9 provides two examples to demonstrate the final reconstruction result of combining video and audio. The first column shows the original image captured by the side cameras; the second and third columns show separately the reconstruction results from video and audio. The final column shows the final result of combining video and audio. The audio result in the first row is unreliable due to silence and is ignored in the combined result. In the second row, the mouth is occluded, causing an unreliable video result, but the audio provides a plausible mouth shape in the combined result.

Figure 4.17: Outdoor Virtual Tour. (Top Left) External view of tour Guide. (Top Right) External view of visitor. (Center) Reconstruction at the visitor's point of view.

4.10.4 Application: Virtual Tour

To demonstrate the potential of the system for ego-capture scenarios, the headset device is used to record a short VR tour of the UNC Department of Computer Science. Acting as a tour guide, the wearer moves around the capture space and describes her surroundings. The system then reconstructs the wearer's motions and environment, creating a dynamic 3D representation that remote users can experience in VR, as if they were getting an in-person tour. Figure 4.16 shows example frames from the indoor portion of the tour, and a view of the outdoor portion is shown in Figure 4.17.

For real-time visualization, the animated sequence of per-frame body poses is built into an Alembic geometry cache (Alembic, 2010) using Autodesk Maya 2018, which is then represented as an animated non-skeletal 3D mesh in Unreal Engine 4. The viewer wearing the head-mounted display is provided controller-based locomotion in addition to physical locomotion to walk with the reconstructed tour guide in a reconstructed virtual environment that is larger than the available physical environment.



Joints	Error
RS	0.8
RE	2.4
RW	3.0
LS	1.7
LE	3.3
LW	10.1
RH	2.2
RK	2.1
LH	2.1
LK	11

Figure 4.18: 2D and 3D pose estimation result where the left wrist cannot be seen from the right-side camera, and the knees are barely visible in either camera. Such a situation can result in large errors for the system: the 3D error of the left wrist is 10.13cm, while the error of the right wrist is 2.94cm.

4.11 Conclusion and Future Work

This chapter presented the egocentric 3D capture of an individual and their environment without relying on any instrumented environment but relying only on cameras and sensors worn by the individual. This approach allows for the reconstruction and communication of experiences from any location, indoors or out. With a vision of the fully mobile capture systems of tomorrow, I outlined the key technological advances necessary for capturing the wearer's body pose, facial expression, and limbs—entirely from near-body views—and I also showed how the surrounding environment could be reconstructed using outward-facing views, which enables completely egocentric content capture. The results demonstrate workable methods that leverage state-of-the-art machine learning approaches to overcome the profound problems of poor visibility for body capture from head-worn cameras.

One limitation of this system is that it captures the raw data in real time and processes it offline. This inspired us to try to develop techniques accelerating to real time for the interaction between people in two different places as well as integrating the capture and processing components of the system into a wearable package, e.g., a backpack connected to the headset, in order to allow telepresence-type interactions. These results are described in Chapter 5.

A key limitation in pose estimation is that the approach is user-specific. This also inspired us to train the neural network with data from multiple users and increase the amount of variation in

training data to make the approach more broadly applicable. For example, improving train data generalization can improve the detection of unseen joints, such as the left arm in Figure 4.18 or the ankles. Another limitation is the accuracy for pose estimation for legs is significantly worse when the leg joints are occluded. This also inspired us to add body-worn inertial sensors to be better reconstructed. This result is described in Chapter 5.

With respect to device tracking and environment reconstruction, the main direction for next work is to reconstruct dynamic environments. Adding front-facing external cameras would improve observations of moving objects as well as user comfort since it is easier for the user to know what parts of the environment have been captured when those views line up with their line of sight. To improve VR visualization of the environment reconstruction, exploring meshing techniques extracting a mesh from point clouds is also encouraging to obtain better 3D environment mesh.

The user reconstruction part of the system also offers directions for research. The body re-targeting technique uses a body model with limited degrees of freedom in motions. Employing a recent body model such as SMPL (Loper et al., 2015) or SMPLX (Romero et al., 2017) can be used to obtain more natural body movements with more degrees of freedom in motions. Similarly, the face re-targeting approach uses deformation transfer, which results in limited facial expressions. Using a rigged face model can yield more-natural-looking facial animations. Another research direction is to fully model hand and finger motions and enable capture and reconstruction of arbitrary objects being carried or manipulated. Finally, it is also can be explored using mirrors to allow reconstruction of the user's body model directly from images captured using the headset-mounted cameras, rather than requiring a separate body pre-scan process.

CHAPTER 5: MOBILE HUMAN MOTION RECONSTRUCTION USING ONLY EYEGLASSES-MOUNTED CAMERAS AND A FEW BODY-WORN INERTIAL SENSORS

Toward a convenient telepresence system available to users anywhere, anytime, mobile 3D capture systems require displays and sensors embedded in commonly worn items such as eyeglasses, wristwatches, and shoes. To this end, this chapter describes a learning-based method for egocentric human pose estimation using only eyeglasses-mounted cameras and sparse body-worn inertial sensors worn on the wrists and ankles for widespread acceptability. The method in this chapter overcomes challenges such as inconsistent limb visibility in eyeglasses form factor views and pose ambiguity due to a small number of IMUs by learning the visibility-awareness of joints and the temporal correlations between instrumented and non-instrumented body parts. The experimental results demonstrate the system by reconstructing various human body movements and show that the learning-based visual-inertial fusion method for 3D pose estimation, which runs in real time, outperforms both visual-only and inertial-only approaches.

This chapter is based mainly on "Mobile, Egocentric Human Body Motion Reconstruction Using Only Eyeglasses-mounted Cameras and a Few Body-worn Inertial Sensors", Young-Woon Cha, Husam Shaik, Qian Zhang, Fan Feng, Adrian Ilie, Andrei State, and Henry Fuchs, published in IEEE Virtual Reality (VR), March, 2021. [1] [2]

5.1 Introduction

Telepresence enables remote social interaction without physical presence. 3D display greatly enhances the sense of presence but requires the ability to fully capture and reconstruct human

[1] Cha et al. (2021)

[2] A Best Conference Paper Award: `https://ieeevr.org/2021/awards/conference-awards` Accessed: 2021-06-01

subjects as well as their environment. I expect 3D capture of user experiences to become a feature of common head-worn devices with the form factor of conventional eyeglasses to be worn all day like ordinary eyeglasses. With widely available wearable technology embedded in commonly worn accessories (cameras in eyeglasses, IMUs in wristwatches and shoes), a mobile 3D acquisition and display system such as the one in Figure 5.1 (right) will enable 3D telepresence. Inspired by the success and to overcome the limitation of the system described in Chapter 4, this chapter introduces approaches for better limb pose estimation and for real-time capability to enable interaction between remote participants.

One of the challenges of targeting an eyeglass-frame form factor is that the user's limb motions are frequently unobservable by the cameras due to occlusion, or to being outside of the camera views, as illustrated in Figure 5.2 and Figure 5.3. This problem makes many prior pose estimation methods inapplicable to situations. For example, per-frame visual 3D pose estimation methods can produce unreliable estimates for occluded joints (Cheng et al., 2019) due to incomplete visibility. Similarly, while human performance capture approaches that use external cameras have achieved high accuracy and real-time performance (Habermann et al., 2019; Xiang et al., 2019; Kocabas et al., 2020; Habermann et al., 2020), they require all joints to be visible. Joint heatmap estimation methods (Cao et al., 2019; Newell et al., 2016; Zhang et al., 2019) are also unable to handle the joints that are outside the image because they cannot be labeled within the 2D heatmap. Extending the heatmap size by padding the boundary is likely to generate high 3D joint errors due to the high distortion of wide-FoV or fisheye lenses. Finally, prior egocentric capture headgear (Rhodin et al., 2016; Cha et al., 2018; Xu et al., 2019; Tome et al., 2019) featured cameras mounted farther away from the face; while they offer better body and limb visibility, they are obtrusive and thus unacceptable for daily use.

Another challenge is reducing the number of IMU sensors for widespread acceptability. Prior visual-inertial fusion approaches for 3D pose estimation (Von Marcard et al., 2016; Malleson et al., 2017; von Marcard et al., 2018) require more than 10 body-worn sensors, a number unlikely to be accepted for general use, even with miniaturization. Reducing that number results in pose

Figure 5.1: Mobile, egocentric real-time body motion capture system using only eyeglasses-mounted cameras and a few body-worn inertial sensors. Fast body motion reconstructions of indoor and outdoor user (a), shown in VR (b). Current mobile user (c), and future vision (d) depicting casual everyday use of streamlined system with miniaturized cameras embedded in the frames of wide-field-of-view AR eyeglasses, and IMUs on wrists and in shoes.

ambiguities and lower accuracy for non-instrumented body parts (Tautges et al., 2011; von Marcard et al., 2017; Huang et al., 2018). For example, a knee raise cannot be reliably distinguished from a standing pose, as the IMU data is insufficient for inferring thigh orientation if no sensor is worn on it.

This chapter presents a wearable 3D acquisition system for real-time 3D mobile telepresence relying only on eyeglass-frame-mounted cameras and IMUs on wrists and ankles. This approach allows for convenient, unobtrusive reconstruction and communication of experiences at any indoor or outdoor location. To support the vision of such a fully mobile capture system, the wearer's 3D body pose is captured using learning-based visual-inertial sensor fusion. Unlike methods that rely on instrumented environments (Habermann et al., 2019; Yu et al., 2019), this enables completely self-contained egocentric content capture and overcomes inconsistent limb visibility, as well as IMU pose ambiguity caused by sparse IMUs.

The approach consists of three components that allow visual and inertial measurements to complement each other when tracking joints. First, a *visibility-aware visual 3D pose network* estimates visible 3D joints while suppressing unreliably detected occluded joints. Second, an *online IMU offset calibration method* improves the inertial measurements by aligning the visual and inertial bone orientations, over time, for forearms and lower legs with attached IMUs. Third, a *visual-inertial 3D pose network* estimates the poses of upper arms and thighs without IMUs by using a sequence of inertial measurements of the corresponding lower bones, as well as visual detection of the upper bones in previous frames. At each instant, the estimated body pose is re-targeted to a human surface model, resulting in a high-fidelity reconstruction of the user. The full-body pose, including 3D joint locations as well as 3D bone orientations, is estimated continuously and kept temporally coherent, even when some joints are out of image or occluded.

The system presented in this chapter is demonstrated on reconstructions of various human body movements in a remotely assisted physical therapy scenario, and its mobile capability is shown in an outdoor scenario. For training and evaluation, a new large-scale egocentric visual-inertial 3D human pose dataset is collected. None of the existing datasets includes occlusion, out-of-image labels in

egocentric views, and densely worn inertial sensors. The collected dataset is made publicly available at EgoVIP Dataset (2021). In experiments, the learning-based visual-inertial fusion method runs in real time, at 30 Hz, on a standard PC and outperforms both visual-only and inertial-only approaches, showing significant improvements in out-of-image and self-occlusion situations.

The main contributions are:

- The first egocentric 3D human pose estimation approach that can handle both sparse visibility and sparse inertial sensors.
- A working, standalone, proof-of-concept prototype in an eyeglasses form factor for mobile capture and real-time body motion estimation.
- The first egocentric human motion dataset that includes multiple views with joint visibility information as well as inertial measurements.

5.2 Related Work

5.2.1 Body Reconstruction

Deformable body model-based surface estimation has been a focus in computer vision (Loper et al., 2015). Estimation of model parameters approximates the human surface in conjunction with visual pose estimation (Bogo et al., 2016), by estimating dense correspondences between the body model and imagery (Alp Güler et al., 2018), or by direct volumetric inference (Varol et al., 2018). Recent work shows advances in real-time performance by using temporal poses (Kocabas et al., 2020), as well as face and hand poses (Xiang et al., 2019). High-fidelity geometry can also be estimated by fitting image silhouettes (Habermann et al., 2019), or by cloth simulation (Yu et al., 2019). These approaches require full-body visibility in external camera views to be able to fit full-body shapes and poses. In egocentric views, however, body parts are often invisible.

5.2.2 Visual Pose Estimation

Recent advances in learning-based approaches for deep neural networks have shown significant improvements in accuracy when used for pose estimation. 2D joint heatmap-based estimation has been successful using Convolutional Neural Network (CNN) architectures (Wei et al., 2016; Cao et al., 2019; Newell et al., 2016; Zhang et al., 2019). CNN-based 3D joint estimations also have shown significant accuracy in real time for a single outside-in looking view (Mehta et al., 2017b, 2018, 2020). Human pose constraints (Dabral et al., 2018; Sun et al., 2018) and occlusion information (Cheng et al., 2019) have been incorporated during training. In the case of continuous human motions over time, Recurrent Neural Network (RNN)-based pose estimations have shown promising results for a sequence of motion predictions (Villegas et al., 2017; Butepage et al., 2017; Martinez et al., 2017). These approaches estimate joint locations, but 3D bone orientation estimation is still an open problem when using only visual information to estimate a full-body pose.

5.2.3 Visual Egocentric Pose Estimation

High-quality reconstruction from egocentric data captured by body-worn cameras remains a challenge, requiring reconstruction methods that operate in arbitrary, uninstrumented environments. Outside-looking-in camera-based human pose estimation methods are not directly applicable to egocentric views of the body.

Prior egocentric motion capture approaches in Subsection 4.2.4 exploited egocentric, body-worn cameras for 3D pose estimation of certain parts of the body. However, without direct observation of the body, the pose estimation accuracy is limited.

Significant improvements in egocentric full-body pose estimation have been made using downward-looking stereo head-worn views (Rhodin et al., 2016; Cha et al., 2018) or learning-based approaches using a single head-worn camera view (Xu et al., 2019; Tome et al., 2019) discussed in Subsection 4.2.4, which enable improved views of the wearer's body with wide-FoV cameras. The approaches using downward near-body views, however, have yet to fully address the challenges of

self-occlusion and out-of-view joints, which need to be resolved in order to estimate a full-body pose of the wearer solely from body-worn cameras.

5.2.4 Inertial Pose Estimation

Human pose estimation can also be performed using body-worn inertial measurement units (IMUs). IMUs can capture fast motions (Malleson et al., 2017) and track body parts that might be occluded in camera views, but they suffer from measurement noise and drift over time, and require careful calibration for the initial pose.

Even with miniaturization of sensors, using a relatively large number of worn sensors is unlikely to be widely accepted. To increase acceptability, recent approaches have attempted to reduce the number of IMUs to a sparse set by employing temporal orientations and accelerations (Tautges et al., 2011; von Marcard et al., 2017; Huang et al., 2018). The IMUs are worn only on forearm and lower leg; the missing upper arm and thigh orientations are estimated by assuming that the temporal motions of lower and upper bones are highly correlated. Inference results are promising but suffer from pose ambiguity, as multiple poses can be possible with similar measurements. This issue is addressed only partially by using more temporal measurements such as future frames or an entire sequence. To overcome this problem, visual and inertial sensor fusion (Von Marcard et al., 2016; Malleson et al., 2017; Trumble et al., 2017; von Marcard et al., 2018) leverages outside-looking-in cameras jointly with IMUs to calculate a 3D body pose. Visual pose estimates from the outside-looking-in cameras help constrain the possible 3D poses of the inertial sensors, and alleviate the IMU measurement noise (von Marcard et al., 2018). However, so far these approaches require complete body visibility, which is seldom achievable from egocentric views.

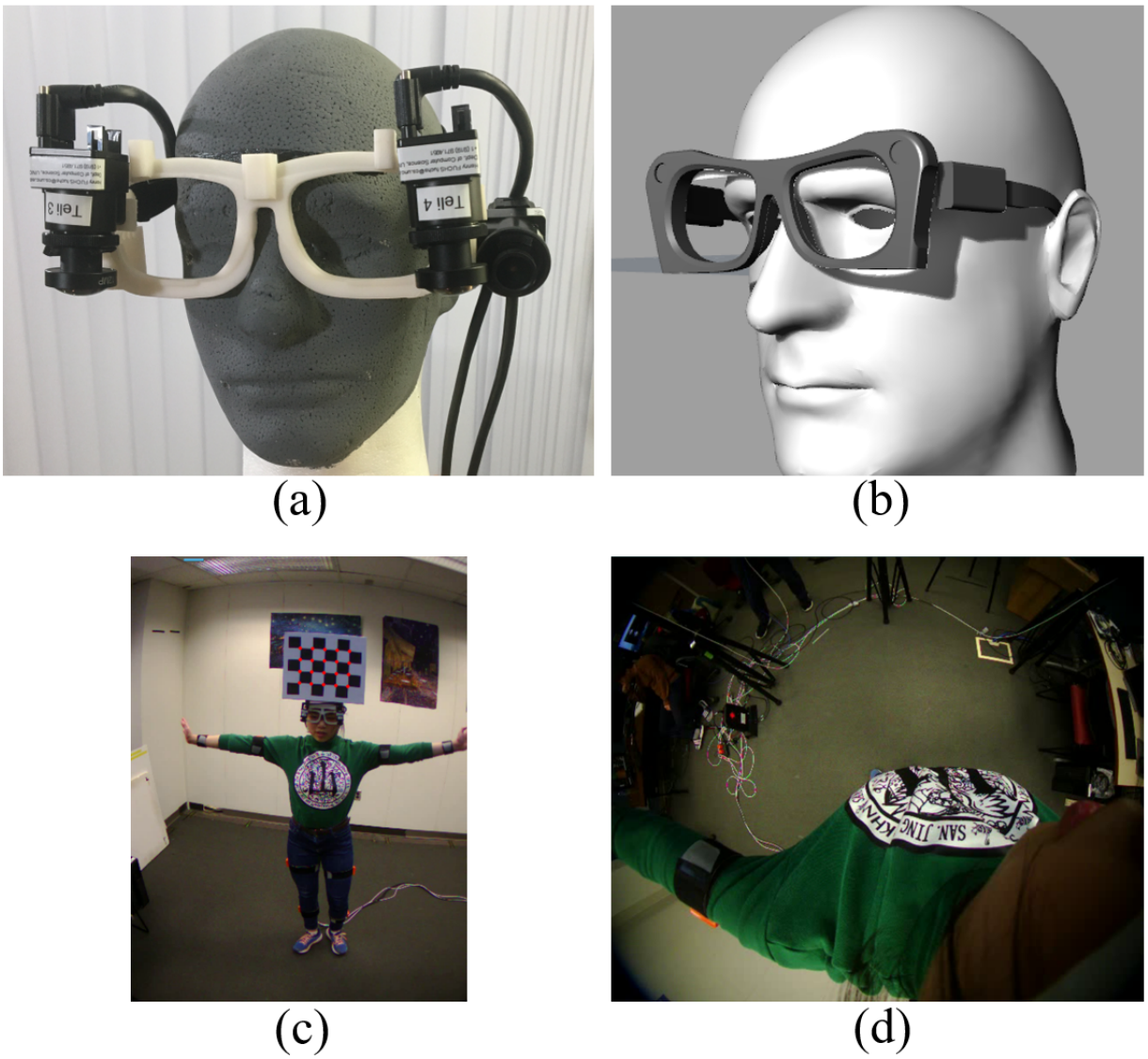


Figure 5.2: (a) Current headset capture prototype. (b) Future eyeglass-form factor design. (c) T-pose from external viewpoint. (d) T-pose in downward camera, with a worse viewpoint than in prior egocentric setups (Xu et al., 2019; Tome et al., 2019).

5.3 Wearable Capture and Egocentric Dataset

5.3.1 Eyeglasses and IMUs Prototype

The system introduced in this chapter aims to develop a fully mobile telepresence system whose sensors are embedded in commonly worn items such as eyeglasses, wristbands, and shoes. Toward that end, the prototype here uses cameras in eyeglasses frames and only 4 IMUs (Xsens MTw Awinda on wrists and ankles). Adding more IMUs (e.g., on the torso, elbows, and knees) improves the results, but the added inconvenience would considerably reduce acceptability. As shown in Section 5.5, the combination of multiple cameras, 4 IMUs, and deep learning-based techniques are sufficient to fill in the "missing" sensor data from elbows and knees.

I envision a headset design (shown in Figure 5.2d) with 4 miniature cameras: 2 downward-looking cameras placed at the bottom outside corners of the frame to observe the user's body, and 2 forward-looking cameras placed at the top outside corners of the frame to observe the environment. Compared to previous egocentric headsets (Rhodin et al., 2016; Cha et al., 2018; Xu et al., 2019; Tome et al., 2019), the design is more user-friendly but makes the 2 downward-looking viewpoints significantly more challenging as body parts are frequently out of view or occluded.

Working towards this design, a preliminary prototype was built with available larger cameras (Toshiba Teli BU505MCF) mounted on a 3D-printed eyeglasses frame, as shown in Figure 5.2a. Currently only 3 cameras are used; (two 160°FoV downward-looking cameras; one 121°FoV forward-looking camera).

5.3.2 Egocentric Visual+Inertial Human Pose Dataset

Following the work in egocentric video and IMU-based pose estimation in Subsection 5.2.3 and Subsection 5.2.4, I decided on a learning-based approach to use with the prototype. However, none of the available egocentric datasets were suitable for training because their viewpoints are farther away from the user's face, they contain no visibility information, and they are monocular. I could not use existing IMU datasets either, as they were lacking accompanying egocentric video

Figure 5.3: Incomplete body visibility in eyeglass-form factor views. Left column: Selected external views with reference data from Egocentric Visual+Inertial Human Pose Dataset (Ego-VIP dataset). Right column: Corresponding head-worn views with labeled visibility information.

Table 5.1: Egocentric Visual+Inertial Human Pose Dataset (Ego-VIP dataset), in number of frames.

	Real Data Size	Synthetic Data Size	Training Data Size	Test Data Size
Visual Dataset	11,822	38,588	50,410	13,213
Inertial Dataset	38,971	350,739	389,710	13,213

data. Consequently, I collected a new human pose dataset with users wearing the prototype headset in this chapter and 8 body-worn IMUs. The ground truth full-body 3D joints are acquired using multiple wall-mounted cameras in a capture studio (Cha et al., 2018). I recorded various types of motions for multiple users, including normal-speed as well as high-speed actions such as walking, sitting, gesturing, running, and physical therapy. A few examples are shown in Figure 5.3.

I collected 22 sequences for training and 9 sequences for evaluation with 6 human subjects, for a total of $38k$ frames of visual+inertial data. The summary of the dataset is shown in Table 5.1.

For the visual training data, $11k$ real images were uniformly sampled and manually filtered from the full recording. $38k$ synthetic images were generated using the body pose from the real data with the following random augmentations (Xu et al., 2019; Tome et al., 2019): clothing and background texture, head rotation, and headgear translation. Each joint visibility was estimated using the z-buffer of the projected body model onto the egocentric image and labeled as visible, occluded, or outside the FoV. Torso joints (neck, shoulders, and hips) were labeled as visible regardless of occlusion because they play an essential role as root joints for pose estimation.

The inertial data from the 8 sensors was synchronized with the visual data and calibrated using the method in Subsection 5.4.4. $38k$ frames of real IMU data were augmented by mirroring the pose front-to-back and side-to-side, temporally smoothing pose orientations, and introducing random acceleration noise.

To the best of my knowledge, this is the first dataset that includes stereo egocentric views with joint visibility and calibrated inertial data. The joint visibility information is crucial for training occlusion-aware joint detectors. The collected dataset is made publicly available at EgoVIP Dataset (2021) to contribute to the community of learning-based egocentric reconstruction.

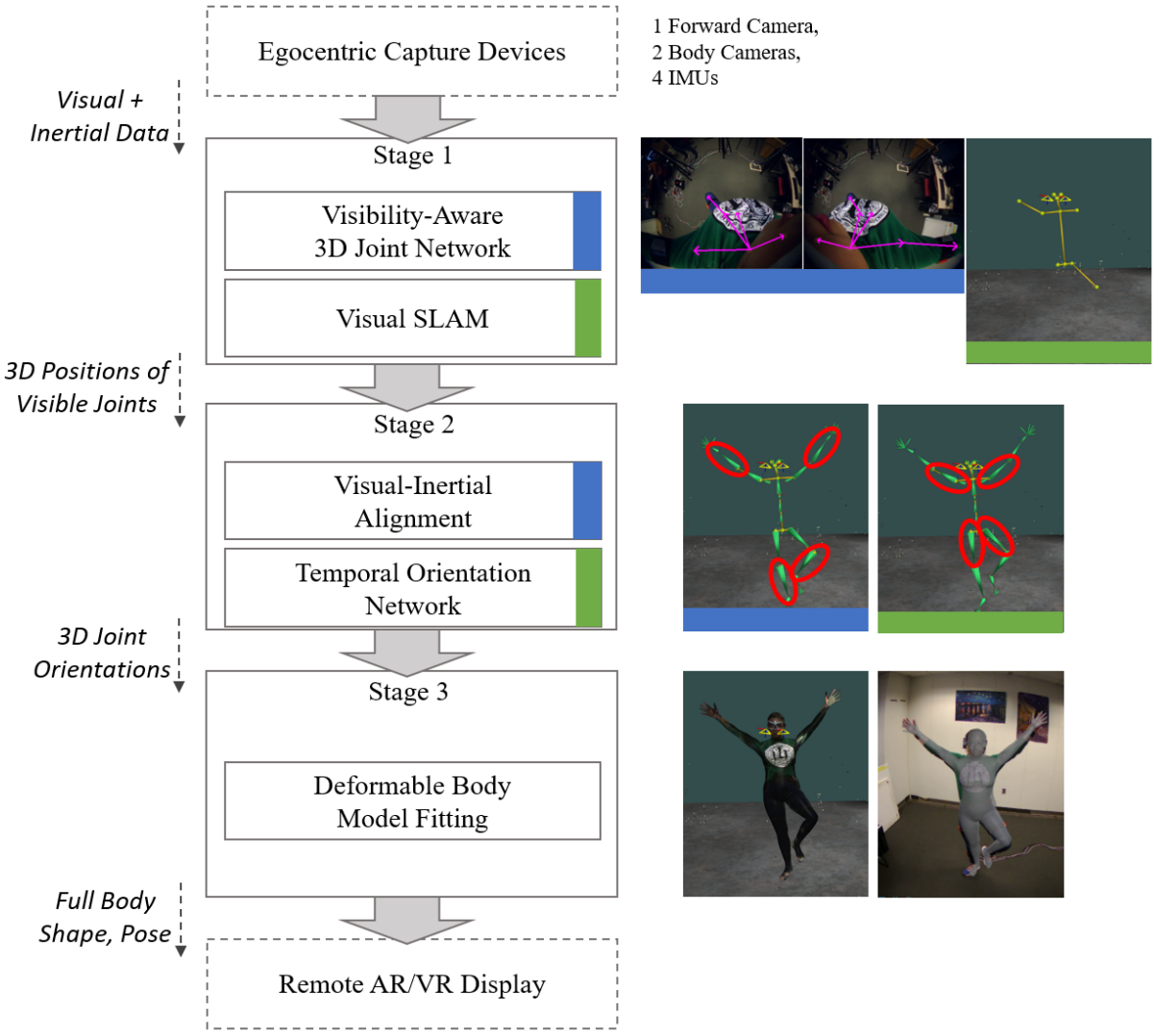


Figure 5.4: 3D Reconstruction Pipeline.

5.4 Egocentric Reconstruction Method

Working toward the goal of fully mobile telepresence, a real-time full-body shape and pose reconstruction method is devised using only egocentric devices I deem convenient and acceptable for daily wear: eyeglasses-mounted cameras and a few body-worn IMUs. The available information from the visual-inertial sensors is too sparse for each sensing modality to estimate the full-body pose by itself. First, limb motions are frequently occluded by the body or are invisible due to being outside the camera views. Second, IMUs are worn only on forearms and lower legs, so upper arm and thigh orientations are missing. To solve this ill-constrained problem, a *visibility-aware visual*

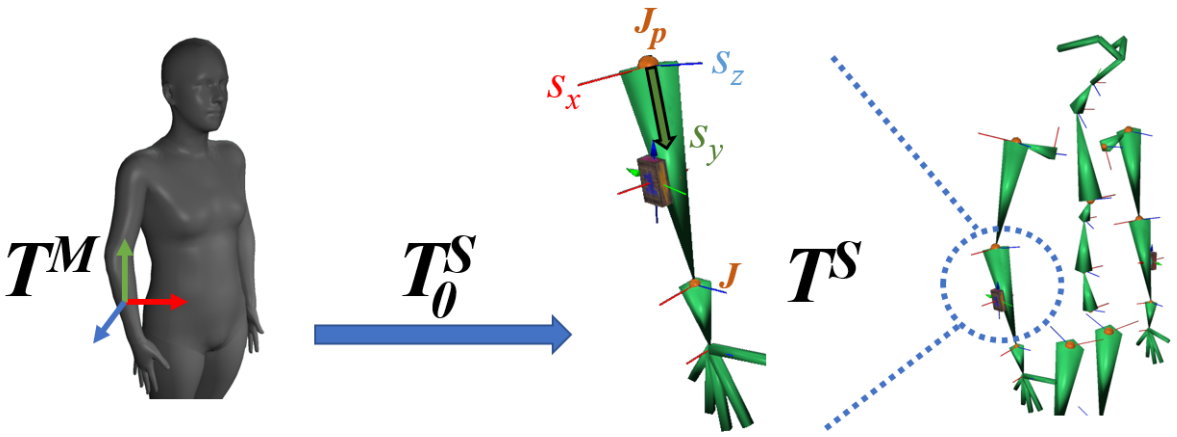


Figure 5.5: Bone representation. A bone (forearm) consists of a base joint (elbow) J_p, a tip joint (wrist) J, and an orientation $R^S = [s_x, s_y, s_z]$. They form a bone transformation T^S in the skeleton. The pose parameter T^M in the 3D mesh can be converted into skeleton space using the bind pose matrix T_0^S from the rest pose.

pose network and a *temporally-integrated visual and inertial pose network* are employed. The 3D reconstruction pipeline is illustrated in Figure 5.4. It consists of three main stages.

In the first stage, a *visibility-aware 3D joint detector network* (Subsection 5.4.2) estimates the 3D positions of joints observable in the two egocentric downward views. The detected 3D joints are transformed to world space (Subsection 5.4.3) using the headset pose estimated via $VSLAM$ (Sumikura et al., 2019).

In the second stage, the 3D orientations of lower bones (forearms, lower legs) and upper bones (upper arms, thighs) are estimated using a *visual-inertial IMU offset calibrator* (Subsection 5.4.4) and a *temporal visual-inertial orientation network* (Subsection 5.4.5), respectively.

In the third stage (Subsection 5.4.6), the shape and pose of the parametric body model are estimated using the estimated full-body 3D joint locations and orientations from the second stage.

5.4.1 3D Body Representation

In this approach, the *SMPL* parametric body model (Loper et al., 2015) is employed to represent the body shape and pose. It consists of 10 shape parameters β and $24 \cdot 3 = 72$ pose parameters θ, which deform a triangular mesh $\mathcal{M}(\theta, \beta)$ with $6,480$ vertices using linear blend skinning.

Instead of representing θ as a set of local bone rotations, the equivalent bone representation is used, which is defined as a set of global transforms $T^M \in \mathbb{R}^{4\times4}$. ^{M} is used to denote the body Mesh space and ^{S} is used to denote the Skeleton space. In this representation, a bone i is defined by two connected joints and a transform (Figure 5.5).

The skeletal bone transformation $T_i^S \in \mathbb{R}^{4\times4}$ in global space is defined as a convenient way to represent the pose in the skeleton as:

$$T_i^S = \begin{bmatrix} R_i^S & J_{p(i)} \\ \mathbf{0} & 1 \end{bmatrix} \tag{5.1}$$

$R^S = [s_x, s_y, s_z] \in \mathbb{R}^{3\times3}$ is the bone rotation and J_p is the base joint position. The column vectors of R^S form the 3D axes of the bone and the axis $s_y = R^{[:,2]}$ represents the bone direction d_i from the base (parent) to tip (child) joint:

$$d_i = \frac{J_i - J_{p(i)}}{||J_i - J_{p(i)}||_2} \tag{5.2}$$

The bone direction computed from a rotation is also denoted as:

$$d_i = d(R_i) = R_i^{[:,2]} \tag{5.3}$$

The pose parameter T_i^M can be directly computed from T_i^S as:

$$T_i^M = T_i^S (T_{i,0}^S)^{-1} \tag{5.4}$$

The bind pose matrix $T_{i,0}^S$ maps the coordinate frames $\mathcal{F}^M \mapsto \mathcal{F}^S$, is calculated using the joint positions in the rest pose of the body model, and updated only when the shape parameters β are changed. In the rest pose, T_i^M is the identity matrix.

The joint positions in rest pose J_0 are described by the joint regressor $\mathcal{J}$ from the shaped vertices. The body shape β using the unposed joints $J_0 = (T^M)^{-1}(J)$ is estimated by minimizing E_{shape}:

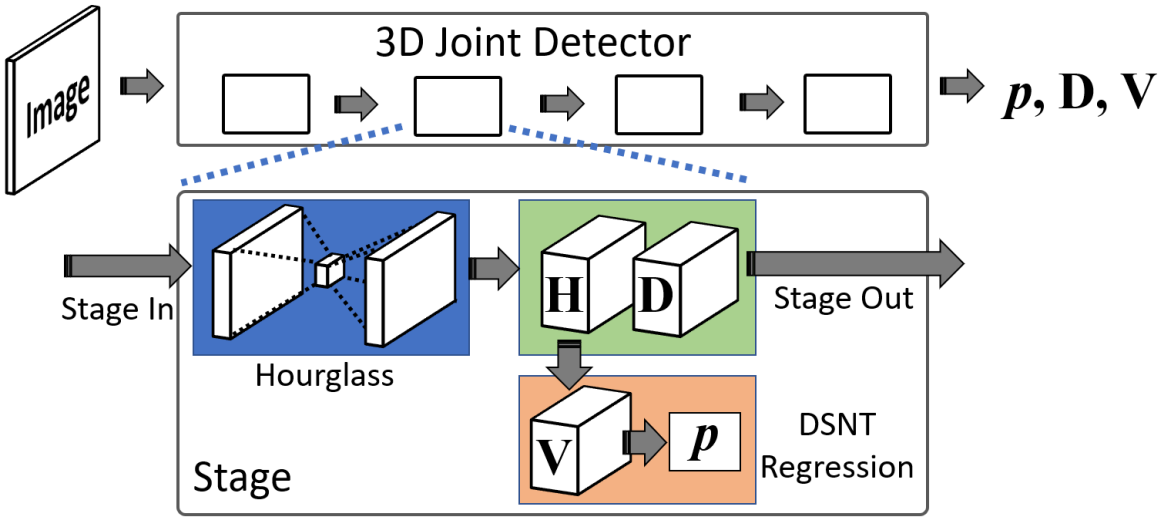


Figure 5.6: Network Structure for the 3D Joint Detector. The Hourglass module outputs joint heatmaps H and depthmaps D as concatenated channels. H and D are propagated into the next stage. The regression module outputs 2D coordinates p from confidence maps V normalized by H. Given a single input image, the 4 stage-network outputs p, D, V, from which 3D joint coordinates are computed.

$$E_{shape} = \sum_{i=1}^{K} ||(T_i^M)^{-1}(J_i) - \mathcal{J}_i(\mathcal{M}_0 + \mathcal{B}_s(\beta))||_2^2 + w_s||\beta||_2^2 \tag{5.5}$$

$w_s = 0.001$ is a weight for the regularization term, and $K = 13$ is the number of joints. The vertices are reshaped by the mean shape $\mathcal{M}_0$ and the linear blend shapes $\mathcal{B}_s(\beta)$.

5.4.2 Visibility-Aware 3D Joint Detection Network

In visual human pose estimation, occluded joints often lead to erroneous results (Cheng et al., 2019). When using egocentric images, legs are frequently occluded by the body, and arms can be out of camera FoV (Xu et al., 2019; Cha et al., 2018). In this subsection, the visibility-aware 3D joint detection network takes a $m \times m$ egocentric image as input ($m = 320$) and estimates only the observable joints while rejecting unreliable joints by incorporating joint visibility information. The egocentric dataset described in Subsection 5.3.2 is labeled with visibility information, enabling visibility awareness training. The $g$round $t$ruth (^{gt}) binary $v$isibility v^{gt} is set to 1 for visible joints and 0 for invisible (occluded or outside of FoV) joints.

The *Stacked Hourglass* architecture (Newell et al., 2016) used in 2D human pose estimation is extended to a 3D joint estimation network (Figure 5.6). In a head-worn wide-FoV camera image, lower body joints appear significantly smaller than upper body joints. Instead of using multi-scale images (Xu et al., 2019), the advantage is taken in that the Hourglass module inherently collects information across all image scales. A $DSNT$ regression module (Nibali et al., 2018) is also used to estimate 2D coordinates from heatmaps. This regression module increases computational efficiency, as heatmaps no longer need to be transferred to the CPU for parsing at runtime.

The Hourglass module infers heatmaps $H \in \mathbb{R}^{(m/4)\times(m/4)\times K}$ in the first K channels and inverse depthmaps $D \in \mathbb{R}^{(m/4)\times(m/4)\times K}$ in the last K channels. H are normalized into confidence maps V by a Softmax layer. V are transformed into 2D coordinates $\mathbf{p}$ by the dot product of the X- and Y-coordinate matrices (Nibali et al., 2018).

The inverse depthmap D is a heatmap containing normalized inverse depth values for joints. The normalized inverse depth value is defined as,

$$(d_{max} - d)/d_{max} \tag{5.6}$$

where d is a depth in meters and $d_{max} = 2$ is the maximum depth. Distances close to the camera are assigned higher values, and farther distances are assigned near-zero values (Wang et al., 2018).

Confidence $\tilde{v}$ and depth d are read out at the estimated $\mathbf{p} = (x, y)$ coordinate in V and D, respectively. When confidence $\tilde{v}$ is large enough ($\tilde{v} > t_v$, with $t_v = 0.05$), coordinate $\mathbf{p}$ is considered valid and visibility v is set to 1, otherwise it is set to 0. The raw inverse depth read-out is transformed back into depth d in meters using Equation 5.6. The 3D joint position is computed by back-projecting (x, y, d) using the camera calibration matrix. The output of the stage, the concatenated H and D, are propagated into the next stage as input. 4 stacked stages are used taking into account both accuracy and speed.

The network is trained to minimize the loss function:

$$\mathcal{L}_{joint\_net} = \mathcal{L}_{DSNT} + \mathcal{L}_V + \mathcal{L}_D \tag{5.7}$$

Given binary visibility v^{gt} for each joint, regression loss $\mathcal{L}_{DSNT}$ and depth loss $\mathcal{L}_D$ are applied for $v^{gt} = 1$, and invisibility loss $\mathcal{L}_V$ is applied for $v^{gt} = 0$.

The *regression loss* $\mathcal{L}_{DSNT}$ is applied for the confidence maps V and coordinates $\mathbf{p}$ with the ground truth positions $\mathbf{p}^{gt}$ and binary visibility v^{gt} as:

$$\mathcal{L}_{DSNT} = \sum_{i=1}^{K} v_i^{gt} \cdot [||\mathbf{p}_i^{gt} - \mathbf{p}_i||_2^2 + \mathcal{D}(V_i||\mathcal{N}(\mathbf{p}_i^{gt}, \sigma I_2))] \tag{5.8}$$

$\mathcal{N}(\mu, \sigma)$ is a 2D Gaussian map drawn at μ with standard deviation σ ($\sigma = 1$ for training). $\mathcal{D}(\cdot||\cdot)$ is the Jensen-Shannon divergence to encourage H to resemble the 2D Gaussian map (Nibali et al., 2018).

The *invisibility loss* $\mathcal{L}_V$ suppresses H to a zero heatmap for invisible joints:

$$\mathcal{L}_V = \sum_{i=1}^{K} (1 - v_i^{gt}) \cdot ||H_i||_2^2 \tag{5.9}$$

The invisibility loss forces the uniform distribution in V, which encourages the confidence value to be smaller for invisible joints.

The *depth loss* $\mathcal{L}_D$ is applied for depthmaps D with ground truth depthmaps D^{gt} and joint masks $\mathcal{M}(\mathbf{p}^{gt})$ as:

$$\mathcal{L}_D = \sum_{i=1}^{K} v_i^{gt} \cdot ||\mathcal{M}(\mathbf{p}_i^{gt}, \sigma I_2) \oslash (D_i - D_i^{gt})||_2^2 \tag{5.10}$$

$\mathcal{M}(\mu, \sigma)$ is a 2D binary maskmap drawn at μ with radius σ (set to 1.8 during training), and $\oslash$ is the Hadamard product. Note that the depthmap is trained only for the interest joint area so that the outside area is left unchanged to prevent over-fitting, which results in zero depthmap output when not using the maskmap (Mehta et al., 2017b).

The network is trained in multiple stages. First, the 2D layers are trained on the MPII Human Pose dataset (Andriluka et al., 2014) to learn low-level texture features. Only the regression loss $\mathcal{L}_{DSNT}$ is used in the training, while visibility is ignored. Then, the network is trained on the dataset

in Subsection 5.3.2 with the full loss function $\mathcal{L}_{joint\_net}$. Intermediate supervision is applied during training.

For the right-sided image, the advantage of the symmetry between the two downward-looking camera views is taken by flipping the image to use the same network as the left image. The output joint coordinates from the right image are then flipped back. This strategy allows a single network to be used at training and runtime for both views.

5.4.3 Temporally, Multi-view Consistent Joint Estimation

3D joints are detected in the left and right downward camera views independently and are reprojected into a single 3D space using the camera calibration matrices. Joints that are not consistent with their counterparts due to erroneous detection are filtered out such that the results are both multi-view-consistent and temporally coherent.

First, the raw detection of a joint is filtered out if its bone direction d_i is temporally inconsistent, which is defined as a change of more than $30°$ between frames. Next, the filtered measurements are used to estimate the multi-view-consistent and temporally-coherent joint position $X \in \mathbb{R}^3$, by minimizing the weighted sum:

$$E_{consist\_joint} = E_{proj} + w_d E_{dep} + w_l E_{len} + w_t E_{temp} \tag{5.11}$$

where w_d, w_l, and w_t are non-negative weights. For torso joints including neck, hips, and shoulders, $w_d = 1, w_l = 0, w_t = 10$. $w_d = 2, w_l = 2, w_t = 1$ for arm joints, and $w_d = 1, w_l = 5, w_t = 2$ for leg joints.

The *projection cost* E_{proj} is defined as:

$$E_{proj} = \sum_{c=1}^{C} ||\mathbf{p}_c - P_c \cdot X||_2^2 \tag{5.12}$$

where C is the number of views, $\mathbf{p}_c$ is the 2D location measurement in camera image c, and P_c is camera c's projection matrix.

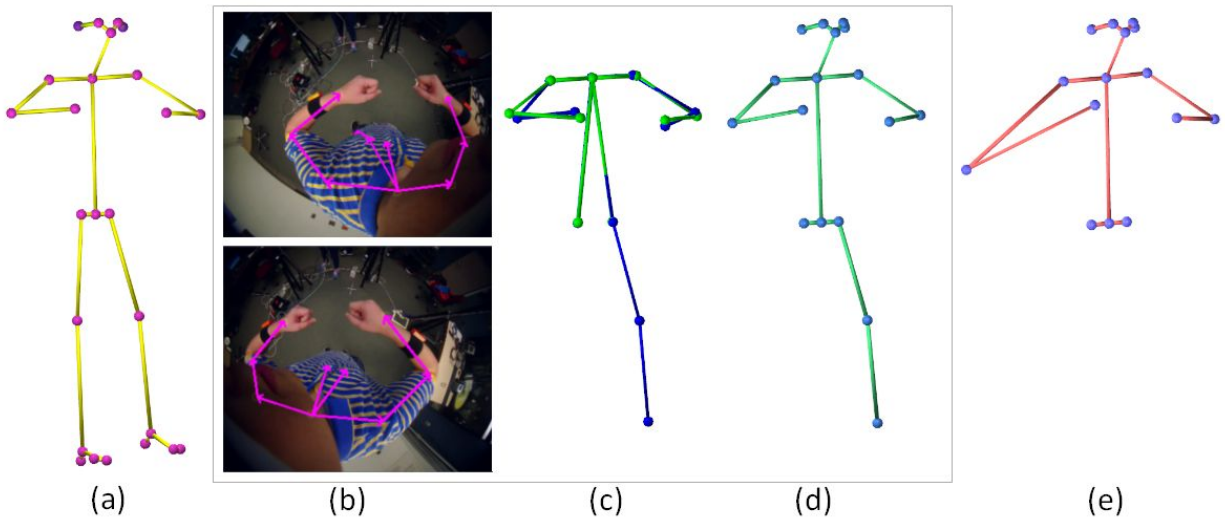


Figure 5.7: Consistent 3D joints. (a) Reference 3D joints. (b) Joint detections from left camera (top), and right camera (bottom). (c) 3D joints from left camera (blue), and right camera (green). (d) Joints reconstructed by the method in Subsection 5.4.3. (e) Joints reconstructed using direct triangulation for comparison.

The *depth cost* E_{dep} is defined as:

$$E_{dep} = \sum_{c=1}^{C} ||d_c - T_c^{[3,:]} \cdot X||_2^2 \tag{5.13}$$

where d_c is the depth measurement in camera c, and $T_c^{[3,:]}$ is the third row of the extrinsic matrix of camera c.

Bone lengths are maintained over time, starting with the initialization and averaging with new detection measurements. The initial bone lengths are taken from the body model in its rest pose and scaled by the ratio between the model and detected spine lengths. The *bone length consistency* E_{len} is measured as:

$$E_{len} = ||X_l - ||X_p - X||_2||_2^2 \tag{5.14}$$

where X_p is the parent joint's position and X_l is the bone length of joint X.

The *temporal smoothness cost* E_{temp} is defined as:

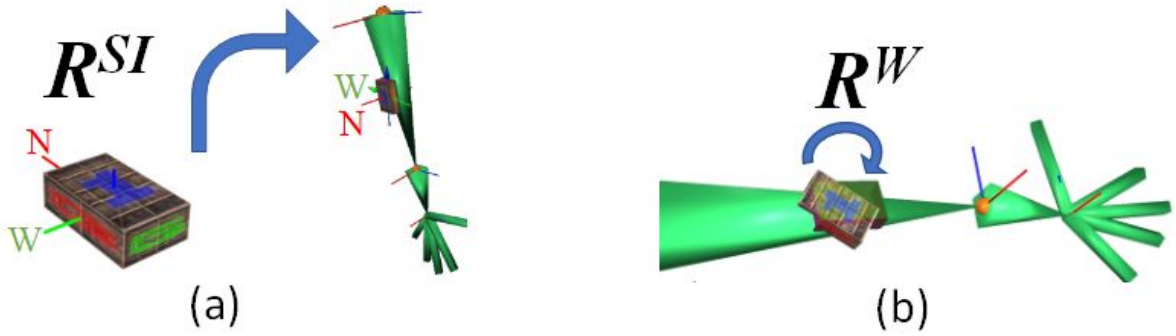


Figure 5.8: Coordinate frame transformations. (a) Rotation of inertial sensor to skeleton space R^{SI}, indicating the predefined wear pose. (b) IMU rotation offset R^{W}, used to compensate for misaligned IMUs.

$$E_{temp} = ||X_{t-1} - X||_2^2 \tag{5.15}$$

where X_{t-1} is the joint position in the previous frame.

The estimated 3D joint positions X in headset space are transformed into joint positions J in 3D world space using the current estimated headset pose acquired via $VSLAM$ (Sumikura et al., 2019) running in a separate thread at 35 fps.

The entire process, shown in Figure 5.7, results in better reconstruction than when using direct triangulation, even when joints are detected in both views.

5.4.4 Visual-Inertial Alignment

A human pose can be estimated with body-worn inertial sensors by using the sensor measurements to track the orientations of the corresponding bones. IMUs are typically calibrated using a specific initial pose (Von Marcard et al., 2016; von Marcard et al., 2017, 2018; Huang et al., 2018). Prior methods assume that the sensors are placed accurately at designated poses (positions and orientations), and that the user assumes the correct body pose in the beginning. However, even slightly misaligned body-worn IMUs can interfere with visual-inertial consistent pose estimation, yielding inaccurate results.

These inaccuracies can be corrected by estimating an *IMU rotation offset* $R^W \in \mathbb{R}^{3\times3}$ using collected samples of visual and inertial pairs of IMU-instrumented bone directions over time. It represents how much a sensor is offset from the assumed initial orientation of the bone (Figure 5.8b).

The bone rotation R_t^S at time step t from Equation 5.1 can be computed for the lower bones from the IMUs mounted on them as:

$$R_t^S = R^W \cdot R_t^I \cdot (R^{SI})^{-1} \cdot R_0^S \tag{5.16}$$

R_t^I is the orientation read from the Inertial sensor at time t, R_0^S is the rotation in rest pose from T_0^S in Equation 5.4, and $(R^{SI})^{-1}$ maps the coordinate frame $\mathcal{F}^S \mapsto \mathcal{F}^I$ (Figure 5.8a).

The Inertial lower bone direction d_t^I is defined as:

$$d_t^I = R_t^I \cdot (R^{SI})^{-1} \cdot d(R_0^S) \tag{5.17}$$

$d(R_0^S)$ indicates the bone direction in the rest pose from Equation 5.3.

The IMU rotation offset R^W is updated whenever measurements from the visual detector of the same bone are available, so that all prior bone directions $d(R_1^S), ..., d(R_t^S)$ agrees with the corresponding visual bone directions $d_1^V, ..., d_t^V$ from Equation 5.3. Note that $R^W = I_3$ when the sensor is worn in exactly the designated position and orientation. R^W can be estimated from a sequence of Visual d^V and Inertial d^I directions by solving the least square problem:

$$\min_{R^W} \sum_t ||d_t^V - R^W \cdot d_t^I||_2^2 \tag{5.18}$$

Solving Equation 5.18 for all available (d^I, d^V) pairs is computationally intensive. Instead, the visual-inertial pairs are grouped and R^W is updated using the online k-means algorithm described in Table 5.2 with an online k-d tree structure.

At runtime, a fixed $k = 200$ number of cluster pairs is maintained in the k-d tree. The sampling strategy maximizes between-cluster distances, which favors uniform distribution of the clusters and minimizes the number of colinear samples.

Table 5.2: Online IMU Rotation Offset Calibration Algorithm.

Input: Inertial direction d^I, Visual direction d^V

Data: k clusters $\mathbf{c} = (d_c^I, d_c^V)$ in k-d Tree T,
cluster $c_{min} \in \mathbf{c}$ with minimum nearest neighbor distance ($nndist$)

Output: IMU rotation offset R^W

$x \leftarrow$ next-sample (d^I, d^V);
$c \leftarrow$ nearest(x) in T;

if $dist(x, c) < nndist(c_{min})$ **then**
    $c' \leftarrow$ average(x,c);
    replace c with c' in T; // (cluster updated)
else
    remove c_{min} from T;
    push x to T; // (new cluster created)
    find new c_{min} in T;
end

Update R^W from **c** pairs using Equation 5.18;

The lower bone orientations R^S can always be estimated from R^W, regardless of their visibility, using Equation 5.16.

5.4.5 Temporal Visual-Inertial Orientation Network

Upper arm and thigh orientations can be estimated at every step using a sequence of forearm and lower leg motions, respectively, under the assumption that the movements of the lower and upper bones of the same limb are highly correlated (von Marcard et al., 2017; Huang et al., 2018). However, multiple upper arm or thigh orientations are possible for a single forearm or lower leg pose (pose ambiguity problem). To overcome this difficulty, the approach in this subsection uses visual observations of the upper bones when available as well as inertial measurements of the lower bones. The subscripts $\mathbf{i}$ and $\mathbf{u}$ are used to distinguish between the sensor-**i**nstrumented lower bones and the **u**ninstrumented upper bones.

The calibrated forearm and lower leg orientations $R^S_{\mathbf{i}}$ are computed using the IMU offset matrix $R^W_{\mathbf{i}}$ in Equation 5.16. Similarly, the raw accelerations $a^I_{\mathbf{i}}$ can be used to compute $a^S_{\mathbf{i}} = R^H_{\mathbf{i}} \cdot a^I_{\mathbf{i}}$ using the IMU acceleration offset matrix R^H, indicating the Heading reset, a rotation along the up direction computed from R^W.

The un-instrumented upper arm and thigh orientations $R^S_{\mathbf{u}}$ are estimated from a sequence of previous $R^S_{\mathbf{i}}$, $a^S_{\mathbf{i}}$ for the forearms and lower legs, as well as the availability of visual upper arm and thigh directions $d^V_{\mathbf{u}}$ from the visual detector in Subsection 5.4.3, while enforcing the constraint $d(R^S_{\mathbf{u}}) = d^V_{\mathbf{u}}$ from Equation 5.3. To be invariant to the body direction, $R^S_{\mathbf{i}}$, $a^S_{\mathbf{i}}$, and $d^V_{\mathbf{u}}$ are normalized with respect to the root joint (hip center) orientation R^S_{root} at time step t (Huang et al., 2018):

$$R^N(t) = (R^S_{root}(t))^{-1} \cdot R^S_{\mathbf{i}}(t) \tag{5.19}$$

$a^S_{\mathbf{i}} \rightarrow a^N$, and $d^V_{\mathbf{u}} \rightarrow d^N$ are similarly normalized. $^{\mathbf{N}}$ is used to indicate the **N**ormalized torso space.

The input feature vector at time t is defined as:

$$x_t = [r_t, \omega_t, a_t, v_t \cdot d_t]^T \tag{5.20}$$

r_t denotes $[r_1^N(t), ..., r_4^N(t)]^T$ for 4 input bones. ω_t, a_t, and $v_t \cdot d_t$ are similarly defined. r_i^N is the vectorized R_i^N, and $\omega_i^N(t)$ is the angular velocity between $R_i^N(t)$ and $R_i^N(t-1)$. The input feature vector incorporates the lower bone motions represented by rotation, velocity, and acceleration. If the joints of the upper bone i are provided by the visual detector, its direction d_i^N is added and its visibility v_i is set to 1. Otherwise, experiments showed that using $v_i = 0.1^{-3}$ and $d_i^N = (1, 1, 1)$ yields better performance than setting both to 0. The dimension of x_t is $(9 + 3 + 3 + 3) \cdot 4 = 72$ for the 4 IMU-instrumented bones (r_t, ω_t, a_t) and for the 4 uninstrumented bones ($v_t \cdot d_t$).

The output feature vector at time t is defined as:

$$y_t = [r_1^o(t), ..., r_4^o(t)]^T \tag{5.21}$$

y_t contains the vectorized uninstrumented bone orientations. r_i^o are reshaped to the output orientations $R_i^o(t)$. The dimension of y_t is $(9) \cdot 4 = 36$ for the 4 upper arm and thigh bones.

The task of the orientation network is to learn a function $f : \mathbf{x} \rightarrow y_t$ that predicts the uninstrumented bone orientations from a sequence of input features $\mathbf{x} = [x_{t-n+1}, ..., x_t]$. A Transformer network is employed, which has been shown to outperform LSTM in many applications (Vaswani et al., 2017). The input sequence is composed of measurements from the last $n = 20$ frames (Huang et al., 2018). The network architecture is shown in Figure 5.9.

The network is trained with the following loss function:

$$\mathcal{L}_{bone\_net} = ||y - y^{gt}||_2^2 + \sum_{i=1}^{4} v_i^{gt} \cdot acos(d(R_i^o), d_i^{gt}) \tag{5.22}$$

The orientation loss is measured using the ground truth y^{gt}. $d(R^o)$ represents the output bone direction computed using Equation 5.3. It is penalized by the ground truth d^{gt} bone direction, which encourages the output bone direction to be consistent with the visual input bone direction if provided. The second term is only computed if $v^{gt} = 1$.

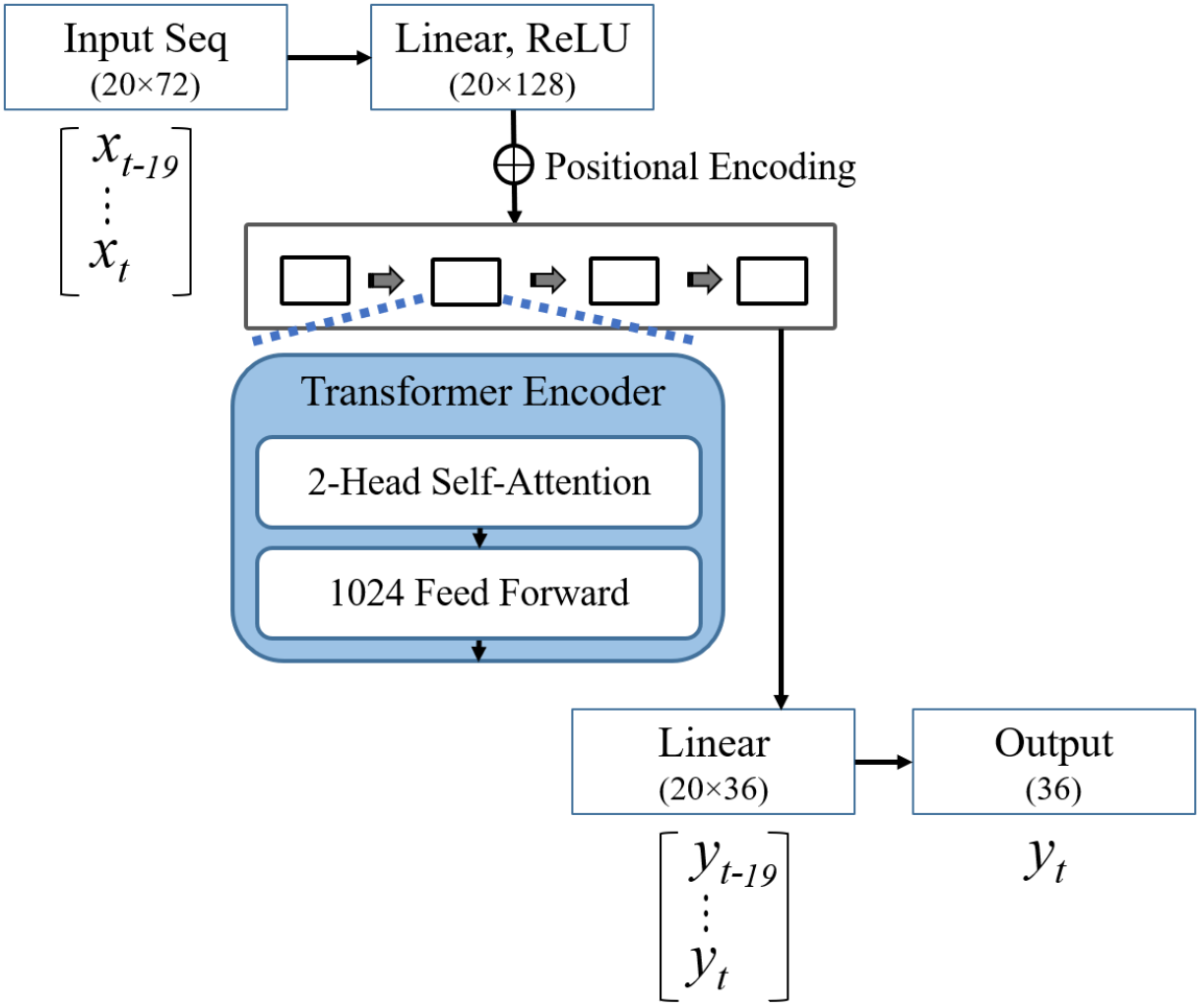


Figure 5.9: Temporal Visual-Inertial Orientation Network architecture. Using a sequence of visual-inertial input feature vectors $\mathbf{x}$, the uninstrumented orientations y are estimated. All layers use dropout 0.2 in training. The numbers in brackets indicate the output dimensions of each layer.

At run-time, the estimated R^o in normalized torso space are transformed to $R^S_{\mathbf{u}}$ in world space using Equation 5.19.

5.4.6 Deformable Body Model Fitting

The pipeline estimates the full-body shape and pose from the estimated joint positions J and bone rotations R^S in the previous subsections. Unobserved joint positions are recovered using forward kinematics from R^S and the corresponding bone lengths. The body shape is updated by solving Equation 5.5 using the full-body joint positions J.

The bone rotations R^S are further corrected by using the detected visual direction outputs d^V when available. The estimated R^S are temporally coherent, but the motion may be over-smoothed when sudden changes in motion or visibility occur along the edges of the camera images. This issue can be avoided by fitting bone orientations R^S closer to visual directions d^V, which encourages a quicker reaction to changes. The corrected bone rotations $\bar{R}^S$ can be estimated if d^V are available:

$$\bar{R}^S = R^{v2v}(d(R^S), \alpha \cdot d^V + (1-\alpha) \cdot d(R^S)) \cdot R^S \tag{5.23}$$

$R^{v2v}(v_1, v_2)$ is the rotation from v_1 to v_2 vectors, and $\alpha = 0.8$ at run-time. The joint positions $\bar{J}$ are also updated by the forward kinematics using $\bar{R}^S$. The pose parameters T^M are estimated by using $\bar{R}^S$ and $\bar{J}$ in Equation 5.1 and Equation 5.4. The estimated joints $\bar{J}$ are transferred to the next frame for the temporally consistent joint estimation in Subsection 5.4.3.

5.5 Results and Evaluation

The 3D pose estimation method described in this chapter is not directly comparable to any prior methods I am aware of. Outside-looking-in camera-based methods (Mehta et al., 2020; Xiang et al., 2019; Kocabas et al., 2020; Habermann et al., 2020) require all joints to be visible. Prior visual+inertial fusion approaches (Malleson et al., 2017; Trumble et al., 2017; von Marcard et al., 2018) additionally require more than 10 densely-worn IMUs. The method in this chapter uses as

Table 5.3: Performance of monocular HG3D on the Mo2Cap2 dataset (Xu et al., 2019) showing mean joint position errors (cm).

	Indoor (cm)	Outdoor (cm)
3DV'17 (Mehta et al., 2017a)	7.628	9.446
VNect (Mehta et al., 2017b)	9.785	11.375
Mo2Cap2 (Xu et al., 2019)	6.140	8.064
xR-EgoPose (Tome et al., 2019)	4.816	6.019
Monocular HG3D (Hourglass 3D)	8.680	8.823

input stereo head-worn views that almost never capture the entire body, and only 4 inertial sensors worn on wrists and ankles. The performance of the method, *Egocentric Visual+Inertial Poser* (**EgoVIP**), is compared with the following three baseline approaches:

HG3D (*stereo stacked hourglass 3D*) is a visual-only method that uses the 3D joint detector in Subsection 5.4.2 without the visibility awareness term in Equation 5.9 (Newell et al., 2016; Nibali et al., 2018; Mehta et al., 2017b). It detects both visible and invisible joints, and merges the joints from the two downward camera views as shown in Subsection 5.4.3 to produce full-body 3D joint positions. The 3D bone rotations are estimated from the detected joints using the inverse kinematics (IK) algorithm in Cha et al. (2018). A monocular stacked hourglass 3D is also separately evaluated on the publicly-available egocentric dataset in Xu et al. (2019) and shows competitive results in Table 5.3.

DIP is my implementation of *Deep Inertial Poser* in Huang et al. (2018), an IMU-based method which uses 6 sensors placed on wrists, ankles, torso, and head. The ground truth values are used for head and torso orientations and accelerations. DIP is unable to estimate global position of the body; thus including only limb motions in the comparison. Due to the incapability of DIP, ground truth body shapes and pre-calibrated inertial measurements are also used. 20 past frames and 5 future frames are included for DIP, along with the best configuration of the LSTM architecture. In contrast, the method introduced in this chapter estimates the body shapes and sensor calibrations at run-time and does not use future frames.

Table 5.4: Quantitative evaluation on the Ego-VIP dataset as average joint position errors (cm). The joint poses were evaluated for visible, occluded, and outside-camera-FoV cases. Methods: HG3D = Stereo Hourglass 3D (2 head-worn views); DIP = Huang et al. (2018) (6 IMUs); EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs); EgoVIP8 = The extended method in Chapter 5 (2 head-worn views, 8 IMUs). The worst results are shown in bold.

	μ_{cm}^{total}	σ_{cm}^{total}	$\mu_{cm}^{visible}$	$\sigma_{cm}^{visible}$	$\mu_{cm}^{occlusion}$	$\sigma_{cm}^{occlusion}$	$\mu_{cm}^{outside}$	$\sigma_{cm}^{outside}$
HG3D	3.69	4.44	2.67	2.81	6.18	5.58	**18.34**	**11.51**
DIP	**6.06**	**5.32**	**4.33**	**4.31**	**10.52**	**6.91**	13.66	4.95
EgoVIP (Ch. 5)	3.33	2.49	2.46	1.78	5.60	3.47	5.50	2.96
EgoVIP8 (Ch. 5)	3.17	1.68	2.44	1.31	5.08	2.16	4.50	1.63

Table 5.5: Quantitative evaluation on the Ego-VIP dataset as orientation errors (degrees). The joint poses were evaluated for visible, occluded, and outside-camera-FoV cases. Methods: HG3D = Stereo Hourglass 3D (2 head-worn views); DIP = Huang et al. (2018) (6 IMUs); EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs); EgoVIP8 = The extended method in Chapter 5 (2 head-worn views, 8 IMUs). The worst results are shown in bold.

	μ_{degree}^{total}	σ_{degree}^{total}	$\mu_{degree}^{visible}$	$\sigma_{degree}^{visible}$	$\mu_{degree}^{occlusion}$	$\sigma_{degree}^{occlusion}$	$\mu_{degree}^{outside}$	$\sigma_{degree}^{outside}$
HG3D	**19.65**	**16.36**	**21.86**	**16.47**	**16.04**	**12.38**	**83.94**	**19.54**
DIP	18.14	11.70	20.05	12.57	15.60	9.93	30.93	11.79
EgoVIP (Ch. 5)	11.28	6.87	10.88	7.00	11.71	6.28	15.42	7.01
EgoVIP8 (Ch. 5)	8.76	4.72	7.74	4.33	9.99	4.99	11.78	4.29

EgoVIP8 is an extended version of the method that uses 8 IMUs worn on wrists, ankles, upper arms, and thighs. Since actual measurements are available, the temporal orientation network for upper arm and thigh bone estimation is skipped in Subsection 5.4.5. Instead, the visual-inertial alignment in Subsection 5.4.4 is applied to all 8 IMUs over time.

To assess the accuracy of the reconstruction results, the system in this chapter is evaluated by comparing 3D joint position and orientation errors between the estimates and the ground truth. The results for the Ego-VIP dataset are shown in Table 5.4 and Table 5.5, broken down into three categories of joints: visible, occluded, and outside FoV. In all categories, the method (EgoVIP) in this chapter significantly outperforms HG3D and DIP.

HG3D's accuracy is comparable with the method in this chapter for visible joints, but its position errors are significantly higher for both occluded and outside-FoV joints. The orientations computed

Table 5.6: Per-joint average position errors (cm) for the method introduced in this chapter on the Ego-VIP dataset. The joint poses were evaluated in visible, occluded, and outside-camera-FoV cases. The worst results are shown in bold.

	μ_{cm}^{total}	σ_{cm}^{total}	$\mu_{cm}^{visible}$	$\sigma_{cm}^{visible}$	$\mu_{cm}^{occlusion}$	$\sigma_{cm}^{occlusion}$	$\mu_{cm}^{outside}$	$\sigma_{cm}^{outside}$
Neck	1.29	0.69	1.29	0.69	N/A	N/A	N/A	N/A
Shoulder	1.53	0.84	1.53	0.84	N/A	N/A	N/A	N/A
Hip	2.40	1.37	2.40	1.37	N/A	N/A	N/A	N/A
Elbow	2.34	1.76	2.15	1.28	3.55	2.60	**7.08**	**3.63**
Wrist	3.02	2.37	2.74	1.53	4.49	**4.30**	4.95	2.68
Knee	5.40	**3.84**	5.56	**4.42**	5.32	3.25	N/A	N/A
Ankle	**6.32**	3.73	**6.53**	3.78	**6.28**	3.60	N/A	N/A

Table 5.7: Per-bone average orientation errors (degrees) for the method introduced in this chapter on the Ego-VIP dataset, using only forearm- and lower-leg IMUs; upper bones (upper arm, thigh) are estimated. The worst results are shown in bold.

	μ_{degree}^{total}	σ_{degree}^{total}	$\mu_{degree}^{visible}$	$\sigma_{degree}^{visible}$	$\mu_{degree}^{occlusion}$	$\sigma_{degree}^{occlusion}$	$\mu_{degree}^{outside}$	$\sigma_{degree}^{outside}$
Upper Arm	**12.7**	**8.5**	12.5	**8.2**	**13.1**	**8.2**	**25.9**	**9.7**
Thigh	12.4	7.1	**15.0**	7.9	11.1	6.2	N/A	N/A
Forearm	7.4	5.0	7.2	4.7	7.8	5.6	11.7	5.7
Lower Leg	12.5	6.3	12.2	7.3	12.5	6.0	N/A	N/A

using IK are significantly less accurate than when acquired from inertial sensors. This comparison shows that even a few inertial sensors significantly improve pose accuracy in joint positions and orientations.

DIP shows significantly lower accuracy and higher variance than the method in this chapter in both position and orientation. This comparison shows that incorporating even sparse visual information into an IMU-based method significantly stabilizes the temporal accuracy. For invisible joints, the accuracy of the method (EgoVIP) in this chapter drops significantly due to relying entirely on inertial sensors, while still outperforming DIP. The Transformer network-based orientation estimation shows less variance than DIP's LSTM-based network.

Table 5.6 shows the position accuracy for each joint. Leg joints show significantly lower position accuracy due to decreased visibility and increased depths. Table 5.7 shows the orientation accuracy

for each bone. Upper bones have lower orientation accuracy than lower bones because they are not instrumented with IMUs.

Qualitative comparisons in the Ego-VIP dataset are shown in Figure 5.10, Figure 5.11, Figure 5.12, Figure 5.13, and Figure 5.14. HG3D failed to correctly detect the occluding legs in Figure 5.11, Figure 5.14, and was unable to detect outside arm joints in Figure 5.10. DIP underestimated the knee lift in Figure 5.11 and Figure 5.12, and hand raise in Figure 5.10, respectively. In Figure 5.13 and Figure 5.14, DIP outputs the wrong lower body pose due to the pose ambiguity from sparse IMU input. The method (EgoVIP) described in this chapter shows significantly better pose estimates than HG3D and DIP in all cases.

The EgoVIP8 (dense-IMUs) variant of the method here shows the best performance in all three categories because all bones are instrumented with IMUs. However, the accuracy of the method (EgoVIP) described in this chapter, with only 4 IMUs, is comparable to that of EgoVIP8, and both perform significantly better than either HG3D or DIP.

5.6 Applications

To showcase the real-time capability of the system, a remote Physical Therapy (PT) scenario is demonstrated in VR. The user wearing the prototype system and a trainer wearing an Oculus Quest VR headset are in different physical locations. The described learning-based pipeline estimates the current body configuration (10 body shape parameters and 24×3 pose parameters), which is sent to the trainer's VR headset over a wireless network via UDP. The VR headset uses the Unity Game Engine (Unity, 2005) to render the user's pre-scanned environment and body model from the trainer's viewpoint in real time. The trainer evaluates the user's PT motions and gives real-time audio feedback on how to improve them. The trainer is provided with controller-based and physical locomotion to move around the user's environment. This demonstration shows that the system in this chapter is able to reconstruct challenging and fast PT motions in real time and could be a viable tool for remote PT in the future. Figure 5.15 shows an overview of this (unidirectional) PT demo system. Figure 5.1a-b (top two rows) and Figure 5.16 (first row) show sample results.

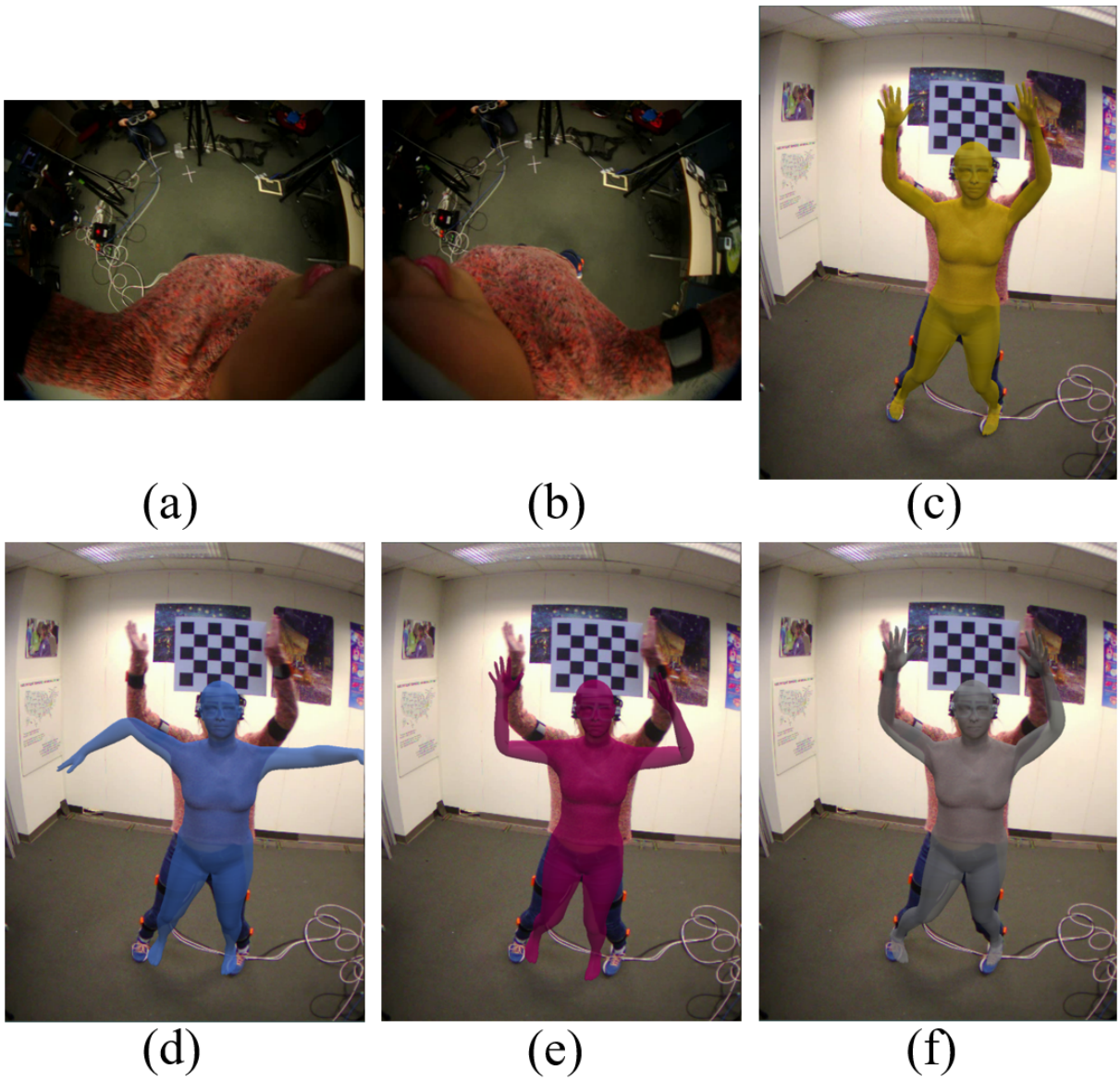


Figure 5.10: Qualitative evaluation in Ego-VIP dataset. (a) Left head-worn view. (b) Right head-worn view. (c) Ground truth (3 external views, 8 IMUs). (d) HG3D = Stereo Hourglass 3D (2 head-worn views). (e) DIP = Huang et al. (2018) (6 IMUs). (f) EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs).

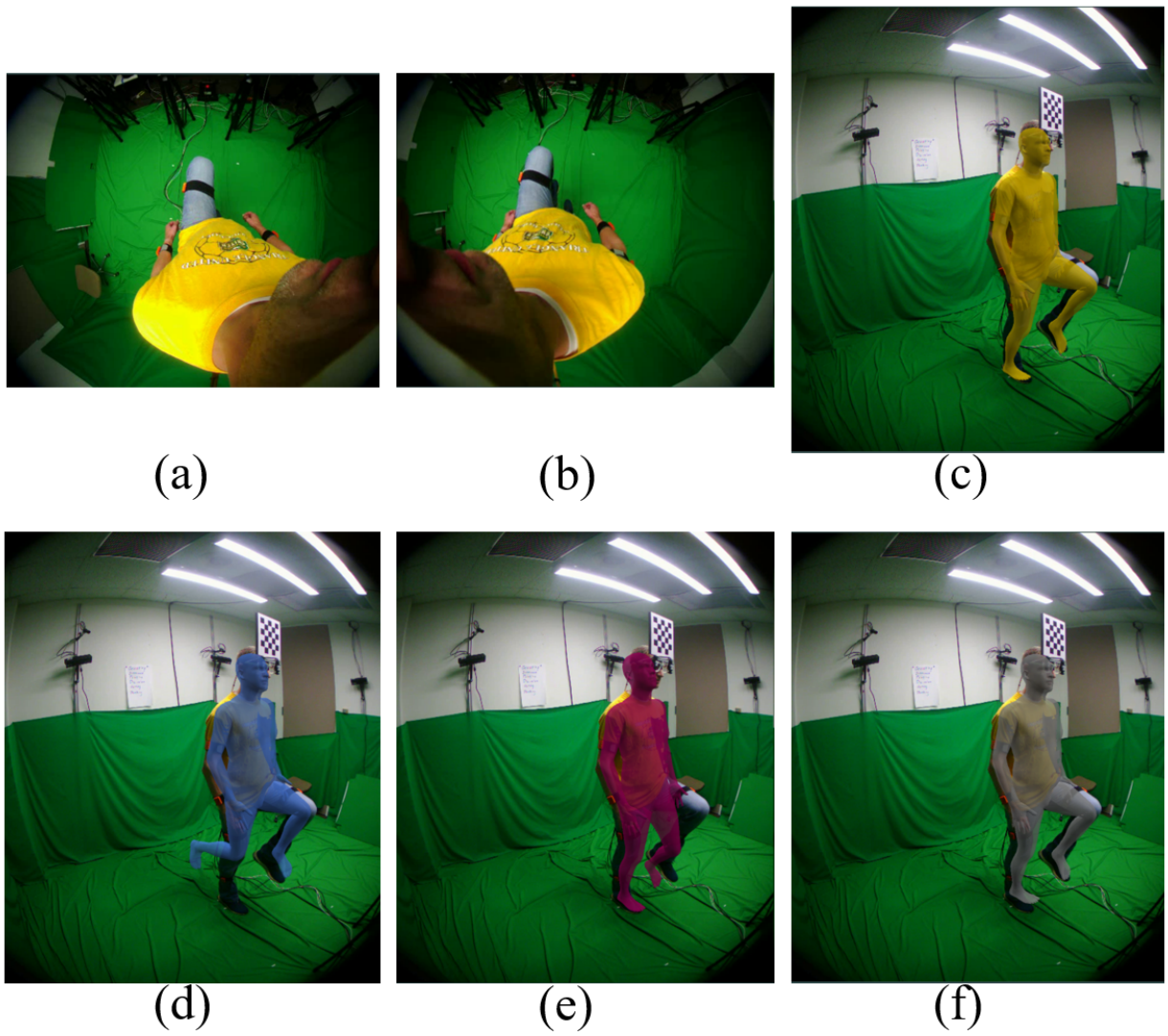


Figure 5.11: Qualitative evaluation in Ego-VIP dataset. (a) Left head-worn view. (b) Right head-worn view. (c) Ground truth (4 external views, 8 IMUs). (d) HG3D = Stereo Hourglass 3D (2 head-worn views). (e) DIP = Huang et al. (2018) (6 IMUs). (f) EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs).

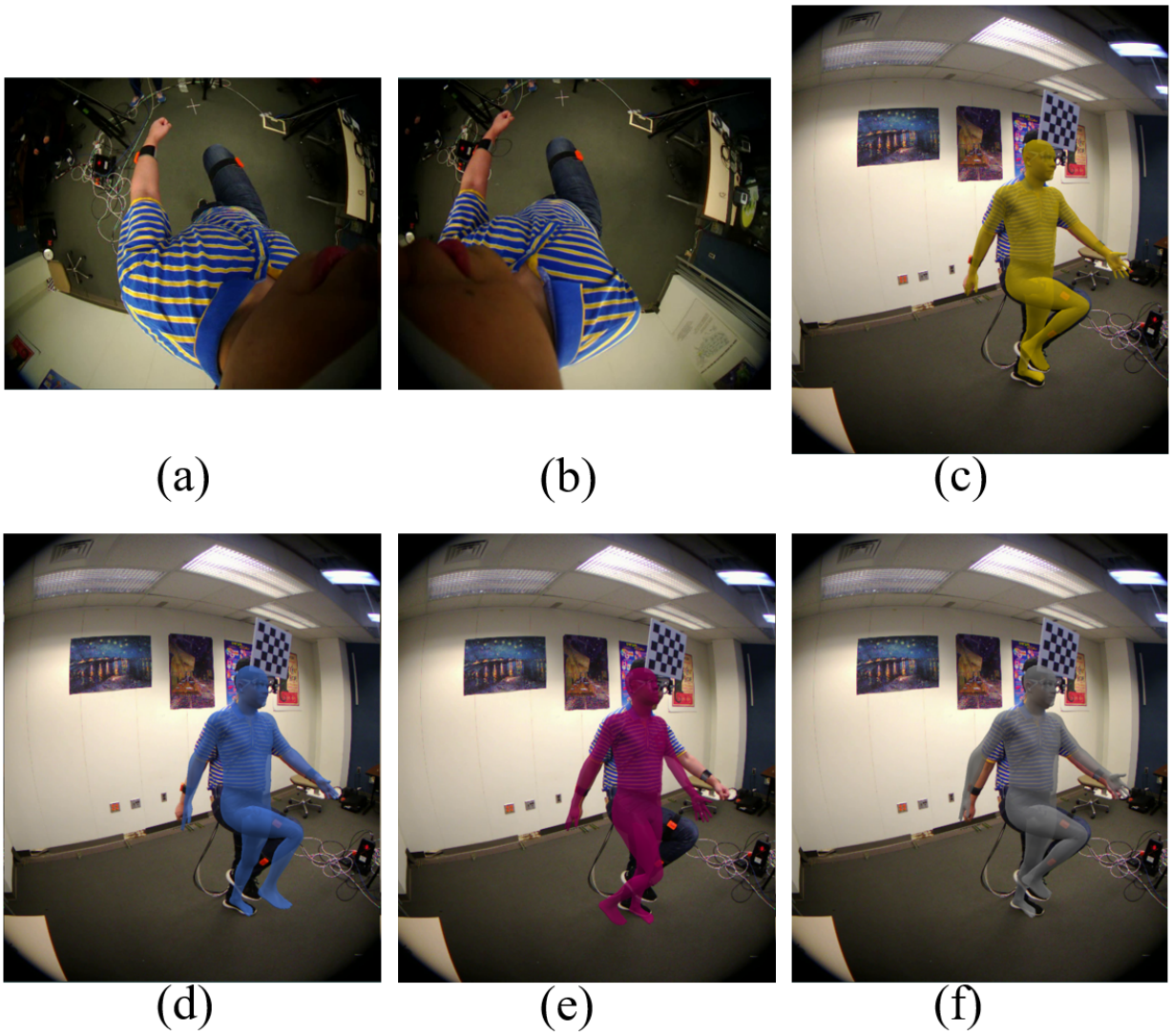


Figure 5.12: Qualitative evaluation in Ego-VIP dataset. (a) Left head-worn view. (b) Right head-worn view. (c) Ground truth (3 external views, 8 IMUs). (d) HG3D = Stereo Hourglass 3D (2 head-worn views). (e) DIP = Huang et al. (2018) (6 IMUs). (f) EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs).

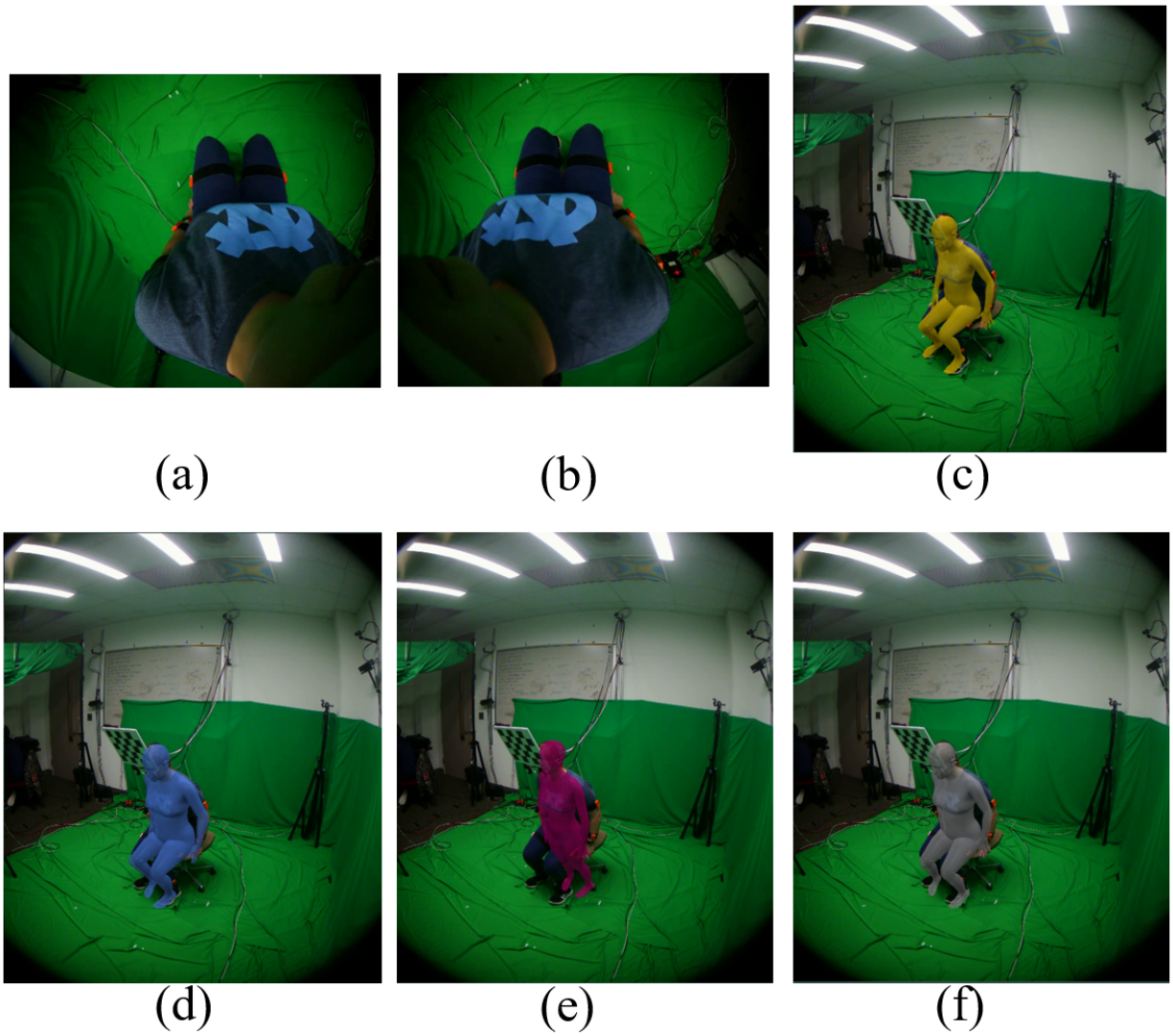


Figure 5.13: Qualitative evaluation in Ego-VIP dataset. (a) Left head-worn view. (b) Right head-worn view. (c) Ground truth (3 external views, 8 IMUs). (d) HG3D = Stereo Hourglass 3D (2 head-worn views). (e) DIP = Huang et al. (2018) (6 IMUs). (f) EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs).

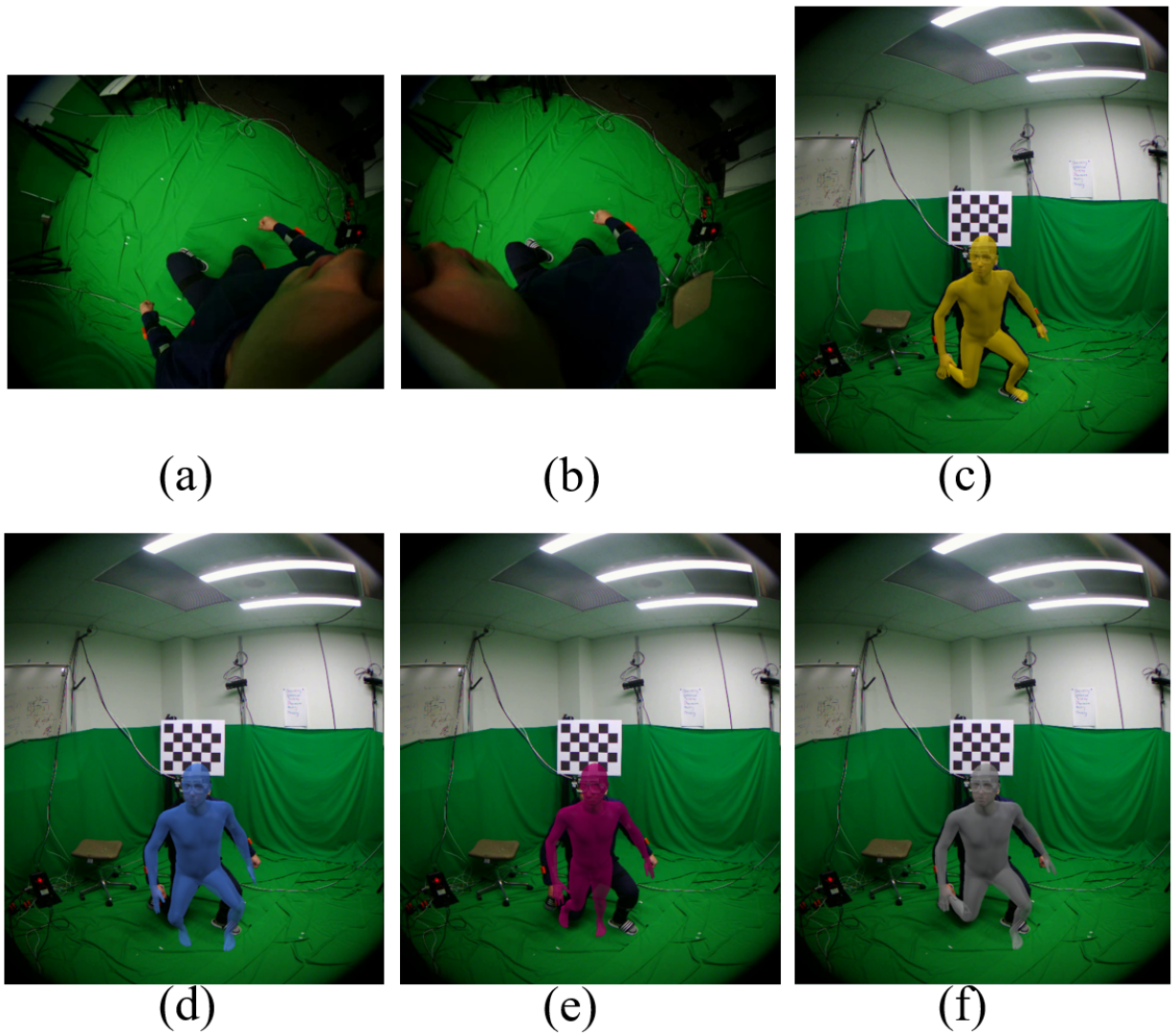


Figure 5.14: Qualitative evaluation in Ego-VIP dataset. (a) Left head-worn view. (b) Right head-worn view. (c) Ground truth (4 external views, 8 IMUs). (d) HG3D = Stereo Hourglass 3D (2 head-worn views). (e) DIP = Huang et al. (2018) (6 IMUs). (f) EgoVIP = The method in Chapter 5 (2 head-worn views, 4 IMUs).

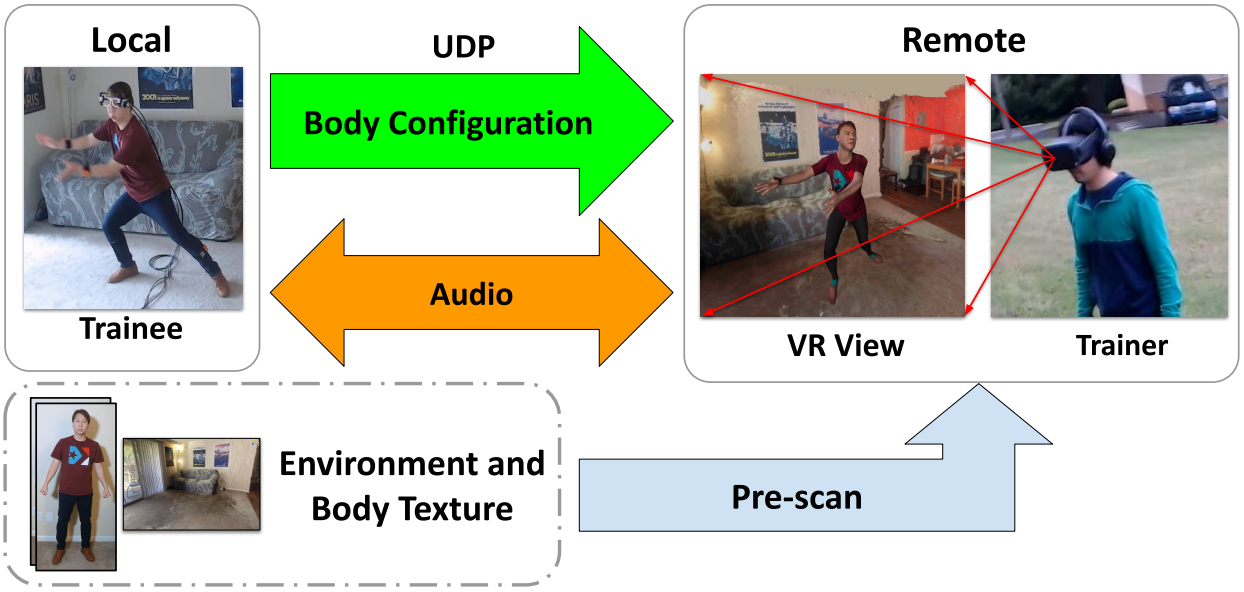


Figure 5.15: Interactive Physical Therapy application in VR. The real-time body reconstruction is only transmitted from trainee to trainer. The trainer's VR display shows the trainee's full-body performance using the pre-scanned environment and body texture. The trainer provides real-time feedback via audio.

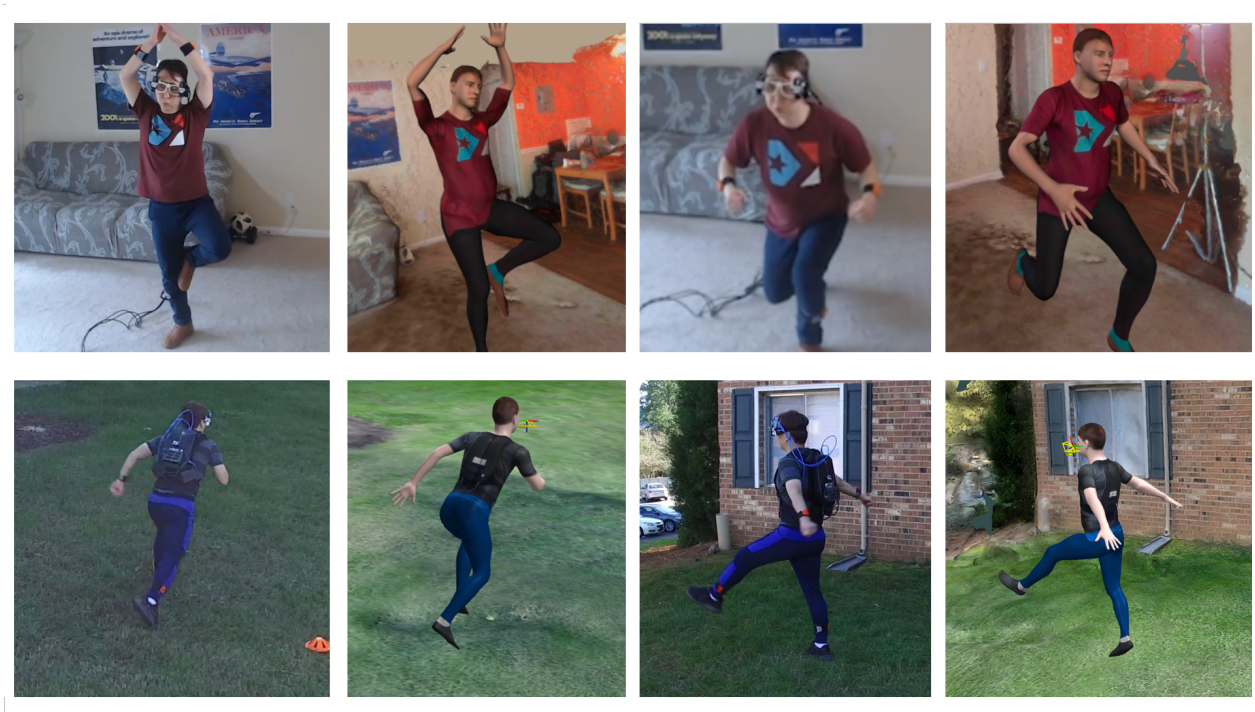


Figure 5.16: Selected Frames in Real-time Demo. Pairs of reference views (not used in reconstruction) and egocentric user reconstructions shown in VR. Indoor user reconstruction (first row). Outdoor user reconstruction (second row).

The system is also demonstrated outdoors, as shown in Figure 5.1a-b (bottom two rows) and Figure 5.16 (second row), using a backpack PC. The motion data was recorded and processed in real time. Wearing the backpack, the user performed a number of standard soccer exercises. The method successfully reconstructed the movements in a grassy area of about 50 square meters. This showcases the mobility of the system.

In both demos, the user's environments were pre-reconstructed using Agisoft's Metashape software (Agisoft Metashape, 2010). The body texture was derived from two full-body images of the user (front and back). *SMPLify-X* (Pavlakos et al., 2019) was used to fit the SMPL body model to the body and facial keypoints (Cao et al., 2019) acquired from the images. The colors from the images were then rasterized to a canonical UV map based on the established correspondence between the fitted meshes and the body part segmentations (Gong et al., 2018).

The prototype system in this chapter runs at 37 fps on a desktop PC (Intel Xeon Gold 6242, 2.8GHz, 128 GB RAM, with NVIDIA Quadro RTX 6000) and at 30 fps on a backpack PC (Intel i7-8850H, 2.6GHz, 32GB RAM with NVIDIA GeForce RTX 2080).

5.7 Conclusion and Future Work

A real-time egocentric 3D capture system is presented as a step toward a fully mobile telepresence system. The system makes use of visual and inertial sensors that are either easy to embed into or are already present in commonly worn personal accessories: eyeglasses, wristwatches, and shoes.

The eyeglasses form factor makes visibility challenging, while the small number of inertial sensors makes the full-body pose difficult to estimate. To address these challenges, the system in this chapter combines visual and inertial information and shows improved full-body pose estimation compared to visual-only or inertial-only information.

The system described in this chapter has many possibilities for improvements. First, unlike the unidirectional PT prototype in Figure 5.15, future application prototypes can be extended to demonstrate bi-directional telepresence. One problem is a choice of a shared environment from the two different environment reconstructions. The shared environment can be selected by the

users as the one of the reconstructed environments, a mixed environment, or another previously reconstructed environment.

For more robustness in head pose estimation, use of multiple forward cameras and integrating an IMU into the headset can be a possible extension in the next iteration of the system. In this chapter, the results of the 3D joint detection network are fed into the temporal orientation network. If the 3D joints are detected erroneously, such errors are propagated throughout. It also can be investigated for a combined network, as well as improving robustness against erroneous detections.

The system introduced in this chapter has some limitations. Since the system only tracks the user's limbs, it does not model interactions with the environment, nor is it able to detect topological or texture changes in the surface of the body model. The system might be improved by using more sophisticated body models such as Osman et al. (2020) for improved body shapes, or Pavlakos et al. (2019) for expressive face and hands.

The joint position accuracy is highly dependent on the *VSLAM* result, which is used to transform the estimated joints into world space. If *VSLAM* is unstable or inaccurate over time, the body pose accuracy drops as well. This observation inspired me to extend the system toward reducing the motion jitter as well as erroneous pose estimations. This extension will be discussed in the next chapter to overcome the unstable tracking by *VSLAM*.

In the next chapter, a physically plausible pose estimation method will be discussed to improve the joint positions due to inaccurate estimation as well as handling 3D environment contacts.

CHAPTER 6: PHYSICALLY PLAUSIBLE EGOCENTRIC MOTION RECONSTRUCTION

This chapter discusses a physically plausible egocentric human motion reconstruction based on rigid body dynamics. The method introduced in this chapter reduces physically implausible motion jitter and interpenetrations between the reconstructed user's model and objects in the environment such as the ground, walls, and furniture. With the efficient method of physics character creation introduced in this chapter, the physics simulation works with the deformable body model while still running in real-time. The experimental results show that the motion reconstruction is further improved, resulting in temporally smooth motions and interaction with the objects in the environment.

6.1 Introduction

The egocentric reconstruction method in Chapter 5 showed how to handle sparse observations of body parts. However, it also suffers from limitations: the estimated body pose can jitter due to *VSLAM* noise and the pose of the lower body may slide or penetrate the ground. To be physically plausible in motion, a human pose should have the feet planted on the ground without sliding or penetration and should move smoothly over time. When any body part is touching an object in the scene, the body motion should be able to react to the environment.

A rigid body dynamics-based physics simulation (Featherstone, 2014) can handle the collisions between objects as well as can generate reactions between them. However, the shape of the parametric body model can deform, so there needs a method to retarget the deformable body model into the rigid bodies of the physics character. In this chapter, a physically plausible pose estimation method is discussed to further improve the pose estimation described in Chapter 5 (Cha et al.,

2021). To achieve this goal, the noisy pose is physically simulated within a given environment so that the body is able to make contact and react to the environment in a physically plausible way. Also, a real-time method for the physics character deformation from the deformable body model is discussed.

6.2 Related Work

The body model interaction with the environment has been studied in Hassan et al. (2019); Zhang et al. (2020); Hassan et al. (2021). Given a static environment, the body pose is constrained by the penetration with the environment. The estimated pose maintains no penetrations with the objects in the scene while body parts are also in contact with each other. However, pose accuracy suffers from the occlusions of body parts, and the detailed environment is needed in advance.

When the environment is not provided, the body pose estimation suffers from a foot skating problem: the positions of feet jitter and do not plant on the ground correctly, which results in unrealistic lower body pose estimation. To reduce the unrealistic foot jitter movements, previous methods focus on detecting contact states onto the ground based on the lower body poses (Kovar et al., 2002; Ikemoto et al., 2006; Zou et al., 2020). These methods may reduce the jittering, but the corrected poses are still not physically plausible by only applying the zero velocity for foot while in contact.

Rempe et al. (2020) showed an approach for physics-based pose estimation with contact estimation. The corrected poses are physically realistic but directly working with all surface points is intractable for real-time applications. Andrews et al. (2016) proposed a real-time physics-based motion capture method using sparse optical markers as well as sparse inertial sensors. Although the method can handle ground contacts, the shape of the physics character is fixed, thus unable to handle arbitrary body shapes such as parametric body models. Shimada et al. (2020) also proposed a rigid body dynamics-based real-time pose estimation approach. The vision-based kinematic pose is improved by the body balance control as well as the estimated ground contact reaction force. However, it is unable to react to the environment other than to an assumed flat ground

plane. Al Borno et al. (2018) proposed a physics character retargeting method for the shape of the physics character to be adaptable toward the deformable body model. The physics character can be retargeted with different shapes by fitting primitive capsule shapes using all the body mesh vertices. This dense vertex fitting method is unable to cope with real-time applications due to its computational cost.

None of the methods above is able to estimate a physically plausible pose both with contacts on any body parts and with different shapes of the body. In the next section, I present a rigid body dynamics-based pose estimation method that considers any contact points on the body as well as an efficient shape deformation method for physics characters.

6.3 Method

The pipeline for the physically plausible egocentric reconstruction is shown in Figure 6.1. A set of 3D planes representing the simplified static environment is provided as input. At each time instant, the method in Chapter 5 is used to compute the egocentrically estimated shape and pose of the parametric body model. The shape of the physics character is deformed based on the estimated shape of the body model. Using the temporal movements of the pose of the body model, the desired accelerations for all joints are calculated by proportional derivative (PD) controllers (Liu et al., 2010; Shimada et al., 2020). Based on the multibody dynamics, the torques (rotational forces) of all joints are computed using the recursive Newton-Euler algorithm (Featherstone, 2014) for inverse dynamics. The given environment planes, the current state of the physics character, and the estimated forces are simulated by the physics engine for collision detection as well as for handling the contact responses. The pose of the physics character with environment contacts handled is retargeted back to the body model.

6.3.1 Deformable Physics Character

Directly working with the entire 3D mesh is computationally heavy and not adequate for real-time processing. In physics simulations, using primitive shapes is preferable because it has

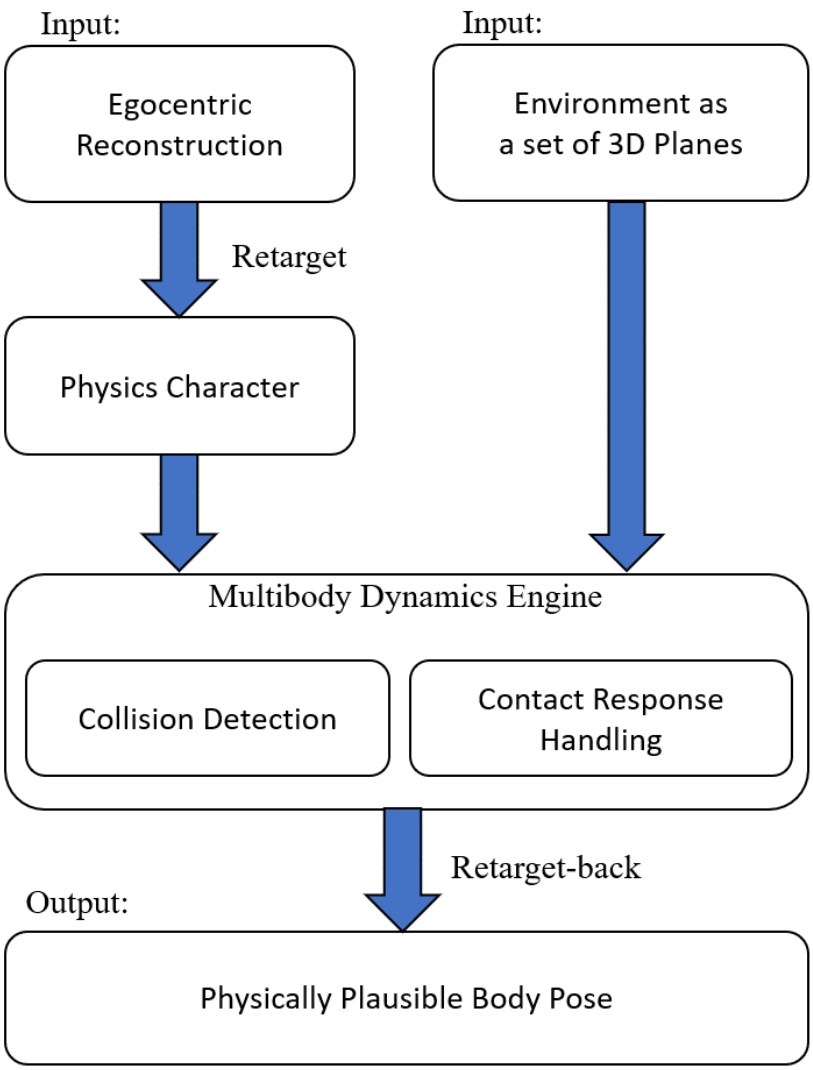


Figure 6.1: Physically plausible egocentric reconstruction pipeline. The shape and pose of the body model reconstructed by the egocentric reconstruction from the method in Chapter 5 are transformed to the shaped and posed physics character. The physics simulation engine for multibody dynamics handles collisions between the physics character and the given plane-based environment. The physically plausible body pose from the simulated physics character is retargeted back to the body model.

computational benefits and makes it easy to handle collisions. The physics character in this section consists of only capsules and boxes for approximating the body shape, taking advantage of the primitive shapes to run in real-time. A capsule shape is parameterized by a radius and a height. A box shape is parameterized by three lengths. Only the feet and hands are made by boxes and other body parts are approximated by capsule shapes. The lengths of the boxes and capsules are determined by the provided shapes of the body model.

The parametric body model (Loper et al., 2015) used in Chapter 5 can have its shape deformed over time. The physics character needs to be fitted to the body shape in a low computational cost approach, to maintain real-time capability. The advantage of the SMPL body model is that the topology of the model is fixed even when its shape changes. Al Borno et al. (2018) showed that using all the mesh vertices for estimating the primitive shape lengths is not appropriate for real-time applications. However, the correspondences between the mesh vertices and the surface points on primitive shapes are unchanged if the topology is fixed. Using this observation, a smaller, fixed number of vertices can be used to estimate the lengths of the primitive shapes.

In shape changes, the joint and vertex locations are changed from the rest pose. To exploit the fixed topology observation, 57 key vertices are pre-selected among $6,890$ mesh vertices. The key vertices are empirically selected considering the correspondence between the primitive shapes. Only the key vertices and the 24 joint locations are used to determine all the lengths of the capsules and boxes instead of using all vertex points. These pre-defined correspondences enable building a physics character at a low computational cost because there is no need to estimate correspondences between the mesh and the primitives. The constructed physics characters for different body shapes are shown in Figure 6.2 and Figure 6.3. The overlaid shapes of the physics characters are shown in Figure 6.4.

The physics character features the same number of joints and the same bone structure as the body model so that the physics character is able to deform the same way as the body model. A joint is parameterized as $EulerYXZ$ for 3D rotation. In my experiments, using three Euler angles

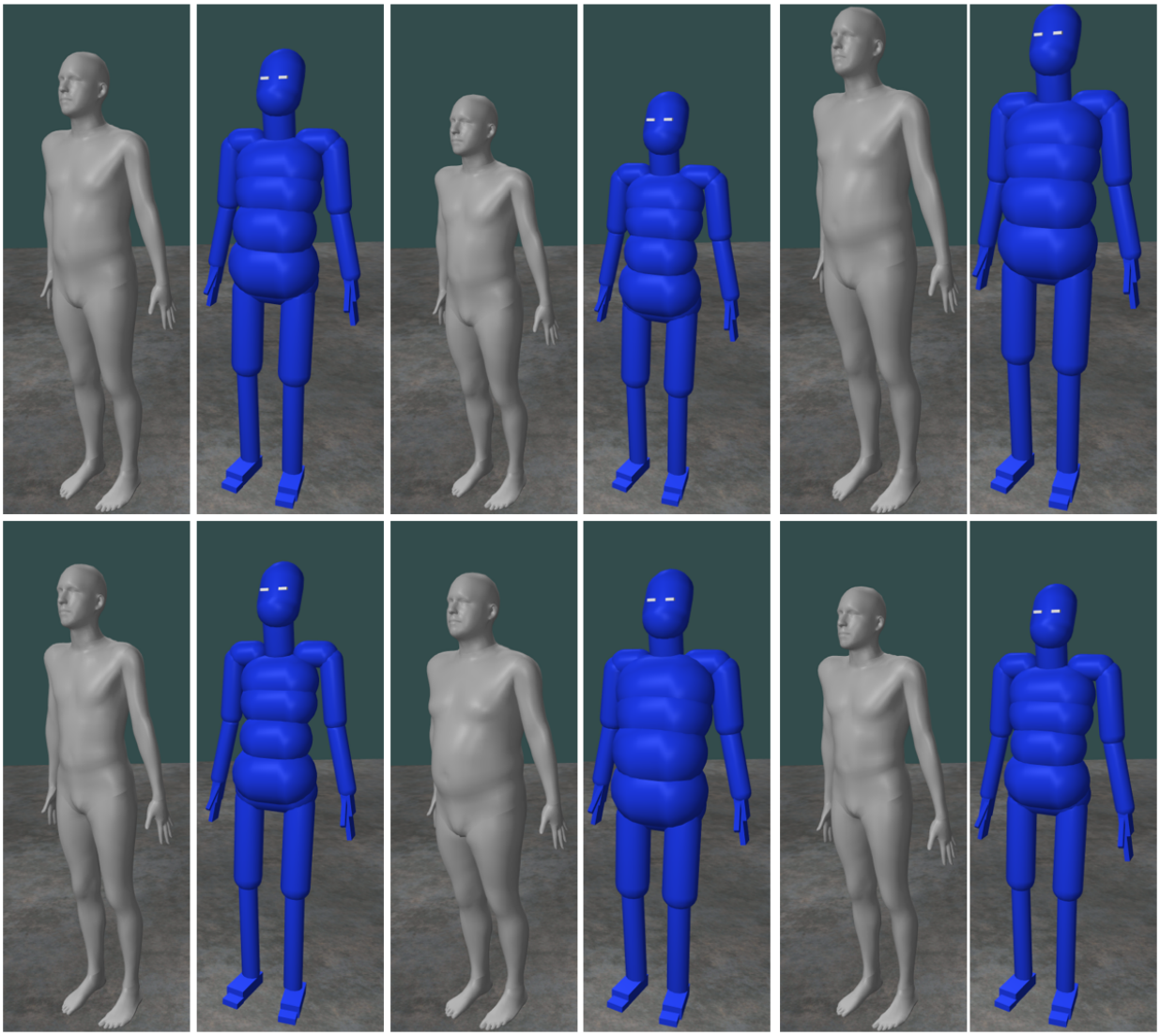


Figure 6.2: Real-time Shape Deformation of Physics Character for Male Body Model. Pairs of different shaped male body models and shape-fitted physics characters. The shape parameters β are varied from left to right and top and bottom: $\beta = 0$, $\beta_1 = 2$, $\beta_1 = -2$, $\beta_2 = 1.5$, $\beta_2 = -1.5$, and $\beta_3 = -5$.

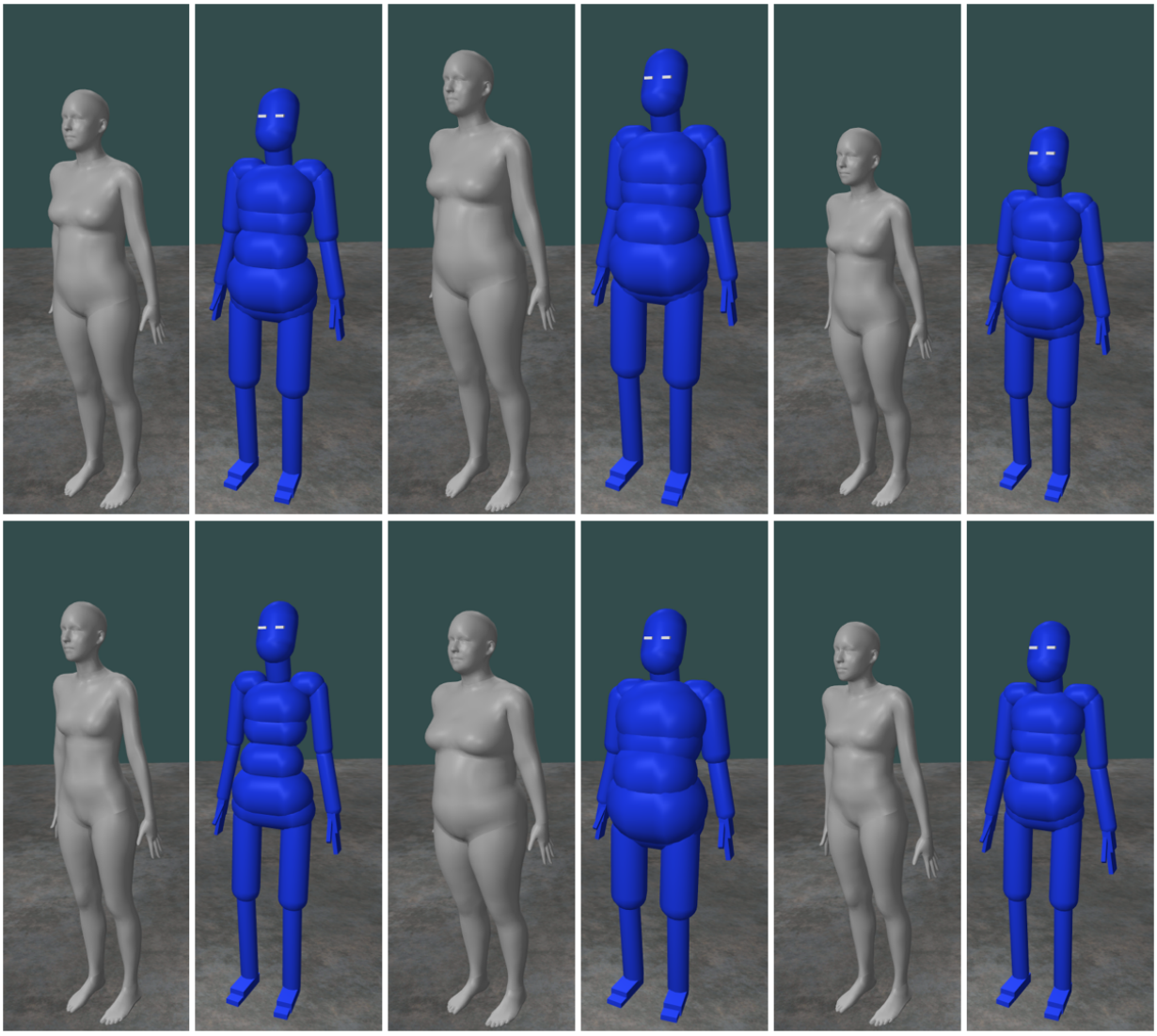


Figure 6.3: Real-time Shape Deformation of Physics Character for Female Body Model. Pairs of different shaped female body models and shape-fitted physics characters. The shape parameters β are varied from left to right and top and bottom: $\beta = 0$, $\beta_1 = 2$, $\beta_1 = -2$, $\beta_2 = 1.5$, $\beta_2 = -1.5$, and $\beta_3 = -5$.

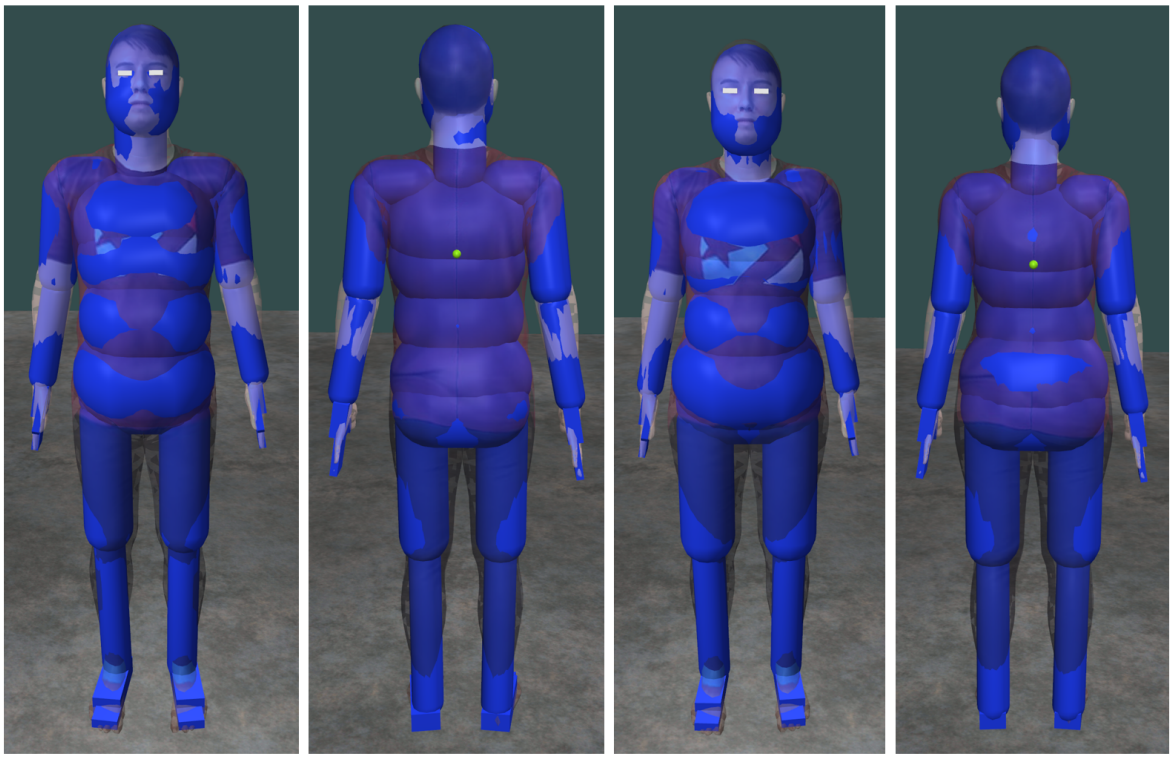


Figure 6.4: Physics Character overlaid with Body Model. The physics character approximates the shape of the body model. From left to right: male character (front), male character (back), female character (front), female character (back).

showed better performance than using 3D spherical joints. The mass distribution of the character is set following Liu et al. (2010), and it remains unchanged during shape changes.

At run-time, the physics character measures the shape changes $\dot{\boldsymbol{\beta}}^t$ from the shape parameters $\boldsymbol{\beta}$ at timestamp t as,

$$\dot{\boldsymbol{\beta}}^t = \sum_i abs(\beta_i^t - \beta_i^{t-1}) \tag{6.1}$$

If the shape change $\dot{\boldsymbol{\beta}}^t$ is significantly larger than a predefined amount, the current physics character is discarded in the physics world. A new shaped character is created in the rest pose and recovers the same joint states from the previous character; the joint angles (previous pose) and joint velocities are copied from the previous character. The newly created character is inserted into the physics world. In my experiments, the shape parameters vary smoothly, so using intermittent shape updates resulted in computational efficiency, compared to re-creating physics characters every frame in the absence of significant shape changes.

6.3.2 Physics Character Control

The physics character is retargeted from the noisy reference pose of the body model. q_{ref}, $\dot{q}_{ref}$, $\ddot{q}_{ref}$ represent the joint angle, the joint velocity, and the joint acceleration of the reference pose. To handle the noise in the reference pose, the joint angle q_{ref} is updated as,

$$R_{ref}^t = Slerp(R_{ref}^t, R_{ref}^{t-1}, 0.5) \tag{6.2}$$

R_t is the current rotation matrix of the joint. The rotation matrix is smoothed before being converted into Euler angles q_{ref}. The reference velocity $\dot{q}_{ref}$ is maintained using finite-difference method as $\dot{q}_{ref}^t = q_{ref}^t - q_{ref}^{t-1}$. The reference acceleration $\ddot{q}_{ref}$ is also similarly maintained.

The desired acceleration $\ddot{q}_{des}$ for the character is computed from the reference pose using Proportional-Derivative Controller (PD-controller) as (Liu et al., 2010; Shimada et al., 2020),

$$\ddot{q}_{des} = \ddot{q}_{ref} + k_p(q_{ref} - q) + k_d(\dot{q}_{ref} - \dot{q}) \tag{6.3}$$

where q and $\dot{q}$ indicates the current joint angle and joint velocity of the physics character. k_p and k_d are the proportional gain and the derivative gain respectively. The proportional and derivative terms act as spring and damper respectively. In my experiments, I use proportional gain $k_p = 50$ and derivative gain $k_d = 5$ for all joints. For the root joint, I use $k_p = 50$ and $k_d = 1$ for the linear acceleration and $k_p = 50$ and $k_d = 0.5$ for the angular acceleration, respectively.

Using the desired acceleration, the torques τ for all joints are estimated using the recursive Newton-Euler algorithm (Featherstone, 2014). The estimated torques τ are applied to the physics character and run the rigid body simulation with the given environment. The contact detection and response are handled by the physics engine. Since the reference pose is estimated at 30 hz but the physics simulation step is 60 hz, the physics control process can be repeated k times for a single reference pose. In experiments, iterating $k = 4$ times showed the best results. The moving delay from the character to the reference is reduced with the iteration.

6.4 Results

This section shows experimental results using the rigid body dynamics-based pose estimation, *Physically Plausible Egocentric Poser* (**PhysEgo**), introduced in this chapter. The egocentric pose estimation is computed using the method (EgoVIP) in Chapter 5. The ground plane is taken from the Ego-VIP dataset in Chapter 5. Since the egocentric pose estimation is computed using Visual SLAM (*VSLAM*), it results in motion jitter due to unstable tracking. From the inaccurate *VSLAM* and lower body pose estimation, the estimated pose frequently penetrates the ground or slides on the ground as shown in Figure 6.5b.

Using rigid body dynamics (PhysEgo), the feet of the physics character do not penetrate the ground as shown in Figure 6.5c. The motion jitter is also reduced as well. Penetration prevention is

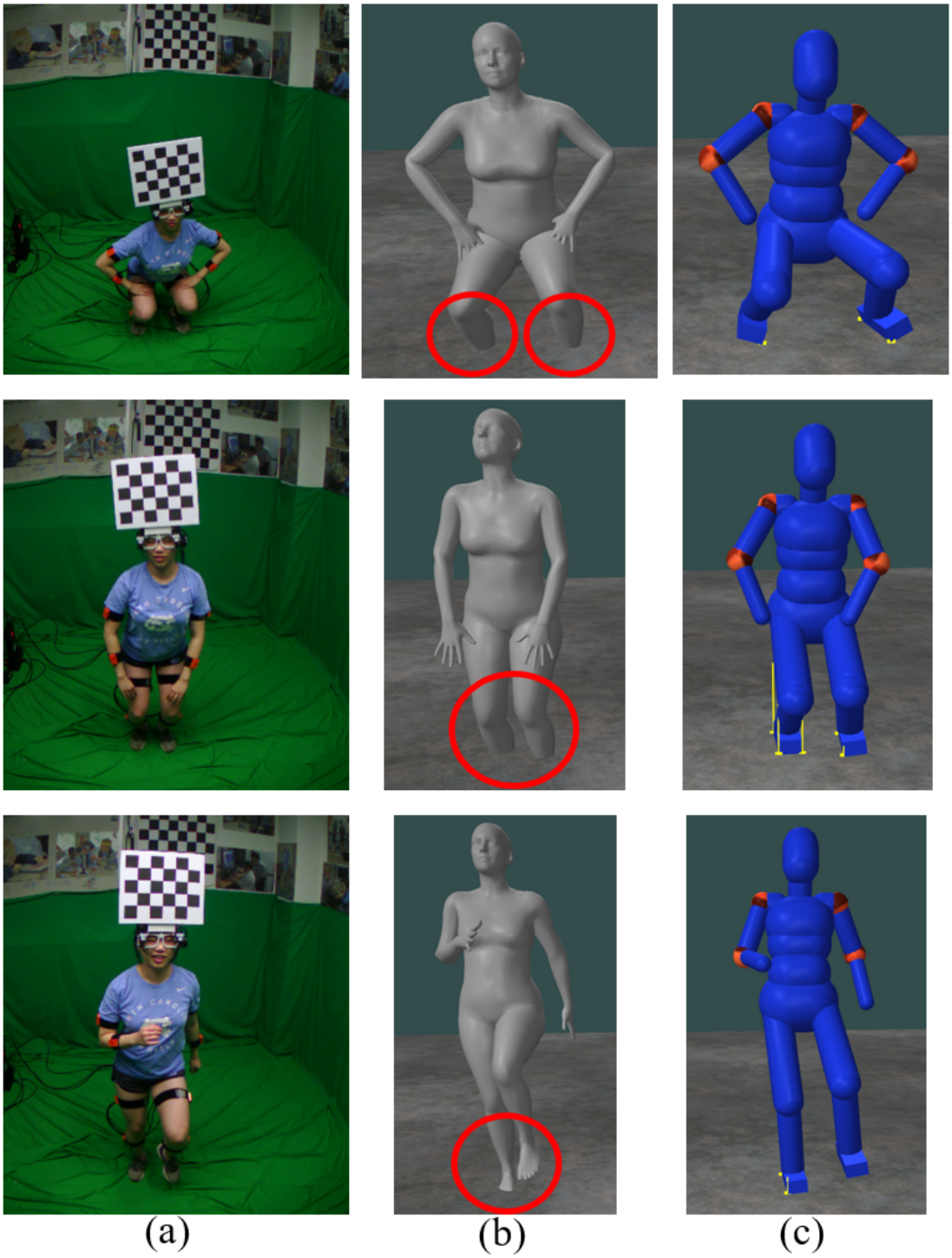


Figure 6.5: (a) User motions in external view from Ego-VIP dataset in Chapter 5. (b) The motions from the egocentric reconstruction using the method (EgoVIP) in Chapter 5. The ground penetrations are highlighted in red circles. (c) The motions of the physics character from the rigid body dynamics-based pose estimation using the method (PhysEgo) in Chapter 6. The penetrations are resolved and the ground contact points are marked in yellow.

Table 6.1: Comparison of motion jitter on the Ego-VIP dataset in Chapter 5. Average temporal joint jitters are reported as the average μ_{jitter} and the standard deviation σ_{jitter} in mm. The rigid body dynamics-based pose estimation method (PhysEgo) in Chapter 6 significantly reduces the motion jitter of the method in Chapter 5 (EgoVIP). The best results are shown in bold.

	μ_{jitter} (mm)	σ_{jitter} (mm)
Ground Truth	**1.24**	1.29
EgoVIP (Ch. 5)	4.90	4.57
PhysEgo (Ch. 6)	1.29	**0.76**

performed by the physics engine automatically, and the motion jitter reduction comes from using PD-control with smooth motions, applying Equation 6.2 and Equation 6.3.

To quantitatively evaluate the motion jitter and the penetration as in Shimada et al. (2020), the Ego-VIP dataset in Chapter 5 is further extended with feet contact labels. The 6 available sequences (13,122 frames) in the Ego-VIP dataset are manually labeled with the ground contacts by both feet (19,987 feet contacts). The motion jitter is measured as the temporal joint jitter e_{jitter} and it is defined as,

$$e^t_{jitter} = \frac{1}{|J|}\sum_{i=1}^{|J|} ||j^t_i - \frac{j^{t-1}_i + j^{t+1}_i}{2}||_2 \tag{6.4}$$

where j^t_i denotes the ith joint position at timestamp t and $|J|$ indicates the number of joints. e_{jitter} measures how much the joint position is off from the smooth transition over time. Table 6.1 reports the comparison of motion jitter between the method (PhysEgo) introduced in this chapter and the method (EgoVIP) in Chapter 5 by evaluating e_{jitter} over the entire frames of the 6 sequences. The method in Chapter 5 showed noticeable motion jitter ($\mu_{jitter} = 4.9$ mm) over time due to inaccurate pose estimation as well as unstable tracking by *VSLAM*. The rigid body dynamics-based method in this chapter significantly reduced the motion jitters ($\mu_{jitter} = 1.29$ mm) and it is closed to the level of ground truth ($\mu_{jitter} = 1.24$ mm). This demonstrates the capability of the method introduced in this chapter in reducing the implausible motion jitter.

The ground penetration is also quantitatively evaluated on the Ego-VIP dataset with the labeled feet contacts. The average ground penetration error (AGP) measures the average distance from ground to foot if the foot penetrates the ground. The vertices located at the soles of both feet

Table 6.2: Comparison of ground penetration on the Ego-VIP dataset in Chapter 5. The Average Ground Penetration Errors (AGP) by feet are reported as the average μ and the standard deviation σ in mm. The Non-Ground-Penetration Rate (NGP) is reported with varying penetration distances (<0 mm, <5 mm, <10 mm, <15 mm, <20 mm) in percentage (%). The rigid body dynamics-based pose estimation method (PhysEgo) in Chapter 6 significantly reduces the physically implausible penetrations of the method in Chapter 5 (EgoVIP). The best results are shown in bold.

	AGP (mm)		NGP (%)				
	μ	σ	$< 0mm$	$< 5mm$	$< 10mm$	$< 15mm$	$< 20mm$
EgoVIP (Ch. 5)	22.01	28.04	67.53	74.57	80.83	86.06	89.63
PhysEgo (Ch. 6)	**5.16**	**3.91**	**90.48**	**96.64**	**97.74**	**99.43**	**99.98**

of the body model are pre-selected, and the deformed locations of the vertices are used as the feet contact points of the body model. The percentage of non-ground penetration (NGP) is also measured with different distance thresholds (from the ground to the foot) including 0, 5, 10, 15, 20 in mm respectively. The comparisons of the two methods in Chapter 5 (EgoVIP) and this chapter (PhysEgo) are reported using AGP and NGP in Table 6.2 respectively. The AGP of the dynamics-based method described in this chapter showed only 5.16 mm average penetration error, and it is a significantly reduced error compared to that of the method in Chapter 5 (22.01 mm). The reduction mainly results from the collision resolution by the physics engine, and the $\sim$ 5mm error of the introduced method is mostly caused by the shape approximation error, especially for feet shapes, between the physics character and the body model. The NGP in Table 6.2 shows that the penetration errors caused by the shape approximation can be mitigated using 15 mm distance tolerance (99.43 % NGP). However, the method in Chapter 5 still showed high penetration errors (89.96 % NGP) even with 20 mm distance tolerance. The results in Table 6.2 show that the physics-based method in this chapter is able to prevent most of penetrations with an acceptable distance tolerance.

The system not only resolves the ground penetration of feet, but is also able to handle any contact between body parts and the environment, such as touching hands or hips when sitting to the furniture in the environment. Figure 6.6 shows examples of the user interacting with a couch. The environment in Figure 6.6c is pre-reconstructed using the method in Section 5.6. The physics world in Figure 6.6b is manually constructed using box shapes to roughly approximate the

(a) (b) (c)

Figure 6.6: (a) User motions in external view. (b) The motions of the physics character in the simplified physics environment. The penetrations are resolved using the rigid body dynamics-based pose estimation. The refined motions are re-targeted back to the body model. (c) The corresponding body reconstruction in the pre-scanned environment.

(a) (b) (c) (d)

Figure 6.7: Interaction with objects in the scene. (a-b) Using hands to receive a virtual object. (c-d) Using feet to kick a virtual object.

pre-scanned environment. This simplified physics environment is pre-aligned with the environment reconstruction. Both the physics world and the environment reconstruction are treated as static scenes. At run-time, the rigid body dynamics-based pose estimation handles any contact between the physics character and the physics world such as hands shown in the first and the second rows in Figure 6.6, and hips shown in the third and the fourth rows in Figure 6.6. These examples demonstrate the penetrations of body parts that can be handled using the method in this chapter.

Additionally, the rigid body dynamics-based pose estimation method enables interacting with objects in the scene. Figure 6.7 shows examples. Rigid body virtual objects such as boxes or balls can be integrated into the physics world. In Figure 6.7, the virtual objects are manually introduced at specific times and locations. The physics engine automatically handles the collisions between the physics object and the physics character. The physics character reacted similarly to the actual object movements in the real world. This demonstrates the capability to interact with movable objects in the scene.

The rigid body dynamics-based pose estimation runs at 40 fps on a desktop PC (Intel Xeon Gold 6242, 2.8GHz, 128 GB RAM, with NVIDIA Quadro RTX 6000). The algorithm is implemented in

C++ using RBDL library (Felis, 2017) for rigid body dynamics computations, and Bullet Physics (Coumans and Bai, 2021) as a physics engine for collision detection and response.

6.5 Conclusion and Future Work

This chapter introduced a physically plausible pose estimation method using rigid body dynamics. Using efficient physics character creation, the physics simulation works with the deformable body model while still running in real-time. Since the physics character introduced in this chapter has the same joint structure as the parametric body model, the character is able to express any body model pose without motion restrictions.

This system has some limitations. It assumes the environment is provided in advance and does not change at run-time. However, the environment can be changed by moving objects in the scene. As future work, the system can be extended with the integration of real-time environment reconstruction. To cope with running the pose estimation pipeline in real time, the environment reconstruction should be able to represent the scene using primitive shapes. 3D plane estimation methods (Liu et al., 2019) can be used for environment reconstruction.

The method described in this chapter can handle penetrations for physical plausibility. However, feet can still slide on the ground if the noisy input pose is both floating and penetrating. The physics engine corrects the pose such that it is moved along the direction of the collision response to resolve the penetration. To alleviate the sliding feet problem, the pose correction should take into account the fact that the feet are stationary while in contact. As future work, a constrained pose correction method can be investigated to prevent the penetration and contact sliding simultaneously for better physical plausibility.

In addition, the physics character can lose its control if consecutive noisy rotations for base joint are input over time. Instead of using only current noisy input pose, estimating optimal trajectories of joints by using a few past and future nearby frames can reduce such problematic input noise. This also can be a future extension in order to prevent the control loss problem. Although the use of future frames will cause pose estimation delay, using less than 5 future frames would be acceptable;

the use of 5 future frames takes only 0.16 seconds in a system processing at 30 fps, and the delayed pose is not very noticeable in motions at normal speeds. Extending the optimal trajectory estimation method would contribute to the motion stabilization of the physics character.

The system in this chapter currently runs only for a single person. The system also can be extended for multiple people by computing collisions between them. However, this system does not support topological changes of the user since the system now only supports rigid body dynamics.

The definition of physical plausibility depends on the target application. In this chapter, the physical plausibility was evaluated using the temporal stability and the penetration classification rate. These metrics are not the only methods to account for the physically plausible motions in human bodies. Other sophisticated metrics such as body balance, muscle dynamics, or contact reactions can be used to better evaluate the physical plausibility. Such human motion metrics combined with kinematics and dynamics can be investigated as future work.

CHAPTER 7: DISCUSSION AND CONCLUSION

7.1 Summary

This dissertation discussed a real-time mobile 3D capture system of a user's body using eyeglass-mounted cameras and a few Inertial Measurement Units (IMUs). The system does not rely on any instrumented environments and the wearable sensors are mounted at the locations of commonly-worn accessories for widespread acceptability. Advanced techniques for egocentric human body reconstruction were introduced to overcome the sparse visibility and the insufficient sensor data challenges.

In this dissertation, the introduced methods for real-time mobile 3D capture systems make significant contributions to egocentric human body reconstruction: (1) The parametric-model-based reconstruction method overcomes incomplete body surface visibility. (2) The learning-based visual-inertial body motion reconstruction overcomes the challenges of self-occlusion and outside-of-camera motions, and allows for unobtrusive real-time 3D capture of the user. (3) The rigid body dynamics-based, physically plausible reconstruction method reduces motion jitter and prevents interpenetrations between the reconstructed user's model and the objects in the environment.

The potential usefulness of the approach is demonstrated in a telepresence scenario featuring physical therapy training. The experimental results demonstrated the capability for real-time, self-contained, mobile reconstruction of human bodies in indoor and outdoor scenes.

7.2 Discussion

The introduced egocentric reconstruction approaches still have some limitations to improve upon. This section discusses a few failure cases for the system, potential egocentric configurations, and remaining challenges.

There are some extreme cases that would interfere with the system operation. The system presented in this dissertation assumes that the user's body can be partially observed by the head-worn cameras. However, this partial visibility assumption can be violated with extreme head orientations or extreme lighting conditions. In head-up cases such as looking at the sky, the user's body is entirely invisible from the cameras, which results in pose estimation failure for the base joints (neck and both hips). To alleviate this problem, adding an inertial sensor on the back is encouraged so that the sensor is able to track the root joints regardless of their visibility.

The system may also be unable to operate when the user is in extremely bright or very dark spaces. Although the visibility-aware joint detection network is robust to illumination changes from data augmentation, these severe lighting cases can make the input images completely unusable, even with valid head orientations. Adding another inertial sensor for the head can relieve the problem by estimating the user's pose only from inertial sensors, assuming that the validity of input images can be estimated. Using these six inertial sensors (wrists, ankles, back, and head), the presented extreme cases can be handled.

Depending on the choice of devices, other types of egocentric configurations can be possible. In this dissertation, the visual and inertial sensors are required to be unobtrusively embedded in commonly worn accessories for widespread acceptability. To this end, the introduced system installed the visual sensors directly mounted on the eyeglasses frame as well as the inertial sensors directly attaching to wrists and ankles. Although obtrusively mounted cameras can provide more visibility coverage and make the pose estimation problem easier, they bring significant inconvenience to the users, leading to a decrease in acceptability. For this reason, I skipped the camera configuration off the eyeglasses frame and took on the insufficient body visibility problem. The visibility challenge can be handled by the introduced methods in this dissertation.

The choice of camera lenses also affects the estimation performance. With narrower FoV lenses, the body visibility decreases, and the inertial sensor measurements will get fewer chances to be corrected by the visual sensors. However, the entire system pipeline still works if the minimal condition is met; the base joints are visible from the cameras. Using wider FoV lenses or using more mounted cameras, the body visibility increases, leading to fewer cases where limb joints are outside of camera. In this case, the two IMUs worn on the wrists can be discarded. The configuration of the number of cameras and the number of IMUs are closely related to each other. In this dissertation, the configuration is chosen for using the minimal number of inertial sensors to cope with the limited views from the two eyeglasses-mounted cameras.

There are some remaining challenges for the egocentric reconstruction problems. The introduced system used a generic body model for estimating body shapes. The estimations of detailed body appearances such as geometry of clothed body shapes and clothing textures are active in research using external cameras but unsolved in the egocentric case. The estimation of realistic body motions such as muscles or clothing dynamics is also encouraging as an investigation direction. These advanced research topics have inherent difficulty caused by the incomplete egocentric visibility. Although the estimation of probable full-body shape and motion only from sparse visibility is challenging, incorporating external knowledge such as applying temporal body part correlations, or improved observation by another user would lessen the problem complexity. Solving these problems will lead to creation of convincing virtual avatars of the user and increase the acceptability of the system.

7.3 Future Work

There are several next research steps towards improving the capabilities of mobile 3D capture systems.

Integration with Environment Reconstruction: Chapter 6 showed that human performance capture with the environment could enable interactions between the user and the objects in the

environment. The existing environment reconstruction methods are not directly applicable as the representation is too detailed for real-time dynamics. It is reasonably expected that the investigation of simplified geometry-based environment reconstruction can be performed in the near future. With the integration of such an environment reconstruction method, the mobile reconstruction can be applied to many other domains, as described in Chapter 1.

Integration with Hand/Face Motion Reconstruction: The mobile human motion estimation can be a starting point for reconstructing hand and face motions. The study of mobile hand or face reconstructions can be encouraged by the availability of the system described in this dissertation. Reconstructing hands will enable diverse types of interactions with objects in the scene. That is, more complicated and detailed interactions will be possible anywhere and anytime. The simultaneous reconstruction of body and face will enable realistic social interactions, which is the key feature in communication. Realistic verbal communication with body motions can help widespread acceptability for remote social interactions.

Multiple People Interactions: Extending the system to multiple users includes many interesting research topics such as sharing environments, real-time interactions in remote places, and reducing latency in sharing 3D content. The interactions in a crowded environment can be an advanced topic in mobile 3D capture systems. Implementation of two-way remote interactions would encourage the move toward real-time immersive 3D telepresence, so that the systems can be eventually deployed in actual remote workplaces.

Smart Virtual Avatar: When the system can handle convincing virtual avatar creation of the user as well as interaction between multiple people, it can also be extended to the interaction with AI-based virtual avatars by integrating the feature of natural language processing (NLP) as shown in Figure 1.2. I anticipate that a system that allows interactions between multiple people can be naturally extended to the interaction with AI avatars by learning long-lasting user behaviors. I also envision that AI avatars can interact with the user not only verbally by using NLP, but also using motions.

7.4 Conclusion

This dissertation presented advances in real-time egocentric 3D capture as a step toward a fully mobile telepresence system. The introduced approaches showed that the combined use of a parametric body model, head-worn cameras, and body-worn inertial sensors enables a parametric model-based full-body reconstruction approach using only egocentric inputs, thereby helping improve consistent body pose and shape estimation even given the challenge of sparse observations such as incomplete body visibility and insufficient sensor data. In the future, as cameras and IMUs become smaller and more ubiquitous with the next generations of smart AR glasses, I anticipate non-encumbering and easy-to-use real-time successors to mobile telepresence systems to become commonplace and useful for many everyday communication tasks.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(11):2274–2282.

Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., and Szeliski, R. (2011). Building rome in a day. *Communications of the ACM*, 54(10):105–112.

Agisoft Metashape (2010). Agisoft Metashape Photogrammetry Software. `https://www.agisoft.com/downloads/installer/`. Accessed: 2021-06-01.

Al Borno, M., Righetti, L., Black, M. J., Delp, S. L., Fiume, E., and Romero, J. (2018). Robust physics-based motion retargeting with realistic body shapes. *Computer Graphics Forum*, 37(8):81–92.

Alembic (2010). Alembic computer graphics interchange framework. `http://www.alembic.io/`. Accessed: 2021-06-01.

Alexandrova, I. V., Rall, M., Breidt, M., Tullius, G., Kloos, U., Bülthoff, H. H., and Mohler, B. (2012). Enhancing medical communication training using motion capture, perspective taking and virtual reality. *Studies in health technology and informatics, In Proceedings of Medicine Meets Virtual Reality / NextMed (MMVR)*, 173:16.

Allen, B., Curless, B., and Popović, Z. (2003). The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2003*, 22(3).

Alp Güler, R., Neverova, N., and Kokkinos, I. (2018). Densepose: Dense human pose estimation in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7297–7306.

Andrews, S., Huerta, I., Komura, T., Sigal, L., and Mitchell, K. (2016). Real-time physics-based motion capture with sparse sensors. In *Proceedings of 13th European conference on visual media production (CVMP 2016)*, pages 1–10.

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

Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., and Davis, J. (2005). Scape: shape completion and animation of people. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2005*, 24(3).

Aristidou, A. and Lasenby, J. (2011). Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 73(5):243–260.

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

Bapat, A., Dunn, E., and Frahm, J.-M. (2016). Towards kilo-hertz 6-dof visual tracking using an egocentric cluster of rolling shutter cameras. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2358–2367.

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

Bulat, A. and Tzimiropoulos, G. (2017). How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1021–1030.

Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE Journal of Robotics and Automation*, 17(1-19):16.

Butepage, J., Black, M. J., Kragic, D., and Kjellstrom, H. (2017). Deep representation learning for human motion prediction and classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6158–6166.

Cao, C., Hou, Q., and Zhou, K. (2014a). Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Transactions on graphics (TOG)*, 33(4):43.

Cao, C., Weng, Y., Zhou, S., Tong, Y., and Zhou, K. (2014b). Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425.

Cao, X., Wei, Y., Wen, F., and Sun, J. (2014c). Face alignment by explicit shape regression. *International Journal of Computer Vision (IJCV)*, 107(2):177–190.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Cha, Y., Price, T., Wei, Z., Lu, X., Rewkowski, N., Chabra, R., Qin, Z., Kim, H., Su, Z., Liu, Y., Ilie, A., State, A., Xu, Z., Frahm, J.-M., and Fuchs, H. (2018). Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. *IEEE transactions on visualization and computer graphics (TVCG), In Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR) 2018, Munich, Germany, October*, 24(11):2993–3004.

Cha, Y., Shaik, H., Zhang, Q., Feng, F., State, A., Ilie, A., and Fuchs, H. (2021). Mobile, egocentric human body motion reconstruction using only eyeglasses-mounted cameras and a few body-worn inertial sensors. In *Proceedings of IEEE Virtual Reality (VR) 2021, Virtual, March, a Best conference paper award*, pages 607–616. IEEE.

Cha, Y.-W., Dou, M., Chabra, R., Menozzi, F., State, A., Wallen, E., and Fuchs, H. (2016). Immersive learning experiences for surgical procedures. *Studies in health technology and informatics, In Proceedings of Medicine Meets Virtual Reality / NextMed (MMVR), Los Angeles, USA, April*, 220:55–62.

Chabra, R., Ilie, A., Rewkowski, N., Cha, Y.-W., and Fuchs, H. (2017). Optimizing placement of commodity depth cameras for known 3d dynamic scene capture. In *Proceedings of IEEE Virtual Reality (VR) 2017, Los Angeles, USA, March*, pages 157–166. IEEE.

Chan, L., Hsieh, C.-H., Chen, Y.-L., Yang, S., Huang, D.-Y., Liang, R.-H., and Chen, B.-Y. (2015). Cyclops: Wearable and single-piece full-body gesture input devices. In *Proceedings of 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3001–3009.

Chen, X., Guo, Y., Zhou, B., and Zhao, Q. (2013). Deformable model for estimating clothed and naked human shapes from a single image. *The Visual Computer*, 29(11):1187–1196.

Cheng, Y., Yang, B., Wang, B., Yan, W., and Tan, R. T. (2019). Occlusion-aware networks for 3d human pose estimation in video. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 723–732.

Coumans, E. and Bai, Y. (2016–2021). Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`. Accessed: 2021-06-01.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proceedings of 23rd annual conference on Computer graphics and interactive techniques (SIGGRAPH)*, pages 303–312. ACM.

Dabral, R., Mundhada, A., Kusupati, U., Afaque, S., Sharma, A., and Jain, A. (2018). Learning 3d human pose from structure and motion. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 668–683.

De Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H.-P., and Thrun, S. (2008). Performance capture from sparse multi-view video. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2008*, 27(3):1–10.

De Aguiar, E., Theobalt, C., Stoll, C., and Seidel, H.-P. (2007). Marker-less deformable mesh tracking for human shape and motion capture. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Desselle, M. R., Brown, R. A., James, A. R., Midwinter, M. J., Powell, S. K., and Woodruff, M. A. (2020). Augmented and virtual reality in surgery. *Computing in Science & Engineering*, 22(3):18–26.

Djelouah, A., Franco, J.-S., Boyer, E., Le Clerc, F., and Perez, P. (2013). Multi-view object segmentation in space and time. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2640–2647. IEEE.

Dou, M. and Fuchs, H. (2014). Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In *Proceedings of IEEE Virtual Reality (VR) 2014, Best short paper award*, pages 39–44. IEEE.

Dou, M., Guan, L., Frahm, J.-M., and Fuchs, H. (2012). Exploring high-level plane primitives for indoor 3d reconstruction with a hand-held rgb-d camera. In *Proceedings of Asian Conference on Computer Vision (ACCV) Workshops*, pages 94–108. Springer.

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., and Izadi, S. (2016). Fusion4d: Real-time performance capture of challenging scenes. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2016*, 35(4):114.

Dou, M., Taylor, J., Fuchs, H., Fitzgibbon, A., and Izadi, S. (2015). 3d scanning deformable objects with a single rgbd sensor. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 493–501.

Drexler, D. A. and Harmati, I. (2012). Joint constrained differential inverse kinematics algorithm for serial manipulators. *Periodica Polytechnica. Electrical Engineering and Computer Science*, 56(4):95.

Ebert, L. C., Nguyen, T. T., Breitbeck, R., Braun, M., Thali, M. J., and Ross, S. (2014). The forensic holodeck: an immersive display for forensic crime scene reconstructions. *Forensic science, medicine, and pathology*, 10(4):623–626.

EgoVIP Dataset (2021). Egocentric Visual+Inertial Human Pose Dataset. `https://sites.google.com/site/youngwooncha/egovip`. Accessed: 2021-06-01.

Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(3):611–625.

Engel, J., Schöps, T., and Cremers, D. (2014). Lsd-slam: Large-scale direct monocular slam. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 834–849. Springer.

Facebook Reality Labs Project Aria (2020). Facebook Reality Labs Project Aria Smart AR Glasses. `https://about.facebook.com/realitylabs/projectaria/`. Accessed: 2021-06-01.

Featherstone, R. (2014). *Rigid body dynamics algorithms*. Springer.

Felis, M. L. (2017). Rbdl: an efficient rigid-body dynamics library using recursive algorithms. *Autonomous Robots*, 41(2):495–511.

Ferracani, A., Pezzatini, D., and Del Bimbo, A. (2014). A natural and immersive virtual interface for the surgical safety checklist training. In *Proceedings of ACM International Workshop on Serious Games*, pages 27–32. ACM.

Fuchs, H., State, A., and Bazin, J.-C. (2014). Immersive 3d telepresence. *IEEE Computer*, 47(7):46–52.

Gall, J., Stoll, C., De Aguiar, E., Theobalt, C., Rosenhahn, B., and Seidel, H.-P. (2009). Motion capture using joint skeleton tracking and surface estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1753. IEEE.

Gong, K., Liang, X., Li, Y., Chen, Y., Yang, M., and Lin, L. (2018). Instance-level human parsing via part grouping network. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 770–785.

Google Jump (2015). Google Jump VR. `https://xinreality.com/wiki/Jump`. Accessed: 2021-06-01.

Habermann, M., Xu, W., Zollhoefer, M., Pons-Moll, G., and Theobalt, C. (2019). Livecap: Real-time human performance capture from monocular video. *ACM Transactions on Graphics (TOG)*, 38(2):1–17.

Habermann, M., Xu, W., Zollhofer, M., Pons-Moll, G., and Theobalt, C. (2020). Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5052–5063.

Hassan, M., Choutas, V., Tzionas, D., and Black, M. J. (2019). Resolving 3d human pose ambiguities with 3d scene constraints. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2282–2292.

Hassan, M., Ghosh, P., Tesch, J., Tzionas, D., and Black, M. J. (2021). Populating 3d scenes by learning human-scene interaction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Heinly, J., Schonberger, J. L., Dunn, E., and Frahm, J.-M. (2015). Reconstructing the world* in six days*(as captured by the yahoo 100 million image dataset). In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3287–3295.

Hirshberg, D. A., Loper, M., Rachlin, E., and Black, M. J. (2012). Coregistration: Simultaneous alignment and modeling of articulated 3d shape. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 242–255. Springer.

Huang, Y., Kaufmann, M., Aksan, E., Black, M. J., Hilliges, O., and Pons-Moll, G. (2018). Deep inertial poser: Learning to reconstruct human pose from sparse inertial measurements in real time. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2018*, 37:185:1–185:15.

Ikemoto, L., Arikan, O., and Forsyth, D. (2006). Knowing when to put your foot down. In *Proceedings of 2006 symposium on Interactive 3D graphics and games*, pages 49–53.

ItSeez3D (2014). ItSeez3D Scanning App. `https://itseez3d.com/`. Accessed: 2021-06-01.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., and Davison, A. (2011). Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of 22nd ACM international conference on Multimedia*, pages 675–678. ACM.

Jiang, H. and Grauman, K. (2017). Seeing invisible poses: Estimating 3d body pose from egocentric video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3509. IEEE.

Karras, T., Aila, T., Laine, S., Herva, A., and Lehtinen, J. (2017). Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):94.

Kazhdan, M. and Hoppe, H. (2013). Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):29.

Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I., and Olivier, P. (2012). Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of 25th annual ACM symposium on User interface software and technology*, pages 167–176. ACM.

Kocabas, M., Athanasiou, N., and Black, M. J. (2020). Vibe: Video inference for human body pose and shape estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5253–5263.

Kovar, L., Schreiner, J., and Gleicher, M. (2002). Footskate cleanup for motion capture editing. In *Proceedings of 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 97–104.

Kurillo, G., Bajcsy, R., Kreylos, O., and Rodriguez, R. (2009). Teleimmersive environment for remote medical collaboration. *Studies in health technology and informatics, In Proceedings of Medicine Meets Virtual Reality / NextMed (MMVR)*, 142:148–150.

Lenovo ThinkReality A3 (2021). Lenovo ThinkReality A3 Smart AR Glasses. `https://www.lenovo.com/us/en/thinkrealitya3`. Accessed: 2021-06-01.

Li, H., Vouga, E., Gudym, A., Luo, L., Barron, J. T., and Gusev, G. (2013). 3d self-portraits. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2013*, 32(6).

Liao, M., Zhang, Q., Wang, H., Yang, R., and Gong, M. (2009). Modeling deformable objects from a single depth camera. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 167–174. IEEE.

Lin, Q., Xu, Z., Li, B., Baucom, R., Poulose, B., Landman, B. A., and Bodenheimer, R. E. (2013). Immersive virtual reality for visualization of abdominal ct. In *Proceedings of SPIE Medical Imaging*, volume 8673, page 17. International Society for Optics and Photonics.

Liu, C., Kim, K., Gu, J., Furukawa, Y., and Kautz, J. (2019). Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4450–4459.

Liu, L., Yin, K., van de Panne, M., Shao, T., and Xu, W. (2010). Sampling-based contact-rich motion control. *ACM Transctions on Graphics (TOG), In Proceedings of SIGGRAPH 2010*, 29(4):Article 128.

Liu, Y. (2014). Virtual neurosurgical education for image-guided deep brain stimulation neurosurgery. In *Proceedings of IEEE International Conference on Audio, Language and Image Processing (ICALIP)*, pages 623–626.

Liu, Y., Xu, F., Chai, J., Tong, X., Wang, L., and Huo, Q. (2015). Video-audio driven real-time facial animation. *ACM Transactions on Graphics (TOG)*, 34(6):182.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., and Black, M. J. (2015). SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia*, 34(6):248:1–248:16.

Lorensen, W. E. and Cline, H. E. (1987). Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169.

Magic Leap 1 (2018). Magic Leap 1. `https://www.magicleap.com/en-us/magic-leap-1`. Accessed: 2021-06-01.

Maimone, A. and Fuchs, H. (2011). Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Proceedings of 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 137–146. IEEE.

Maimone, A. and Fuchs, H. (2012). Reducing interference between multiple structured light depth sensors using motion. In *Proceedings of IEEE Virtual Reality (VR) 2012, Best short paper award*, pages 51–54. IEEE.

Malleson, C., Gilbert, A., Trumble, M., Collomosse, J., Hilton, A., and Volino, M. (2017). Real-time full-body motion capture from video and imus. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 449–457. IEEE.

Martinez, J., Black, M. J., and Romero, J. (2017). On human motion prediction using recurrent neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2891–2900.

Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., and Theobalt, C. (2017a). Monocular 3d human pose estimation in the wild using improved cnn supervision. In *Proceedings of international conference on 3D vision (3DV)*, pages 506–516. IEEE.

Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., Seidel, H.-P., Rhodin, H., Pons-Moll, G., and Theobalt, C. (2020). Xnect: Real-time multi-person 3d motion capture with a single rgb camera. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2020*, 39(4):82–1.

Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., and Theobalt, C. (2018). Single-shot multi-person 3d pose estimation from monocular rgb. In *Proceedings of International Conference on 3D Vision (3DV)*, pages 120–130. IEEE.

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

Microsoft HoloLens 1 (2016). Microsoft Hololens 1. `https://docs.microsoft.com/en-us/hololens/hololens1-hardware`. Accessed: 2021-06-01.

Microsoft Hololens 2 (2019). Microsoft Hololens 2. `https://www.insight.com/en_US/shop/partner/microsoft/hardware/hololens.html`. Accessed: 2021-06-01.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163.

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 343–352.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., Kohi, P., Shotton, J., Hodges, S., and Fitzgibbon, A. (2011). Kinectfusion: Real-time dense surface mapping and tracking. In *Proceedings of 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136. IEEE.

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 483–499. Springer.

Nibali, A., He, Z., Morgan, S., and Prendergast, L. (2018). Numerical coordinate regression with convolutional neural networks. *arXiv preprint arXiv:1801.07372*.

Olszewski, K., Lim, J. J., Saito, S., and Li, H. (2016). High-fidelity facial and speech animation for vr hmds. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2016*, 35(6).

Orts-Escolano, S., Rhemann, C., Fanello, S., Chang, W., Kowdle, A., Degtyarev, Y., Kim, D., Davidson, P. L., Khamis, S., and Dou, M. (2016). Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM.

Osman, A. A. A., Bolkart, T., and Black, M. J. (2020). Star: Sparse trained articulated human body regressor. In *Proceedings of European Conference on Computer Vision (ECCV)*, volume LNCS 12355, pages 598–613.

Parvati, D., Heinrichs, W. L., and Patricia, Y. (2011). Clinispace: A multiperson 3d online immersive training environment accessible through a browser. *Studies in health technology and informatics, In Proceedings of Medicine Meets Virtual Reality / NextMed (MMVR)*, 163:173.

Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A. A. A., Tzionas, D., and Black, M. J. (2019). Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Pishchulin, L., Wuhrer, S., Helten, T., Theobalt, C., and Schiele, B. (2017). Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 67:276–286.

Ramakrishna, V., Munoz, D., Hebert, M., Bagnell, J. A., and Sheikh, Y. (2014). Pose machines: Articulated pose estimation via inference machines. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 33–47. Springer.

Rempe, D., Guibas, L. J., Hertzmann, A., Russell, B., Villegas, R., and Yang, J. (2020). Contact and human dynamics from monocular video. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 71–87. Springer.

Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.-P., Schiele, B., and Theobalt, C. (2016). Egocap: egocentric marker-less motion capture with two fisheye cameras. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2016*, 35(6):162.

Romero, J., Tzionas, D., and Black, M. J. (2017). Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia*, 36(6).

Rose, A. S., Kim, H., Fuchs, H., and Frahm, J.-M. (2019). Development of augmented-reality applications in otolaryngology–head and neck surgery. *The Laryngoscope*, 129:S1–S11.

Samsung Glasses Lite Concept (2021). Samsung Glasses Lite Concept Smart AR Glasses. `https://www.entrepreneur.com/article/366097`. Accessed: 2021-06-01.

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision (IJCV)*, 47(1-3):7–42.

Schonberger, J. L. and Frahm, J.-M. (2016). Structure-from-motion revisited. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113.

Schönberger, J. L., Zheng, E., Frahm, J.-M., and Pollefeys, M. (2016). Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 501–518. Springer.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 519–528. IEEE.

Shah, M., Eastman, R. D., and Hong, T. (2012). An overview of robot-sensor calibration methods for evaluation of perception systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, pages 15–20. ACM.

Shimada, S., Golyanik, V., Xu, W., and Theobalt, C. (2020). Physcap: Physically plausible monocular 3d motion capture in real time. *ACM Transactions on Graphics (TOG)*, 39(6):1–16.

Shiratori, T., Park, H. S., Sigal, L., Sheikh, Y., and Hodgins, J. K. (2011). Motion capture from body-mounted cameras. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2011*, 30(4).

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations (ICLR) 2015*.

Snavely, N., Seitz, S. M., and Szeliski, R. (2006). Photo tourism: exploring photo collections in 3d. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2006*, 25(3):835–846.

Starck, J. and Hilton, A. (2007). Surface capture for performance-based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31.

Sumikura, S., Shibuya, M., and Sakurada, K. (2019). Openvslam: a versatile visual slam framework. In *Proceedings of 27th ACM International Conference on Multimedia*, pages 2292–2295.

Sumner, R. W. and Popović, J. (2004). Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)*, 23(3):399–405.

Sun, X., Xiao, B., Wei, F., Liang, S., and Wei, Y. (2018). Integral human pose regression. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 529–545.

Taneja, A., Ballan, L., and Pollefeys, M. (2011). Image based detection of geometric changes in urban environments. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2336–2343. IEEE.

Tateno, K., Tombari, F., Laina, I., and Navab, N. (2017). Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tautges, J., Zinke, A., Krüger, B., Baumann, J., Weber, A., Helten, T., Müller, M., Seidel, H.-P., and Eberhardt, B. (2011). Motion reconstruction using sparse accelerometer data. *ACM Transactions on Graphics (ToG), In Proceedings of SIGGRAPH 2011*, 30(3):18.

Tome, D., Peluse, P., Agapito, L., and Badino, H. (2019). xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 7728–7738.

Tong, J., Zhou, J., Liu, L., Pan, Z., and Yan, H. (2012). Scanning 3d full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics (TVCG), In Proceedings of IEEE VR 2012*, 18(4).

Trumble, M., Gilbert, A., Malleson, C., Hilton, A., and Collomosse, J. P. (2017). Total capture: 3d human pose estimation fusing video and inertial sensors. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 14.1–14.13.

Ulusoy, A. O. and Mundy, J. L. (2014). Image-based 4-d reconstruction using 3-d change detection. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 31–45. Springer.

Unity (2005). Unity Game Engine. `https://unity.com/`. Accessed: 2021-06-01.

Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., and Schmid, C. (2018). Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 20–36.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 5998–6008.

Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. (2017). Learning to generate long-term future via hierarchical prediction. In *Proceedings of 34th International Conference on Machine Learning (ICML)*, pages 3560–3569. JMLR. org.

Vlasic, D., Baran, I., Matusik, W., and Popović, J. (2008). Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2008*, 27(3).

von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., and Pons-Moll, G. (2018). Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 601–617.

Von Marcard, T., Pons-Moll, G., and Rosenhahn, B. (2016). Human pose estimation from video and imus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(8):1533–1547.

von Marcard, T., Rosenhahn, B., Black, M., and Pons-Moll, G. (2017). Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. *Computer Graphics Forum 36(2), In Proceedings of 38th Annual Conference of the European Association for Computer Graphics (Eurographics)*, pages 349–360.

Wang, C., Miguel Buenaposada, J., Zhu, R., and Lucey, S. (2018). Learning depth from monocular videos using direct methods. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2022–2030.

Wang, R., Schwörer, M., and Cremers, D. (2017). Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

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

Weiss, A., Hirshberg, D. A., and Black, M. J. (2011). Home 3d body scans from noisy image and range data. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Welch, G., Russo, D., Funaro, J., van Dam, A., Ilie, A., Low, K.-L., Lastra, A., Cairns, B., Towles, H., and Fuchs, H. (2005). Immersive electronic books for surgical training. *IEEE MultiMedia*, 12(3):22–35.

Wu, C., Stoll, C., Valgaerts, L., and Theobalt, C. (2013). On-set performance capture of multiple actors with a stereo camera. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2013*, 32(6):161.

Xiang, D., Joo, H., and Sheikh, Y. (2019). Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10974.

Xiong, X. and De la Torre, F. (2013). Supervised descent method and its applications to face alignment. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539.

Xsens Mtw Awinda (2015). Xsens Mtw Awinda Inertial Sensors. `https://www.xsens.com/products/mtw-awinda`. Accessed: 2021-06-01.

Xu, W., Chatterjee, A., Zollhoefer, M., Rhodin, H., Mehta, D., Seidel, H.-P., and Theobalt, C. (2018). Monoperfcap: Human performance capture from monocular video. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH 2018*, 37(2):27.

Xu, W., Chatterjee, A., Zollhofer, M., Rhodin, H., Fua, P., Seidel, H.-P., and Theobalt, C. (2019). Mo2cap2: Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE transactions on visualization and computer graphics (TVCG), In Proceedings of IEEE VR*.

Ye, G., Liu, Y., Hasler, N., Ji, X., Dai, Q., and Theobalt, C. (2012). Performance capture of interacting characters with handheld kinects. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 828–841. Springer.

Yu, T., Zheng, Z., Zhong, Y., Zhao, J., Dai, Q., Pons-Moll, G., and Liu, Y. (2019). Simulcap: Single-view human performance capture with cloth simulation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5499–5509. IEEE.

Zeng, M., Zheng, J., Cheng, X., and Liu, X. (2013). Templateless quasi-rigid shape modeling with implicit loop-closure. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 145–152. IEEE.

Zhang, F., Zhu, X., and Ye, M. (2019). Fast human pose estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3517–3526.

Zhang, P., Siu, K., Zhang, J., Liu, C. K., and Chai, J. (2014). Leveraging depth cameras and wearable pressure sensors for full-body kinematics and dynamics capture. *ACM Transactions on Graphics (TOG), In Proceedings of SIGGRAPH Asia 2014*, 33(6):221.

Zhang, S., Zhang, Y., Ma, Q., Black, M. J., and Tang, S. (2020). Place: Proximity learning of articulation and contact in 3d environments. In *Proceedings of 8th international conference on 3D Vision (3DV)*.

Zhu, X., Lei, Z., Liu, X., Shi, H., and Li, S. Z. (2016). Face alignment across large poses: A 3d solution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155.

Zollhöfer, M., Nießner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., and Theobalt, C. (2014). Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (TOG)*, 4.

Zou, Y., Yang, J., Ceylan, D., Zhang, J., Perazzi, F., and Huang, J.-B. (2020). Reducing footskate in human motion reconstruction with ground contact constraints. In *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 459–468.