論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Markerless Human Motion Capture and Visualization from Monocular Videos
著者(和文)	HWANG Dong-Hyun
Author(English)	Dong-Hyun Hwang
出典(和文)	学位:博士(学術), 学位授与機関:東京工業大学, 報告番号:甲第11848号, 授与年月日:2022年3月26日, 学位の種別:課程博士, 審査員:小池 英樹,德永 健伸,三宅 美博,岡崎 直観,齋藤 豪,佐藤 洋一
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第11848号, Conferred date:2022/3/26, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Dissertation

Markerless Human Motion Capture and Visualization from Monocular Videos

by

Dong-Hyun Hwang

Graduate Major in Computer Science
School of Computing
Tokyo Institute of Technology

Supervisor: Hideki Koike

January, 2022

Acknowledgments

I would first like to thank my supervisor, Prof. Hideki Koike, for guiding my research and supporting my life abroad, both physically and mentally. There is no doubt that he is a great teacher and a true mentor. I also greatly appreciate my thesis committee: Yoshihiro Miyake, Takenobu Tokunaga, Naoaki Okazaki, Suguru Saito, and Yoichi Sato, for their beneficial feedback and advice.

I would like to express my deep gratitude to Soonmin Bae of KT AI2XL, Suntae Kim, Nicolas Monet, and other colleagues of NAVER CLOVA. We have had continuous industrial exchanges during my doctoral program. It is fortunate for me to be able to work with them. Through the internship, I could learn about project-based teamwork and the spirit of coworking. I have also been fortunate to collaborate with Prof. Kris Kitani, Ye Yuan, and other students during my visiting scholar program at Robotics Institute, Carnegie Mellon University. They taught me to polish research through critical thinking and active discussion. Thanks to my research group members: Atsuki Ikeda, Kohei Aso, Jacky Liao, Haruki Kikuchi, and others who have worked tirelessly to make wonderful discoveries with me. Working with them, I learned how to manage and lead projects. I want to express my gratitude for the financial support of the Asahi Glass Scholarship Foundation and the Japan Society for the Promotion of Science.

Finally, I dedicate this thesis to my family. Their tireless love and support are the fuel of my scientific exploration.

Abstract

Human motion capture is the process of recording body movements, emotional expressions, and interactions. The captured motion could be utilized as a computer graphics-based animation, healthcare, and an intuitive input interface for Augmented/Virtual Reality (AR/VR). Recent advances in deep neural network-based computer vision algorithms allow realizing markerless motion capture, which can capture the user's motion without an additional suit with markers or sensors. This technology makes it possible to record the motions of people with various appearances in the real world.

However, many cameras are required to record complex human motions with minimal occlusion, and advanced hardware configurations are necessary to synchronize the capturing timing of these cameras. In addition, a 3D model is required to create content using the captured motion. In order to create this model, polygons and textures should be created either manually by experts or through a 3D scanning system using dozens of cameras. Because of the high complexity of existing systems, the cost required to build the system is very high, and only users with expert knowledge can create content using motion capture technology. We define this phenomenon as the "Curse of Cameras". This "Curse of Cameras" allows end-users limited accessibility to motion capture and content creation systems. Recent monocular video-based motion capture and content creation methods using deep neural networks have low generalization capacity. Therefore, model fragmentation requires preparing several machine learning models to cover various cases. Due to these problems, it is challenging to create content using motion capture by end-users, and the low content productivity caused by this could impede the rapid arrival of the XR era.

In this thesis, to solve the "Curse of Cameras", we propose methods for motion capture and a method to create and visualize AR content by retargeting the captured motion from monocular videos. First, we present a multimodal motion-capture system that estimates the various human-motion information using an ultra-wide fisheye camera. Then, we propose a lightweight 3D human motion capture network and an efficient training strategy for devices with limited computation power used in portable motion capture systems. Finally, we introduce a universal system that estimates motion information, creates a 2D texture that could be retargeted, and generates pseudo-2.5D AR content based on this information.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The body says what words cannot.
- Martha Graham (1894-1991)

Human motion means geometric movement in time and space, and our musculoskeletal system, which supports body weights and movement of body segments, makes this possible. Human motion capture is a process of recording human motion as analog or digital signals. The captured motion can be utilized not only in entertainment such as movies and games but also in various fields such as natural input interface for augmented/virtual reality (AR/VR), sports science, healthcare, and social interaction analysis.

1.1 Historical Review

Many efforts to record human movements have been ongoing for a long time. Leonardo da Vinci's (1452-1519) sketchbook contains drawings describing a man going upstairs and up a ladder (See Figure 1.1). In addition, a detailed analysis of the change in the body's center of gravity and weight caused by human movements is described. Although this was written centuries ago, the level of detail in human motion description is impressive, and this has become the basis for modern motion capture and analysis studies.

After the invention of the camera, there were attempts to record the motion of people or animals using the camera. Etienne-Jules Marey (1830-1904) photographed motion information by chronophotography of a runner wearing a black uniform with only the limbs painted by white lines as shown in Figure 1.2. He also invented an early wearable device to record the length and phase of ground contact [92].

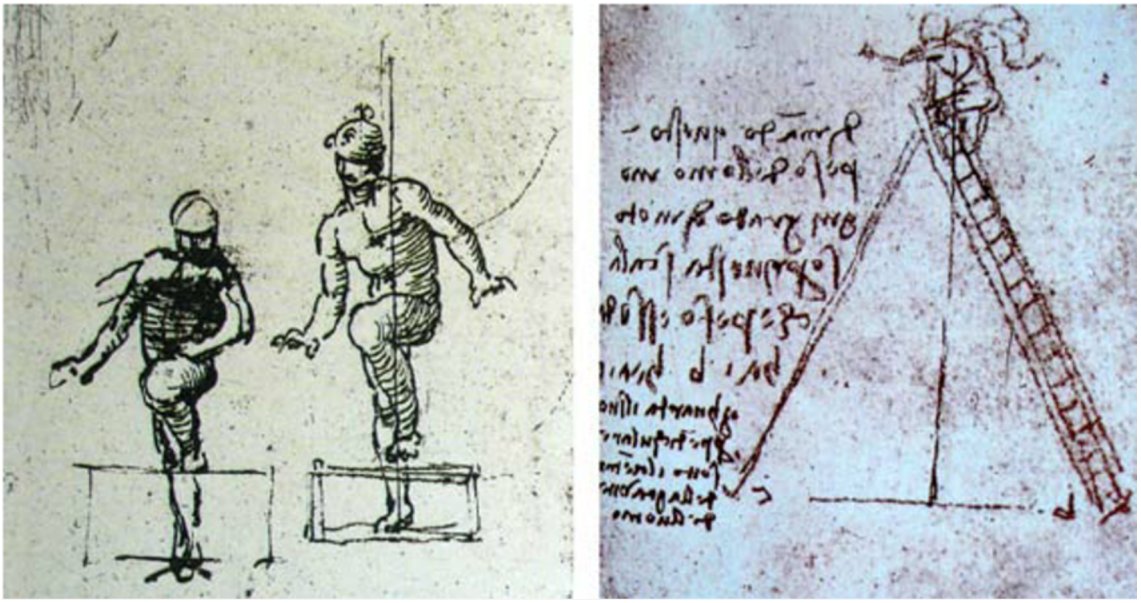


Figure 1.1: Drawings in Leonardo da Vinci's sketchbook.

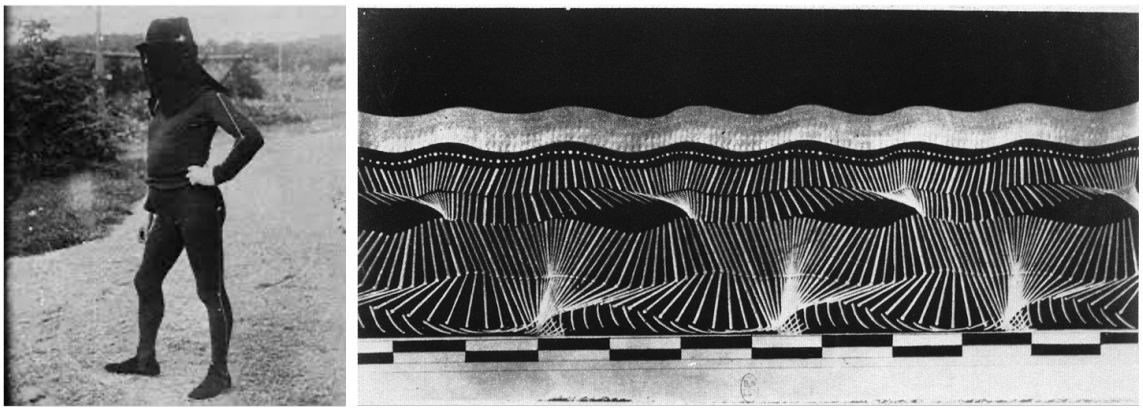


Figure 1.2: Left: a man with a black uniform. Right: captured limbs motion in a chronophotograph [92].

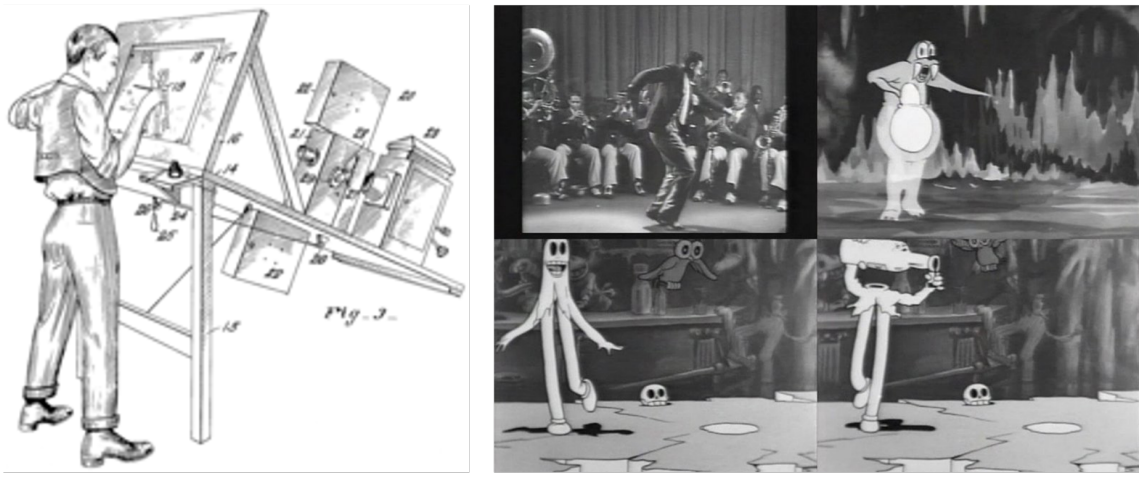


Figure 1.3: Left: patent drawing of Max Fleischer's rotoscope [48]. Right: animation frames created with rotoscoping.

From the 1900s, it became possible to record videos beyond still images, allowing continuous recording of human movements. In 1915, Max Fleischer (1883-1972) proposed rotoscoping motion capture technique. In this method, an actor's motion is first recorded using a camera, and then an animation character is painted over on every frame. This method makes it possible to express the natural movement of an animated character, like the movement of an actual human, and many animations from the 1900s used this method as shown in Figure 1.3.

In 1959, animator Lee Harrison III (1929–1998) developed the world's first motion-capture suit. This suit measures the user's movement with potentiometers for each joint and visualizes the measured data as a rough character in real time (See Figure 1.4). After developing this method, the suit-based motion capture method has become mainstream. It has been developed into an optical motion capture system using a suit with passive markers and multiple infrared cameras and an internal motion capture system that attaches multiple inertial measurement unit (IMU) sensors to a suit.

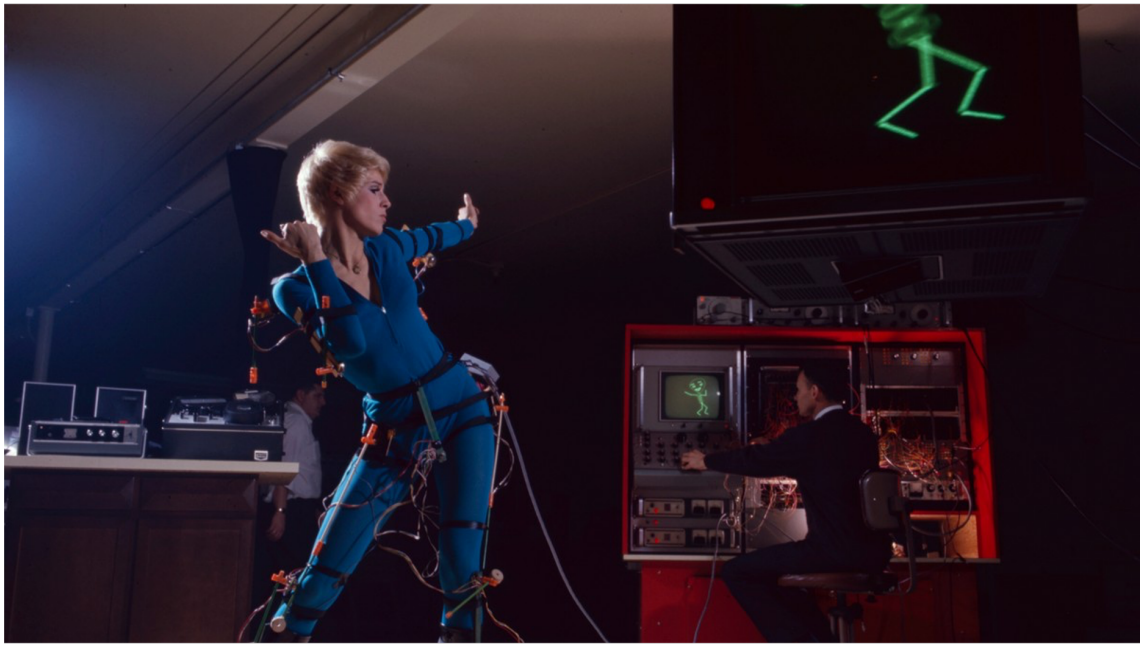


Figure 1.4: First motion capture suit developed by Lee Harrison III [128].

1.2 Optical Motion Capture

The optical motion capture system detects 2D positions of a marker set attached to the suit from images taken using two or more pre-calibrated cameras and then restores the 3D position through triangulation. This system has higher accuracy and stability even for long-term use compared to the inertial motion capture system. In addition, since this method has high scalability, extra information can be easily acquired by adding additional markers. Therefore, optical motion capture systems are mainly used in most industrial fields.

The development of computer vision and parallel computing algorithms made it possible to reconstruct the 3D shape of the human body from images taken from RGB cameras without requiring a special suit and additional devices. In the early stage, Kanade et al. proposed a virtualized reality system [75], in which 51 monocular cameras are arranged in a dome-shaped structure, a target is captured and converted into a free-viewpoint content, and the scale of the system is increased to cover a large area, such as a stadium [80]. Based on this method, EyeVision [3] was commercialized and utilized in Super Bowl 35 broadcasting.

On the other hand, with the recent advance of deep neural networks and hardware accelerators, it is possible to estimate the 2D positions of human joints from an RGB image without the motion capture suit. Accordingly, 3D joint positions can be estimated through triangulation from the 2D joint positions detected in multiple RGB images, and several markerless motion capture systems have been proposed [73, 152].

1.3 Computer Graphics(CG) Modeling

Usually, captured motion information is retargeted to CG models to create content. In order to create 3D models, the 3D modeling process generates polygons and textures in the 3D space of the computer is required.

This process is mainly performed in two ways. The first is to manually define the polygons of the model and texture map by a professional 3D modeler. Despite the recent development of 3D design software, this method requires much time, and there is difficulty in creating photorealistic models and textures.

Next is a camera array-based 3D scan method that places multiple cameras in all directions of an object. In this method, multiple cameras simultaneously capture an object, obtain a point cloud through triangulation, and then calculate a mesh from the acquired point cloud. The method can obtain meshes and textures from

real-world objects faster than the manual design method. However, this requires complex hardware and calculations and is expensive to build.

Recently, like a motion capture system, many works have been proposed to estimate the shape of the human body with clothes [8, ?, 56] and the UV texture map [7, 145] from a monocular image using a deep neural network.

1.4 Beyond The Curse of Cameras

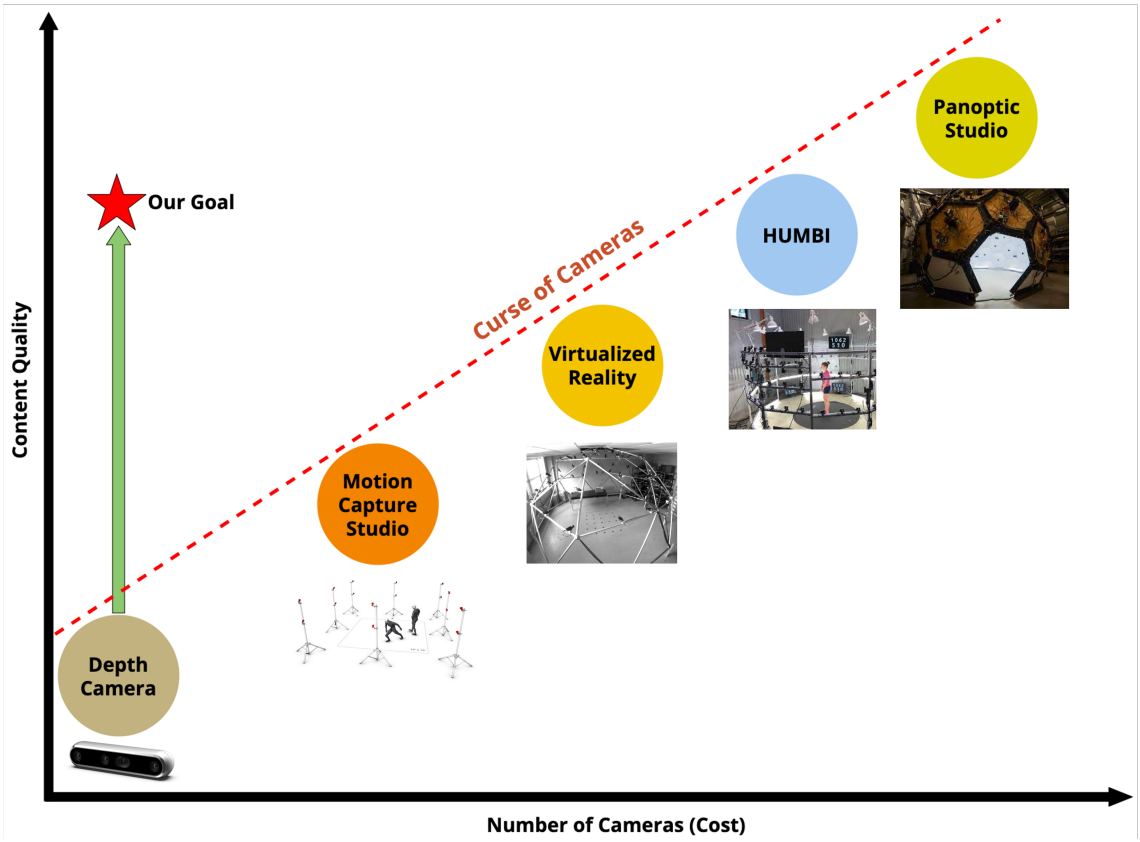


Figure 1.5: Illustration of the "Curse of Cameras" and goal of research.

Humans perceive the depth of an object based on the binocular disparity of images obtained through two eyes. Similarly, in computer vision, the depth information is calculated using epipolar geometry from images received using two cameras. Because human motion includes very diverse and complex shapes, it is hard to capture the user's motion without occlusion with two cameras' narrow-field of view. Therefore, existing motion capture and 3D scan systems have solved this problem by installing multiple cameras to capture objects from all directions.

However, several cameras are installed on fixed structures and calibrated before operation in this method. Furthermore, since precise synchronization between cameras is required, unique hardware and software are needed to handle the high

bandwidth data. Many occlusions are made in proportion to the type and complexity of the motion and the number of target users, and the number of required cameras increases exponentially to minimize these occlusions. For example, HUMBI [152], a markerless motion capture system that captures one person's motion, consists of 110 cameras, while Panoptic studio [73], which supports motion capture of multiple people, consists of 512 cameras. Therefore, the capture volume of the current motion capture system is small, and since the system requires very high cost and human resources, end-users and small groups are hard to use. We called this phenomenon as "Curse of Cameras", and Figure 1.5 shows the "Curse of Cameras".

As mentioned earlier, we need at least two images to calculate the depth of an object. Nevertheless, here is the question, "Can we perceive depth with just one eye?" The answer is "Yes." A human can estimate the depth of an object with a single eye through monocular cues such as perspective, an object's relative size, clarity of texture, and more. Therefore, many studies have been conducted to estimate 3D human motion and texture and generate content from monocular videos.

There are countless monocular cameras in our world today. This may be a typical digital camera and a smartphone held in people's hands, sometimes as a security camera installed in a building. In addition, as AR/VR technology develops, numerous AR/VR devices will be used in daily life in the near future, as shown in Figure 1.6, and a monocular camera will be built-in to acquire visual information in the real world inside these devices.

If we can link all the monocular cameras in space, we can establish a virtual motion capture studio in the real world. This concept will enable robust capture even with occlusion while capturing volume exceeding existing motion capture systems. The motion information captured in this way will be transmitted to AR/VR devices of people in the real world and become a significant component of the metaverse experience.

In preparation for the spread of AR/VR devices, securing content optimized for these devices is also essential. The recent spread of smartphones has promoted the rapid spread of 2D videos, the optimal content for this device. It also triggered the shift from companies or organizations to individuals as the central axis of creating video content. Unlike 2D video, 3D graphics require a 3D model to apply the captured motion data, and experts can only perform the traditional modeling process. Therefore, exploring how users can easily create a CG model that is retargeted with motion data affects the productivity of AR/VR content for general users. Also, it will be an important factor in accelerating the advent of the future XR era.

However, most works using monocular videos have low versatility even with the explosive computations of deep neural networks, and estimated information is often restricted to specific problems. These problems make fragmentation, such as when users use a specific device or machine learning model to create specific content, reducing the rapid production and spread of user-generated content. In addition, a neural network's extensive computation is challenging to utilize for AR/VR devices operating with limited power and computational capability. Despite advances in machine learning technology, we have not yet entirely beyond the "Curse of cameras", which will obstruct the rapid arrival of the XR era.

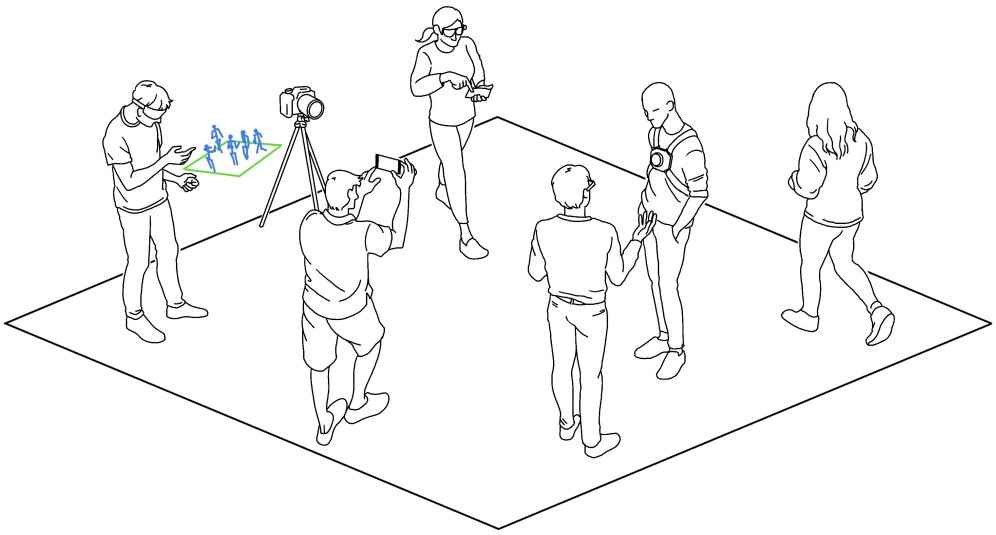
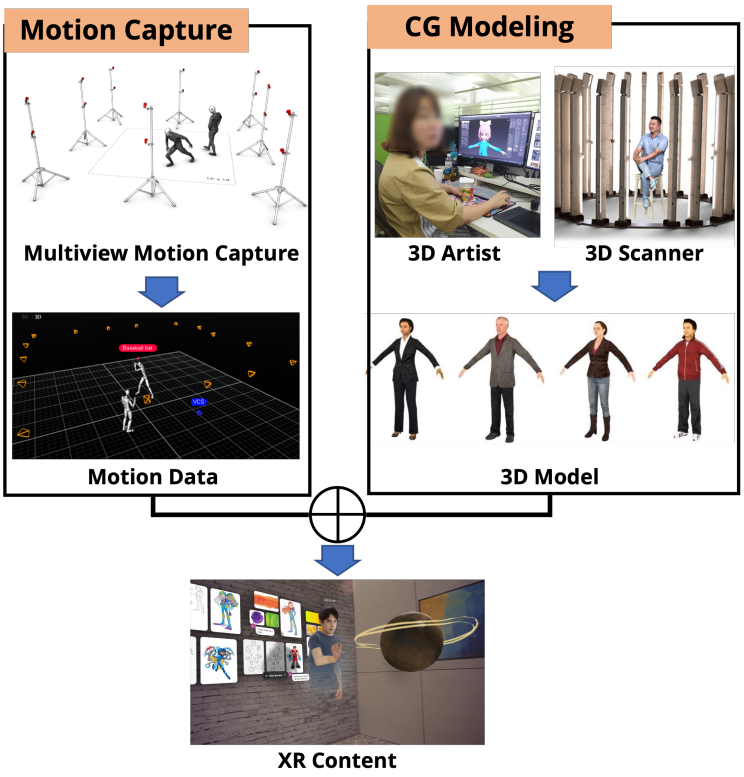
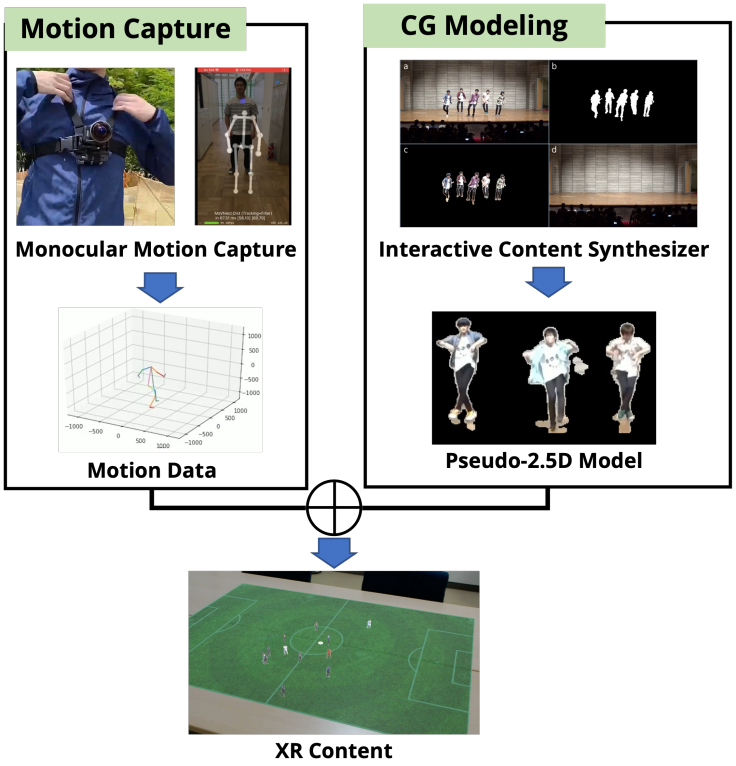


Figure 1.6: Blueprint for the near future. Visual information of the world could be captured from a variety of devices with monocular cameras.

1.5 Thesis Statement



(a) Previous content generation framework.



(b) Proposed content generation framework.

Figure 1.7: Comparison of the existing content creation framework and the proposed framework.

As shown in Figure 1.7.(a), in the existing methods for creating XR content, motion data is captured using motion capture systems. Then the captured motion is applied to models produced with a professional designer or 3D scanning system. As mentioned above, this production framework is complicated in configuration and requires expert knowledge; therefore, it is challenging for general users to create XR content.

In this thesis, we aim to create XR content with reasonable quality with monocular video as illustrated in Figure 1.5 and propose a new framework that captures human motion and creates models to produce XR content using a monocular camera, as shown in 1.7.(b). This framework, composed of a new simple hardware configuration and comprehensive system, is highly accessible to end-users and, therefore, can promote the production of user-based XR content.

In order to realize this framework concept, we present two contributions for human motion capture and one contribution that synthesizes mixed-reality content from monocular videos. Beyond the "Curse of Cameras," we address the challenges of versatility and computation efficiency for machine learning. We explore the feasibility of capturing and generating various information using a monocular camera through system examples using common types of cameras and new kinds of cameras used shortly.

First, we intend to expand the types of information that can be captured with a single wearable camera using next-generation camera hardware and machine learning technology. Our portable multimodal motion-capture system estimates the various human-motion information using an ultra-wide fisheye camera. Then, we would like to find out how to enable motion capture with portable devices with low computing power. To address the computation efficiency for mobile machine learning, we design a lightweight 3D human pose estimation network, which can run real-time on mobile devices, and propose a training strategy that maximizes the performance of the small deep neural network.

Finally, we explore a universal way to create compelling content for AR devices from pre-installed monocular cameras and pre-captured videos on the Internet. Our end-to-end, computer vision-based system generates pseudo-2.5D content based on 2D texture with motion information from various monocular videos. The generated content can be displayed with an AR headset.

1.6 Thesis Overview

1.6.1 Multimodal Human Motion Capture using A Ultra-wide Fisheye Camera (Chapter 3)

We present MonoEye, a multimodal human motion capture system using a single RGB camera with an ultra-wide fisheye lens, mounted on the user's chest. Existing optical motion capture systems use multiple cameras, which are synchronized and require camera calibration. These systems also have usability constraints that limit the user's movement and operating space. Since the MonoEye system is based on a wearable single RGB camera, the wearer's 3D body pose can be captured without space and environment limitations. The body pose, captured with our system, is aware of the camera orientation and therefore it is possible to recognize various motions that existing egocentric motion capture systems cannot recognize. Furthermore, the proposed system captures not only the wearer's body motion but also their viewport using the head pose estimation and an ultra-wide image. To implement robust multimodal motion capture, we design three deep neural networks: BodyPoseNet, HeadPoseNet, and CameraPoseNet, that estimate 3D body pose, head pose, and camera pose in real-time, respectively. We train these networks with our new extensive synthetic dataset providing 680K frames of renderings of people with a wide range of body shapes, clothing, actions, backgrounds, and lighting conditions. To demonstrate the interactive potential of the MonoEye system, we present several application examples from common body gestural to context-aware interactions.

1.6.2 Lightweight 3D Human Pose Estimation using Teacher-Student Learning (Chapter 4)

We present MoVNect, a lightweight deep neural network to capture 3D human pose using a single RGB camera. In order to improve the overall performance of the model, we apply the teacher-student learning method based knowledge distillation to 3D human pose estimation. Real-time post-processing makes the CNN output yield temporally stable 3D skeletal information, which can be used in applications directly. We implement a 3D avatar application running on mobile in real-time to demonstrate that our network achieves both high accuracy and fast inference time. Extensive evaluations show the advantages of our lightweight model with the proposed training method over previous 3D pose estimation methods on the Human3.6M dataset and mobile devices.

1.6.3 Synthesizing Pseudo-2.5D Mixed Reality Content from Monouclar Videos (Chapter 5)

We propose MonoMR, a system that synthesizes pseudo-2.5D content from monocular videos for mixed reality (MR) HMDs. Unlike conventional systems that require multiple cameras, the MonoMR system can be used by casual end-users to generate MR content from a single camera only. In order to synthesize the content, the system detects people in the video sequence via a deep neural network, and then the detected person's pseudo-3D position is estimated by our proposed novel algorithm through a homography matrix. Finally, the person's texture is extracted using a background subtraction algorithm and is placed on an estimated 3D position. The synthesized content can be played in MR HMD, and users can freely change their viewpoint and the content's position. In order to evaluate the efficiency and interactive potential of MonoMR, we conducted performance evaluations and a user study with 12 participants. Moreover, we demonstrated the feasibility and usability of the MonoMR system to generate pseudo-2.5D content using three example application scenarios.

The relevant major publication list for this thesis is following:

1. MonoEye: Multimodal Human Motion Capture System Using A Single Ultra-Wide Fisheye Camera, ACM UIST 2020. (Chapter 3)

2. Portable 3D Human Pose Estimation for Human-Human Interaction using a Chest-Mounted Fisheye Camera, Augmented Humans 2021. (Chapter 3)

3. MonoEye: Monocular Fisheye Camera-based 3D Human Pose Estimation, IEEE VR 2019. (Chapter 3)

4. Lightweight 3D Human Pose Estimation Network Training Using Teacher-Student Learning. IEEE/CVF WACV 2020. (Chapter 4)

5. MonoMR: Synthesizing Pseudo-2.5D Mixed Reality Content from Monocular Videos, Applied Sciences. (Chapter 5)

6. ParaPara: Synthesizing Pseudo-2.5d Content from Monocular Videos for Mixed Reality, ACM CHI 2018. (Chapter 5)

Chapter 2

Related Work

Our proposed system is related to motion capture system, 2D and 3D human pose estimation, monocular video-based content synthesis, and free-viewpoint video systems. In this section, we briefly discuss these related works.

2.1 Multi-view Motion Capture

Multi-view motion capture systems that have two or more cameras are usually used in a studio. Marker-based systems require the user to be equipped with an active or passive marker suit. Marker-less motion capture algorithms beyond this equipment constraint have been proposed [17, 39, 50, 63, 73, 99, 126, 127, 129, 137], and recent works have made it possible to operate outdoors with a small number of cameras [9, 22, 46, 108, 114, 115]. The multiple camera-based setups guarantee very high accuracy and capture a subject from multiple viewpoints which allow the system to be more robust to occlusion. However, capturing equipment is usually expensive. Preprocessing such as synchronization and calibration is required before capturing. In addition to this, professional operators to handle this equipment are also required.

2.2 Monocular Motion Capture from a Third Person Viewpoint

Monocular-camera-based human pose estimation is preferred for consumer-level products because of their simple configuration. Early infrared (IR)-based depth cameras have provided natural input interfaces for entertainment systems [13, 125, 140]. However, IR-based cameras have an unstable outdoor performance and the detection

range depends on the intensity of the IR light.

Human pose estimation with a single monocular RGB camera has developed rapidly with improvements in deep neural networks and large scale training datasets [10, 68, 126, 95]. In 2D human pose estimation, direct regression is provided for the numerical values of 2D joint coordinates from the image by an early-stage deep neural network [136]. The heatmap-based indirect regression method has been used in most subsequent approaches because it has high accuracy and performance [30, 34, 35, 101, 130].

In 3D Human Pose Estimation, there are two main trends: (1) pipeline approaches that consist of separate 2D joint detection and 3D lifting tasks and (2) direct 3D joint regression approaches [87, 93, 110, 133]. The pipeline approaches are easy to implement and can replace 2D detectors without training. However, the accuracy of 3D results highly depends on the 2D detection results. The direct regression methods extend the 2D heatmap to 3D by using a volumetric heatmap [90, 106, 107] or location map [98], and these methods have the advantage that the network can fully utilize the spatial information of the convolutional neural network (CNN) outputs of the layers for 3D regression. Recently, methods [69, 81] that are applied to multi-view geometry learning have been proposed and they achieve state of the art accuracy.

2.3 Egocentric 3D Human Pose Estimation

In recent years, egocentric view-based pose estimation research has been conducted in order to overcome the environment and time restrictions of existing motion capture systems. Early-stage research reconstructed full-body structure using 16 limb-mounted cameras and a structure-from-motion method [124], and head-mounted RGB-D camera-based systems [116, 151] have captured mostly upper body motions. Furthermore, recent work [26] captures not only human motion, but also the wearer‘s body, face mesh and 3D surrounding environment using multiple head-mounted cameras. Some researches [27, 5] also attempted to apply human body gestures to human-computer interaction. Chan et al. [27] use a super-wide IR camera mounted on the user's chest to classify body gestures. However, this work is not robust to outdoor environment because of the limitation of infrared light. Another work [5] uses dual hemispheres which reflect the user's body to estimate the body pose and face landmarks.

Recently, more aggressive methods have tried to estimate the 3D pose of the

camera wearer, even if no one is visible from the camera‘s point of view. In order to reach this objective, some works [70, 153, 154] use optical flow and reinforcement learning, and another work [102] utilizes optical flow, global features, and the pose of the interactee as an input of the recurrent neural network. However, the optical flow input is weak for static motion recognition.

Rhodin et al. [113] discovered the first approach for full-body motion capture with helmet-mounted stereo fisheye cameras. Portable systems [135, 144] that are capable of full-body capture with a single head-mounted camera have been proposed. However, since these methods estimate only the camera-relative joint position, a separate sensor is required to recognize various motions.

2.4 Head Pose Estimation

The typical way for head-pose estimation is a landmark-based method that detects 2D face landmarks using a 2D detector [20, 161] and estimates head pose via 3D computer vision methodologies [23, 40, 143]. However, accuracy highly depends on the detection results of the 2D detector. Recently, direct regression methods for head poses have been proposed from a single image using deep neural networks [28, 111, 118, 146], and high accuracy is achieved. The head position estimation study estimates the head pose from a complete or partially occluded face image from the third-person view. A method for a very limited part of the face image has not been pioneered yet.

Our proposed system can estimate camera pose and by using this pose feature the method can distinguish poses that are indistinguishable by the camera-relative pose estimation method of current state of the art egocentric motion capture systems. We also propose a novel method in order to estimate the head pose from a part of the face image that is captured with our ultra-wide camera and to estimate the viewport using the estimated head pose and the ultra-wide image.

2.5 Knowledge Distillation

Knowledge distillation is to transfer the information between different networks with distinct capacities [12, 18, 62]. The main idea of knowledge distillation is to apply extra supervision using the teacher model in class probabilities [62], feature representations [12, 117], or inter-layer flows [150]. It is used for efficient training of small networks difficult to train using large networks, relatively easy to train [117].

Hinton et al. successfully transferred the knowledge of a large network to a small network [62]. In addition, methods for online-based distillation [49, 156] are proposed and achieve more effective optimization than previous offline methods.

Recently, there are initial attempts to expand knowledge distillation from classification problems to human pose estimation. The initial attempt estimates human pose using radio signals [157]. In addition, a method to train an efficient lightweight 2D pose estimation model by knowledge transfer of joint heatmaps [155] is proposed and shows significant performance improvement. Although previous methods show that knowledge distillation can be applied not only to category-level discriminate knowledge but also human pose estimation [155, 157], these methods are limited to 2D human pose estimation. In this work, we propose a knowledge transfer method for 3D human pose networks using teacher-student learning. We also design MoVNect, a lightweight 3D human pose estimation network with the proposed method. Our lightweight model trained with efficient training method enables accurate pose estimation with very low computation, which can operate on devices with low processing power.

2.6 Monocular Video Based Content Synthesis

Algorithms for enhancing 3D information from monocular images and methods that utilize monocular images as Augmented Reality (AR) or MR content have been explored. The algorithm proposed by Ballan et al. [14] creates an optimal viewpoint path among two monocular videos and an interpolated video between the videos based on the optimal path; therefore, users can recognize the spatial information with a change in the viewpoint of the video. Algorithms [29, 120] have also been proposed to recover 3D information from monocular videos, construct meshes, reconstruct videos based on recovered information, and freely convert viewpoints.

Langlotz et al. proposed a smartphone-based AR system [86]. In this system, the user manually specifies the moving person of the video, and the system synthesizes the AR content by extracting the designated person portion from the video and synthesizing it on other videos. The learning support system proposed by Mohr et al. [100] extracts features from a monocular video, and the features are transformed based on the previously defined three-dimensional (3D) model and projected to the real world with the AR content.

2.7 Free-Viewpoint Video System

Since the method of capturing a real-world object using multiple cameras and converting the object into a 3D object produces a high-quality result, systems utilizing this method are mainstream in free-viewpoint video systems. Initially, Kanade et al. proposed a virtualized reality system [75], in which 51 monocular cameras are arranged in a dome-shaped structure. In this system, a target object can be converted into free-viewpoint content, and since the system has scalability, this system was used for early free-viewpoint sports broadcasting [80]. This multi-view method is applied to various free-viewpoint systems [82, 74, 54, 55, 53], and since these systems consist of RGB cameras, they are less restrictive to the environment. In addition, free-viewpoint sports broadcasting systems, such as Intel True View [2] and Canon's free-viewpoint video system [1], based on this method are actively commercialized by various companies. However, the systems based on this method require many cameras, synchronization devices, and a large amount of computation. Consequently, utilizing these systems by individuals or small groups is still challenging.

Studies on generating highly detailed free-viewpoint video using depth cameras have been conducted. The introduction of commercial depth cameras, such as RealSense [1] and Kinect [2], facilitates these studies. Accordingly, the systems [15, 85, 94, 149] that generate high-quality free-viewpoint video through depth cameras have been proposed. Collet et al. proposed a system [37] that generates free-view video with 106 RGB and depth cameras and compresses the video for real-time free-viewpoint video streaming. Based on this research, Orts-Escolano et al. presented the Holoportation [103] system that enables real-time telepresence on MR HMDs. However, operating this method outdoors is difficult because most depth cameras are not suited for natural light; thus, the capturing environment is restricted to indoor environments. Furthermore, since the systems still require many cameras and synchronization devices, the end user's accessibility is still limited.

Various methods for generating a free-viewpoint video from a monocular video have been proposed. However, it's still challenging because estimating the depth from a single camera is not easy, and the visual information captured by the monocular camera is limited. One of the early proposed systems, Tour into the picture (a system proposed by Horry et al. [65]), transforms artwork into a 3D scene using a perspective transformation based on user interaction. Recently, DNNs [45, 51] have been proposed to estimate a monocular image's depth information, which is

[1] https://software.intel.com/en-us/realsense
[2] https://developer.microsoft.com/en-us/windows/kinect

an essential basis in generating stereoscopic content from monocular videos. As one of the state-of-the-art technologies, Rematis et al. proposed a free-viewpoint soccer video system [112] by restoring the player's position and 3D mesh from a single soccer broadcasting video using multiple DNNs.

Research on how to view the generated free-viewpoint videos has been conducted. Before the development of the HMD, the user controls the viewpoint of the generated free-viewpoint videos through the primary input interface, such as a keyboard, mouse, and joystick. In order to improve the usability of these non-intuitive methods, interactive systems that control the viewpoint using markers [139] and multi-touch gestures [76] have been proposed. With the advancement of HMD technology, Inamoto proposed an early-type interactive MR system [66] that displays a free-viewpoint video in the real-world using a video see-through HMD, and the system proposed by Rematas et al. [112] can intuitively change the viewpoint in a free-viewpoint video using a MR HMD.

Chapter 3

Multimodal Human Motion Capture using A Ultra-wide Fisheye Camera

3.1 Overview

Figure 3.1: Our multimodal human motion capture system is based on a single ultra-wide fisheye camera that is worn on the user's chest. The wearable hardware configuration of MonoEye enables activity capture in everyday life.

Computer vision-based technology has advanced rapidly due to recent developments in deep learning. Particularly, human motion capture (which includes pose

estimation, first-person view estimation, gaze recognition, and more) is the most important research field of computer vision and has been utilized in various application fields (e.g., intuitive user interface, CG animation, and sports science).

On the other hand, conventional motion capture systems have restrictions in operation. Optical motion capture systems require multiple synchronized cameras that are deployed in a small space to triangulate the 3D position of a subject, and a calibration procedure is required whenever the position of the camera changes. In addition to this, the motion of the subject can be acquired only in the limited space where the cameras are installed. Inertial motion capture systems that consist of small inertial sensors with sensor fusion algorithms can overcome space limitations. However, this method has a lower accuracy when compared to the optical method and positional drift can compound over time. Furthermore, the subject must wear a suit with multiple sensors and a calibration process must be performed before capture. Moreover, a head-mounted or eyeglass type camera is additionally required to obtain the first-person view (we call it viewport.) of the subject in addition to the motion capture system. Therefore, the attachment of many sensors decreases the subject's athletic ability and it is difficult to capture the subject's best performance.

There are several attempts proposed [113, 135, 144] to estimate the posture of the wearer from a single body-worn camera and most of them use a head-mounted camera with a top-down view. However, head-mounted camera-based systems are problematic in terms of social acceptance because they are bulky and require a camera mount such as a cap, glasses, a head-mounted display (HMD), and more. Also, this external equipment makes it difficult to use during performance measurement of extreme sports.

In this paper, we present the MonoEye system that estimates the user's multimodal motion which includes 3D body pose and viewport with a single RGB ultra-wide fisheye camera that is worn on the chest. Figure 3.1 illustrates a proof-of-concept prototype hardware and multimodal motion captured by our system. Our ultra-wide fisheye lens has a 280-degree field-of-view and it can capture the user's limbs, face, and the surrounding environment. This allows multimodal motion capture which includes 3D body pose estimation, camera pose estimation, and head pose estimation with a single vision sensor and multiple deep neural networks. Our system realizes the camera pose-aware 3D human pose and viewport estimation by combining the outputs of neural networks. Furthermore, a small necklace-type super-fisheye camera has been released recently. Therefore, we believe that the chest-mounted camera has the high potential to transform into an everyday accessory such

as a tie clip, brooch, or sports gear which are all socially acceptable.

In summary, our contributions include

- We implement a proof-of-concept prototype of MonoEye, which consist of an ultra-wide fisheye camera and three deep neural networks to realize multimodal human motion capture.
- We have created a new large composite dataset that contains 680K frames of image and annotation data.
- The performance and interactive capacity of the proposed system is measured through quantitative and qualitative evaluations.
- Several application examples are presented to demonstrate the interactive potential of MonoEye.

3.2 Hardware Prototype

3.2.1 280-Degree Ultra-wide Fisheye Camera

The main hardware of MonoEye is an ultra-wide fisheye camera (Figure 3.2.a) that captures the user's body and the surrounding environment. Most egocentric camera-based human pose estimation systems use a head-mounted camera in order to get a top-down view to capture the user's entire body because general chest-mounted cameras can't capture the user's limbs. We have made prototype hardware that consists of a general action camera (GoPro Hero 7 Black) and an ultra-wide-angle lens (Entaniya M12-280) instead of using a top-down view-based head-mounted camera. Because our fisheye lens has 280 degrees of field of view (see Figure 3.2.b) the camera can capture the user's limbs and environment view as shown in Figure 3.2.c. In addition, a chest-mounted camera has the following advantages. As shown in Figure 3.2.d, the top-down view has larger joint position and distance variances ($Mean = 869mm, SD = 452mm$) than the chest camera-based view ($Mean = 599mm, SD = 349mm$). Previous works use dual network structures for the upper and lower body [144] or a complicated network [135] in order to address the large distance variance of the top-down view. In contrast, the body part distance distribution of the chest camera is close to uniform rather than the head-mounted camera. Therefore, our deep neural networks are designed with lightweight network structures. The chest-mounted camera can also capture the surrounding environment which includes the visual field of the user's eye. Therefore, our system can estimate not only the pose of the user, but also various user activity cues such as the viewport and camera orientation information by utilizing the environment view.

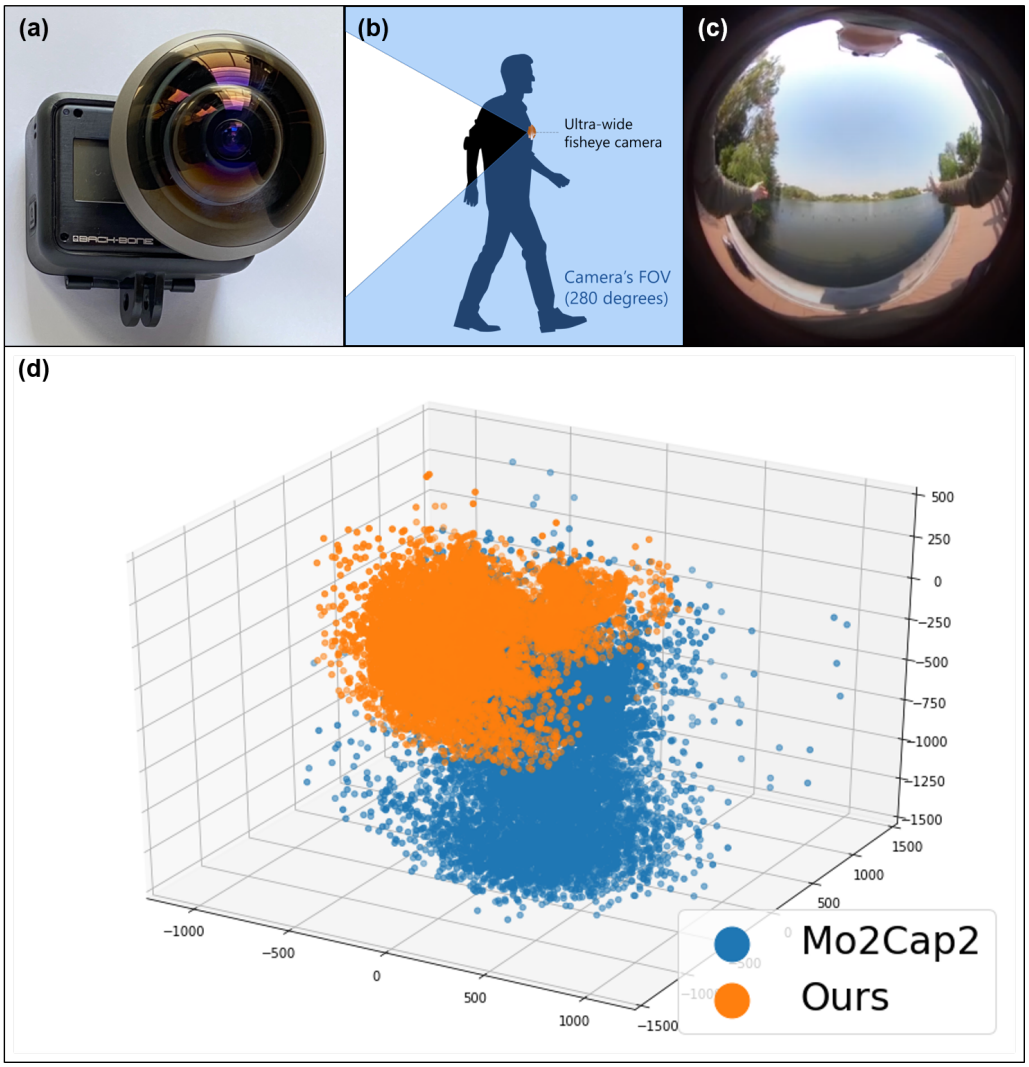


Figure 3.2: (a) Hardware prototype of MonoEye; (b) Illustration of our camera's FoV coverage, and (c) captured ultra-wide image; (d) Visualization of joint position distribution for each dataset ($N = 1000$).

3.3 Deep Neural Networks for Multimodal Motion Capture

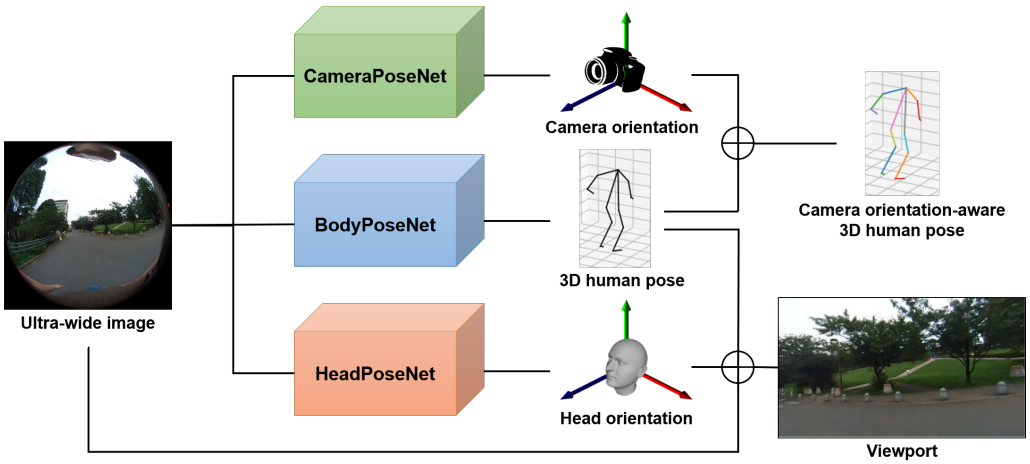


Figure 3.3: Overview of our deep neural network architectures. They consist of three networks: BodyPoseNet predicts the 2D and 3D body joint positions and HeadPoseNet and CameraposeNet estimate the rotation information of the head and camera respectively from a single RGB image. The camera orientation-aware body pose and viewport are estimated by combining the outputs of the neural networks and the ultra-wide fisheye image.

3.3.1 Overview of the Proposed Models

As illustrated in Figure 3.3, the goal of our deep neural networks is to recover the user's camera-relative 3D joint position P, head orientation R_{head}, and global camera orientation R_{camera} from a single fisheye image I. Camera pose-aware 3D human pose and viewport are calculated based on these results. We design multiple deep neural networks consisting of BodyPoseNet, HeadPoseNet, and CameraPoseNet in order to address this problem.

3.3.2 BodyPoseNet

Model Design

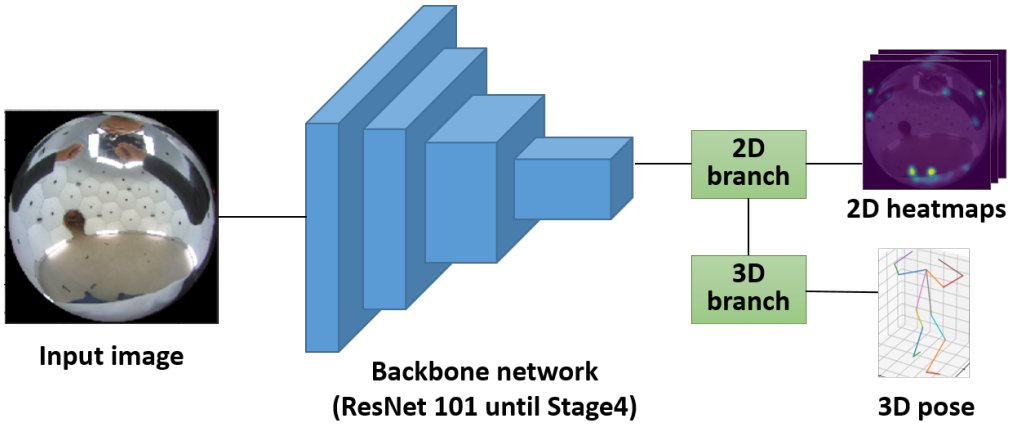


Figure 3.4: The network architecture of BodyPoseNet. BodyPoseNet estimates the 2D joint heatmaps and the camera-relative 3D human pose.

We design the network with a ResNet101 [60] backbone (until Stage 4) which extracts global features from an input image $I \in \mathbb{R}^{368\times368\times3}$ and 2D and 3D branches that estimate 2D heatmaps $H \in \mathbb{R}^{46\times46\times15}$ and 3D positions of joints set $P \in \mathbb{R}^{15\times3}$ respectively as shown in Figure 3.4. The 2D branch estimates the 2D joint positions via heatmap regression instead of direct numerical value regression which solves the complicated nonlinear problems. The 2D branch takes feature maps from the backbone part, up-samples them, and finally outputs 2D joint heatmaps by using a 1-by-1 convolution layer. The 3D branch consists of multiple bottleneck structures and outputs numerical values of the camera-relative to the 3D joint positions. The inputs of the 3D branch are intermediate feature maps of the 2D branch and it makes the network estimate full-body 3D pose from the limited visible body parts.

Loss Function

We train the BodyPoseNet by minimizing the mean squared error (MSE) between the estimated and ground-truth heatmap H and camera-relative 3D coordinates P. The loss functions L_{2D} and L_{3D} are defined as follows:

$$L_{2D} = MSE(H, \widehat{H})$$
$$L_{3D} = MSE(P, \widehat{P}),$$

where $\widehat{\ }$ indicates the ground truth.

3.3.3 HeadPoseNet

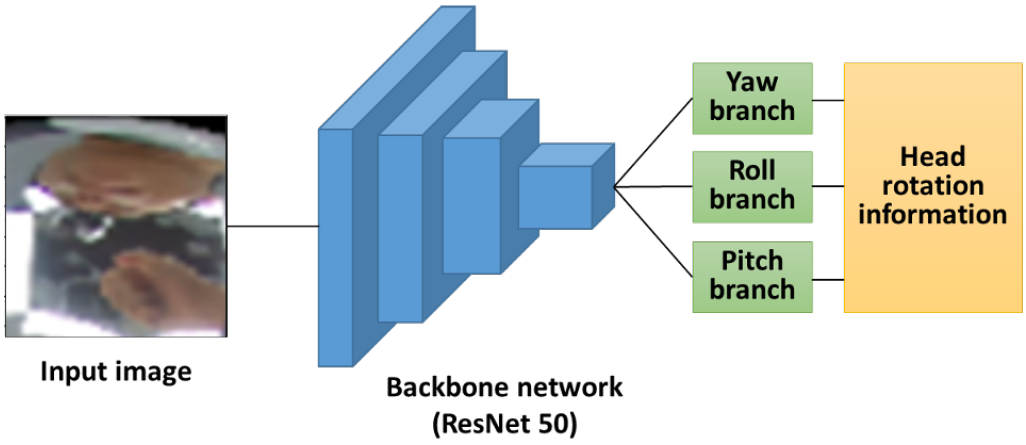


Figure 3.5: The network architecture of HeadPoseNet. HeadPoseNet estimates the head rotation information (yaw, roll, pitch) with three branches.

Model Design

We use HopeNet [118] as the network structure of HeadPoseNet. Because a captured image contains limited head shapes, it is difficult to use key-point-based head pose estimation methods [23, 40, 143]. HopeNet proved that accurate head pose estimation is possible without head key-points. The ResNet50 [60] based backbone extracts global features and each dense layer outputs binned orientation information. The main idea of this approach is to solve the regression problem easily by combining it with the classification loss. The input of the network is a cropped image of the head part $I_{crop} \in \mathbb{R}^{224\times224\times3}$ and the outputs are Euler angles of the head orientation $R_{head} \in \mathbb{R}^3$ as shown in Figure 3.5.

Loss Function

We train the HeadPoseNet by minimizing the combined distance that consists of the cross-entropy loss CE and MSE loss between the estimated and ground-truth orientation information. The loss function L_{hp} is defined as follows:

$$L_{hp} = CE(R_{head}, \widehat{R_{head}}) + \alpha\ MSE(R_{head}, \widehat{R_{head}}),$$

where $\widehat{\ }$ indicates the ground truth, and the blending factor α was identified through experimentation and set to 1e-3.

3.3.4 CameraPoseNet

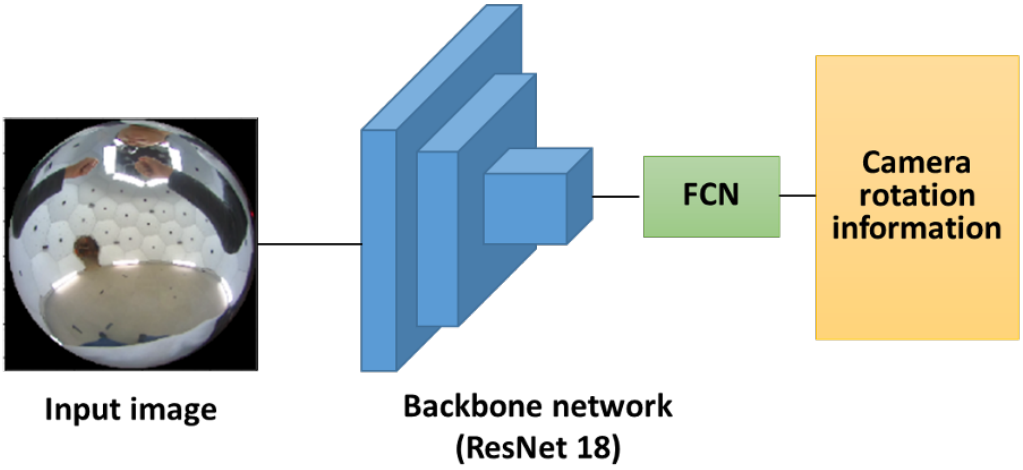


Figure 3.6: The network architecture of CameraPoseNet. CameraPoseNet estimates the camera rotation information (yaw, roll, pitch) with a single branch.

Model Design

CameraPoseNet estimates the current orientation of the camera $(Pitch, Roll)$. The network only estimates roll and pitch because there is no base point of the yaw axis in the real world. CameraPoseNet has a simple network structure that uses a ResNet18 [60] as a backbone and a dense layer. The input of the network is an ultra-wide image $I \in \mathbb{R}^{224\times224\times3}$ and the output of this network is Euler angles of the camera orientation $R_{camera} \in \mathbb{R}^2$ as shown in Figure 3.6.

Loss Function

We trained the CameraPoseNet by minimizing the MSE between the estimated and ground-truth orientation information. The loss function L_{cp} is defined as follows:

$$L_{cp} = MSE(R_{camera}, \widehat{R_{camera}}),$$

where $\widehat{\ }$ indicates the ground truth.

3.3.5 Third-person Pose Estimation Pipeline

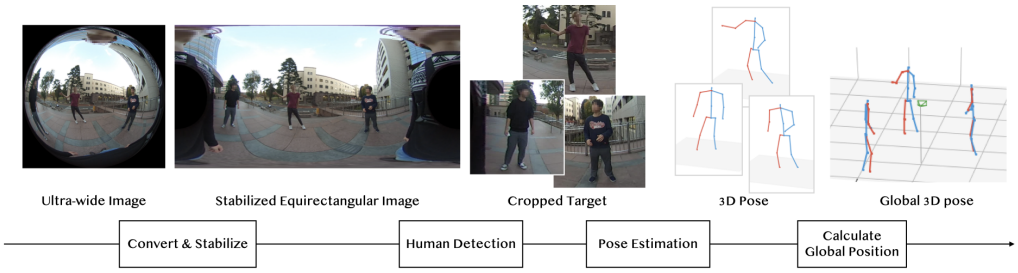


Figure 3.7: Overview of the proposed pipeline. Converting into an equirectangular image, human detection, and pose estimation are applied. Then, the targets' locations are calculated to obtain global 3D poses.

To estimate the body pose of other parties, we propose a pipeline based on a top-down approach (See Figure 3.7).

Preprocessing

A fisheye image obtained from the camera is converted to an equirectangular image. The image has the property that the latitude and longitude lines are evenly spaced. When converting, the camera lens parameters are applied in order to correct lens distortion.

Top-down Multi-person Pose Estimation

We use Mask R-CNN [59] as a human detector in an equirectangular image. Mask R-CNN consists of backbone, region proposal, and classification head networks. Our system crops the bounding boxes of humans in the image based on the inference result of Mask R-CNN, then we add a buffer area of 5% of the bounding box horizontally and vertically to ensure that the entire person is cropped. Since we use a wide-angle wearable camera, the body of the wearer is also misdetected as a person. Therefore, we use an ignore mask to remove the detection result of the wearer.

Next, We estimate the root-relative 3D poses from the cropped human images. First, we use HRNet [131] to estimate the 2D keypoints. This network consists of parallel high-to-low resolution subnetworks with repeated information exchange across the subnetworks. The input is the resized and normalized cropped image, and the output is the 2D joint locations.

Then, we lift the 2D keypoints to 3D pose using the multi-layer perceptron (MLP) model of Martinez et al. [93]. This model consists of two iterations of a building block

consisting of fully connected layers, ReLU activations, batch normalization, and skip-connections. Normalization is applied to the 2D input and 3D output by subtracting the mean and dividing by the standard deviation. The input is the normalized 2D keypoints and the output is the root-relative 3D joint locations.

3.4 Synthetic Dataset

In order to train the deep neural networks, tons of images and annotated data are required. Thus, many datasets have been created to solve the most common computer vision problems such as classification, segmentation, detection, and more. However, our proposed method has a special camera setup (See Appendix A.1 for detailed information on the fisheye camera model.), hence the networks, when trained with common datasets, won't work properly because of the huge domain gap.

There are hardware challenges to overcome when capturing training data in the real world. A marker-based motion capture system is difficult to use because of the limited variety of worn clothes and non-robustness for an outdoor environment. Installation of a marker-less multi-view-based motion capture system to various places also requires complicated procedures depending on the setting. Despite all of these hardware constraints, there are a lot of ways to capture different body shapes, clothes, motions, and background environments.

In this paper, we present the MonoEye dataset[1]. It is a large-scale synthetic dataset with the system's unique hardware setup (Figure 3.8). We render humanoid models using SMPL [89] and animate the model using the CMU MoCap dataset [4]. Four humanoid models (two males, two females) are used to create various body shapes and appearances, and the clothing texture is randomly selected from the SURREAL dataset [138]. We sample and apply omnidirectional images of the SUN360 dataset [142] for background texture.

We applied the camera lens parameters to the virtual camera using a custom shader in order to mimic lens distortion. The virtual camera is mounted to the chest of the human model. Random rotating and positioning are applied in order to consider the movement of the camera in the real world. We cropped the head part from the rendered image for training HeadPoseNet. Furthermore, we apply random gamma correction, lighting, and color temperature to ensure that the networks become insensitive to the specific photometric response characteristics of the specific camera. Finally, we render 680K images with 3D annotated labels.

[1]Dataset is available at: https://github.com/koikelab-team/MonoEye-Dataset

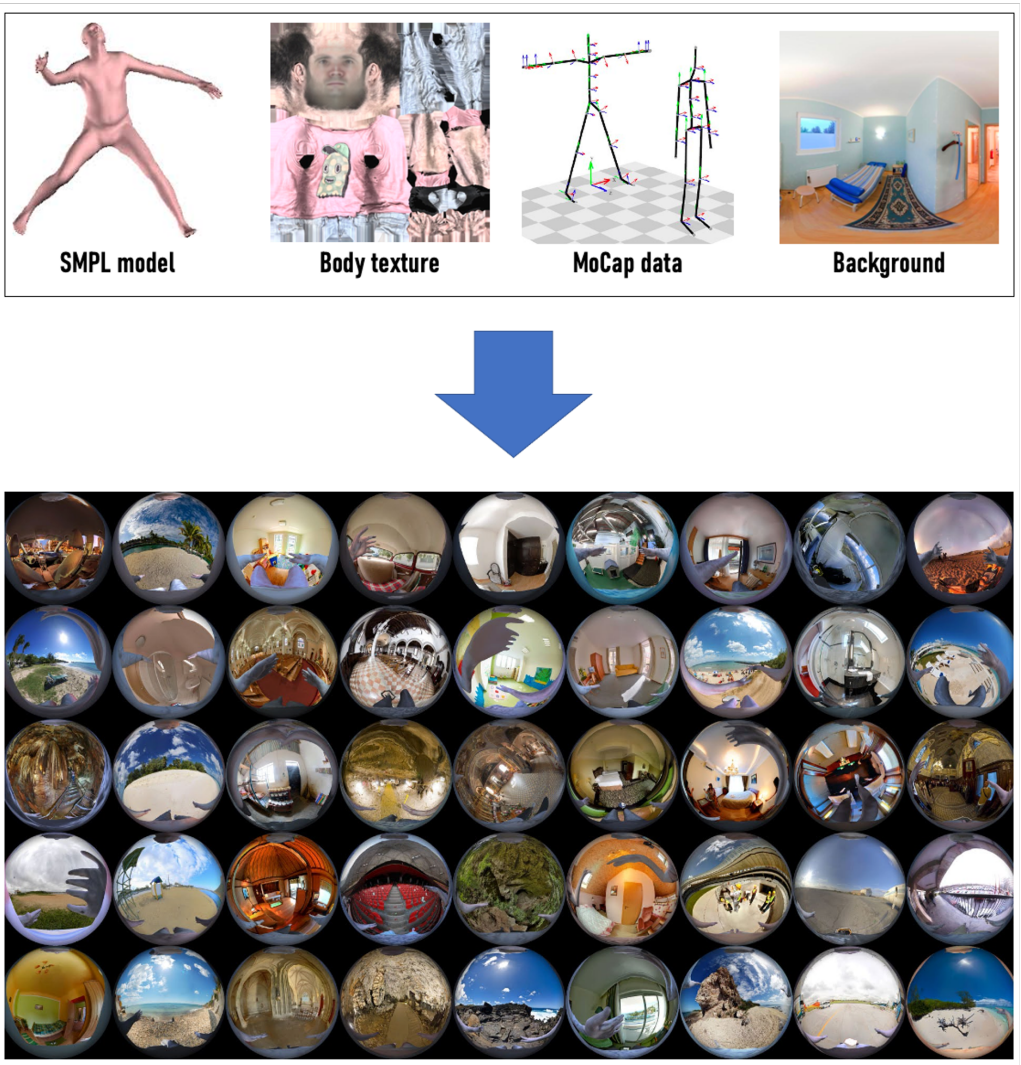


Figure 3.8: Example images of our synthetic dataset. The MonoEye dataset consists of a large variety of poses, body shapes, appearance textures, and backgrounds.

3.5 Network Training

We use PyTorch [105] to implement and train the networks. A multi-stage training method [95] is applied to train BodyPoseNet. First, we train the network with 2D human pose datasets such as LSP [71] and MPII [10] that consist of real-world images. This stage makes the network learn how to extract primitive features from real-world images. Then, we train the 2D branch of BodyPoseNet. The ResNet backbone's learning rate is set to 1/1000 times the current learning rate to keep the current weights for primitive feature extraction. After this, we freeze the backbone and 2D branch. Then, we train the 3D branch to estimate 3D pose from current 2D information. The model has been trained on the entire training set for three epochs using a mini-batch size of 4.

We utilized a similar strategy that is used to train BodyPoseNet in order to train HeadPoseNet. First, we train the network with the 300W-LP dataset [160] that contains real-world head images with rotation annotations. Then we applied the learning rate multiplier with 1e-3 to the ResNet backbone part, and trained the network with our synthetic dataset. The model has been trained on the entire training set for six epochs with a mini-batch size of 16.

To train the networks used in the third-person pose estimation pipeline, we fine-tune the MLP network, pretrained with Human3.6M [68] with our synthetic dataset. Since we confirmed that Mask R-CNN and HRNet work well for equirectangular images, we decided to use these models with pre-trained weights from the COCO dataset [88].

The 2D keypoints obtained by applying Mask R-CNN and HRNet to the images in the synthetic dataset was used as input. The labeled 3D pose must be corrected by rotating it by the azimuthal angle with respect to the vertical direction. This is because a person facing the camera from the side of the camera is facing from the front on the equirectangular image, but the 3D pose is still facing sideways. The model has been trained on the entire training set for 40 epochs with a mini-batch size of 64.

A pretrained ImageNet [119] weight is applied to the backbone part and the network is trained with the synthetic dataset in order to train CameraPoseNet. The model has been trained on the entire training set for six epochs with a mini-batch size of 16. We used the Adam optimizer [79] with an initial learning rate of 1e-3 for training all networks. In order to accelerate convergence to the optimal solution, batch normalization layers and a Leaky ReLU activation function [91] with 0.2 leakiness are applied before the output layer.

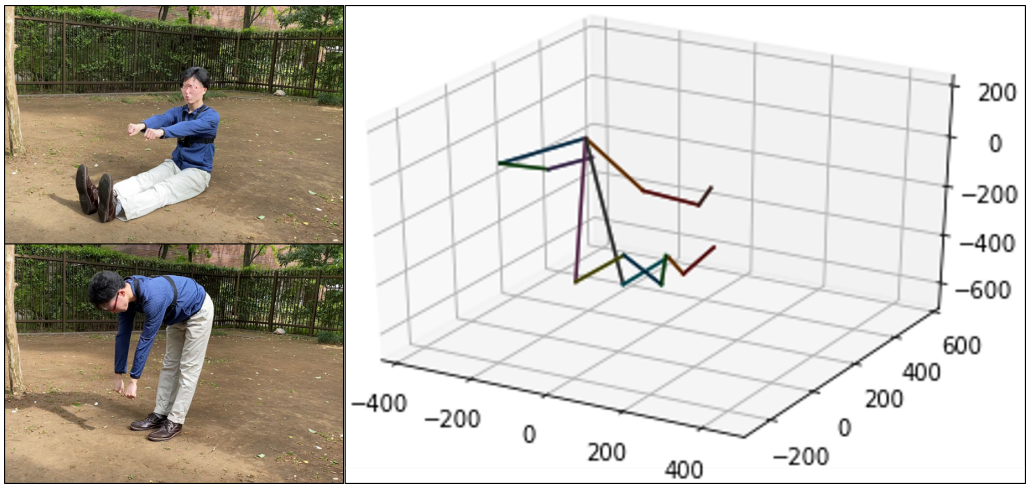


Figure 3.9: Example motions that contain the same camera-relative 3D joint position and the different camera orientations.

3.6 Post Processing

3.6.1 Temporal Filtering

It is difficult to ensure temporal consistency of motion for video input with our deep neural networks' per-frame estimation results. The small error between frames could lead to temporal jitter and this may not be acceptable in some applications. We applied the one-euro filter [25] that is effective for real-time noise reduction to raw prediction results in order to reduce the jitter and obtain temporally stable results.

3.6.2 Camera Orientation-aware Human Pose Estimation

Our method can estimate the camera orientation by leveraging the wide environment view which is captured by the chest-mounted camera. Most egocentric camera-based human pose estimation networks cannot estimate camera orientation and cannot classify poses that contain the same camera oriented joint position with a different camera orientation as shown in Figure 3.9. We combined the camera-relative 3D body pose with the camera orientation feature in order to address this problem. First, the camera rotation information estimated by CameraPoseNet is converted to a rotation matrix. Then the dot product between the BodyPoseNet's estimation result and the rotation matrix is calculated. The result of this simple process is a human pose that contains camera-orientation information. Thus, we can distinguish the poses which have the same joint position and different camera orientations.

3.6.3 Global 3D Position Computation for Third-Person's Pose

The absolute camera-centered global position of the human is reconstructed with a simple and efficient way using the azimuth θ and distance d.

The azimuth is calculated from the 2D keypoint of the target's pelvis $K^{root}_{[x]}$ with the following equation,

$$\theta = (\frac{2K^{root}_{[x]}}{W} - 1)\pi, \tag{3.1}$$

where W denotes the width of the image.

The distance from the camera d is calculated with the following equation based on Mehta et al.'s algorithm [98].

$$d = \frac{\sqrt{\sum_i ||P^i_{[xy]} - \bar{P}_{[xy]}||^2}}{\sqrt{\sum_i ||K^i - \bar{K}||^2}} f, \tag{3.2}$$

where P^i and K^i denote the ith joint position in 3D and 2D, respectively, $P_{[xy]}$ the x, y part of P, $\bar{P}$ the mean of P over all joints, and f the focal length.

Finally, the global location (X, Z) is calculated as:

$$X = d\sin\theta, Z = d\cos\theta. \tag{3.3}$$

The height from the ground Y is the distance of the y-axis from root to ankle, assuming that the feet are on the ground.

3.6.4 Viewport Estimation

The following procedures are performed to estimate the viewport. First, the ultra-wide fisheye image is mapped to a sphere and then the virtual camera, which applies the estimated head orientation, projects the viewport area. However, the fisheye image is captured from the chest of the user and because of the difference in location between head and chest, the estimated viewport is not accurate if we don't consider this difference to estimate viewport (see Figure 3.10.a). The camera-relative location of the user's eye is required to reduce this difference, however, our camera can only capture the lower face part of the user, hence it is difficult to estimate the eye position. Instead of estimating the position of the user's eye, we use the estimated neck position and add the offset value of the distance from the neck to the eye that is calculated by average face statistics data. Finally, the virtual camera is located

with the correct position and projects the viewport of the user with the estimated head orientation as illustrated in Figure 3.10.b. Since our system doesn't estimate a depth map, we use a constant distance value between the sphere and the virtual camera. Therefore the estimation result may have some errors depending on the distance of the real-world object.

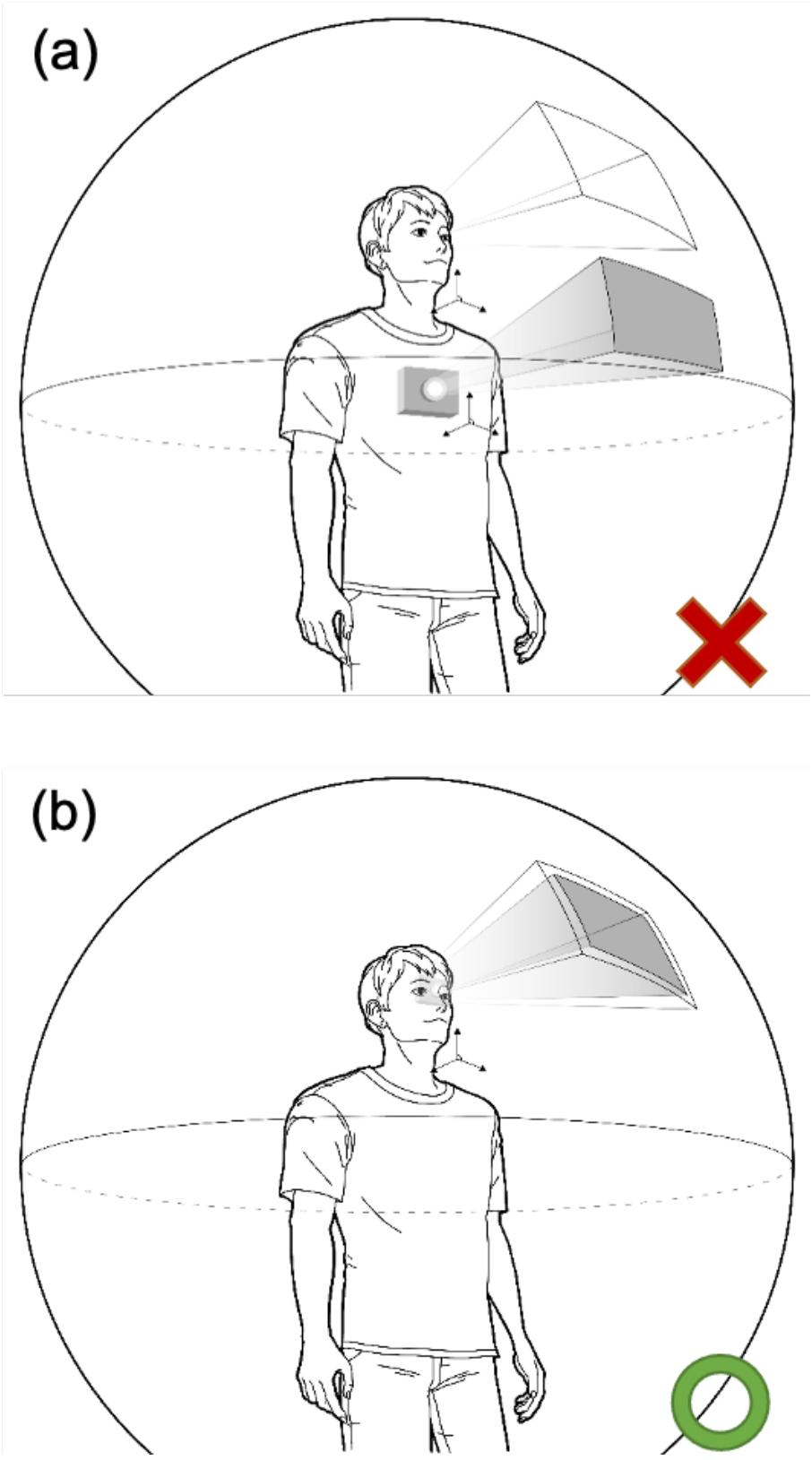


Figure 3.10: (a) Incorrect viewport estimation by the position gap between the virtual camera and user's eyes and (b) correct viewport estimation with aligning the virtual camera's position with the eye position.

3.7 Quantitative Evaluation

The proposed system is quantitatively evaluated on a variety of datasets. In this section, we will describe the datasets and the evaluation metrics that are used for evaluation and then we will analyze the results.

3.7.1 Dataset and Evaluation Metric

MonoEye Dataset

The test set of the MonoEye dataset is used to evaluate our deep neural networks. The test images and annotations are randomly sampled from the MonoEye dataset, and these are not used to train the networks. The test set consists of 200K images that contain various motions, body shapes, and backgrounds.

Small Scale Real-world Dataset

In order to evaluate our neural networks' generalization capability for real-world scenarios, we captured a small scale real-world dataset with our prototype hardware settings. Eight different actions are performed with the human subject who wears the chest-mounted camera along with normal clothing for the test set of BodyPoseNet. The ground truth annotations are acquired using the custom motion capture system of the CMU Panoptic studio [73] (see Figure 3.11.a). We created a test set with a total of 16K frames of images and annotations (2K frames per action, See Figure 3.12).

To evalute our third-person pose estimation pipelien, we captured real-world data with Azure Kinect. The ultra-wide fisheye camera was fixed at a height of 1.2 m to ignore the effect of camera shake. 4 subjects (two males and two females) performed 3 different motions (walking, waving, dribbling). While performing these motions, they moved freely in an area at distances of 1-6 m and azimuth of -90 and 90 degrees from the camera. Each sequence took 20 seconds, resulting in a total of 7.2K frames of data.

We captured 2.1K frames of various head movements of the person using an IMU sensor-based motion capture suit for the test set of HeadPoseNet (see Figure 3.11.b).

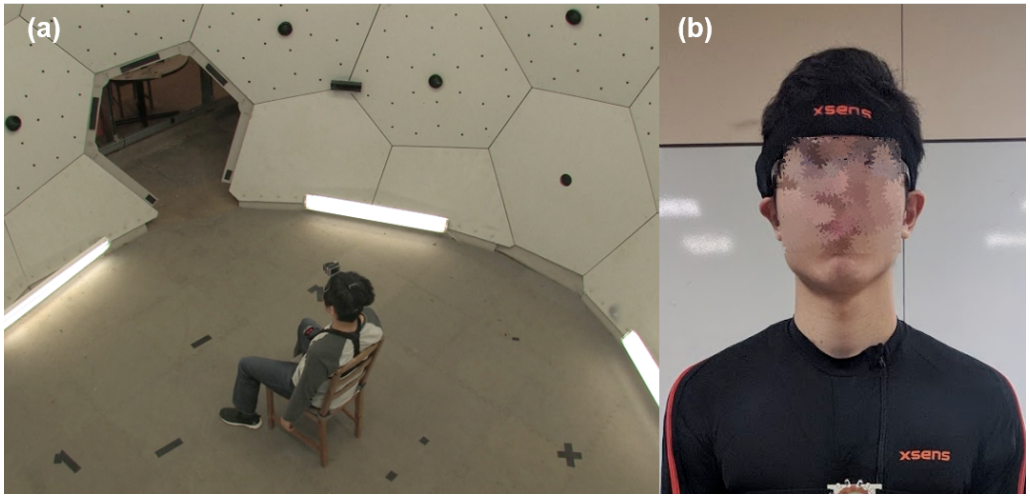
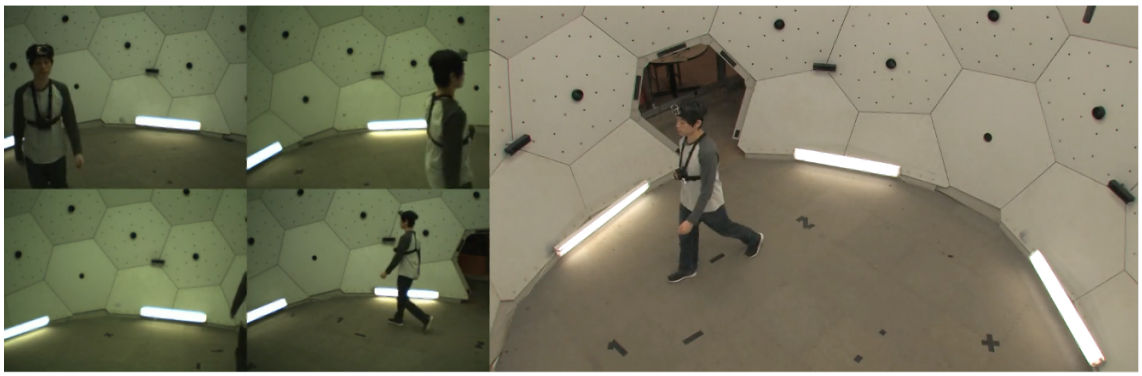


Figure 3.11: (a) An example view of capturing the ground truth of the human pose; (b) The wearable mocap suit to capture the ground-truth head pose.



(a) Walking

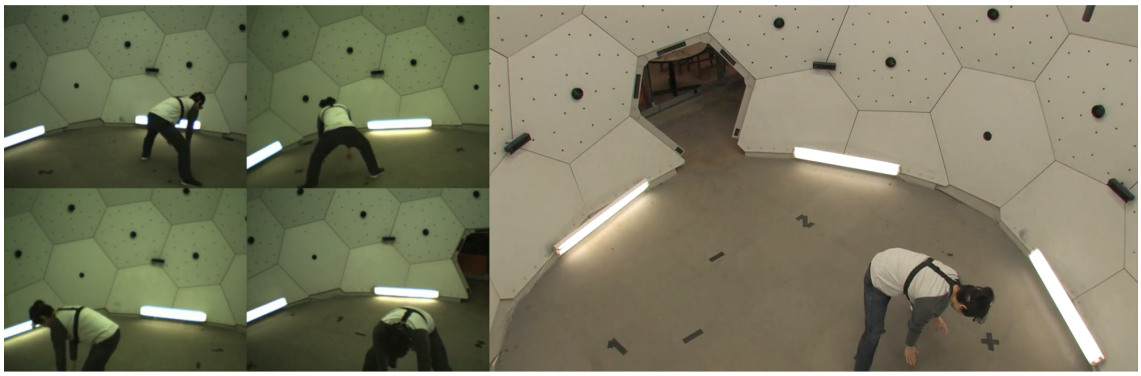
(b) Sitting

(c) Crawling

(d) Crouching

(e) Boxing

(f) Waiving



(g) Stretching

(h) Kicking

Figure 3.12: Example images for each action of our small scale real-world dataset captured from the Panoptic.

Evaluation Metric

We reported the Mean Per Joint Position Error (MPJPE) for BodyPoseNet and the third-person pose estimation pipeline:

$$MPJPE(P, \hat{P}) = \frac{1}{N_f}\frac{1}{N_j}\sum_{f=1}^{N_f}\sum_{j=1}^{N_j}\|P_j^f - \widehat{P_j^f}\|_2,$$

where P_j^f is the set of 3D points of the predicted result at frame f for joint j, and $\widehat{\ }$ is the ground truth. N_f and N_j indicate the number of frames and number of joints respectively. Since our real-world dataset consists of the global position of body joints, we applied Procrustes analysis before calculating MPJPE in order to match our estimation results to the ground truth.

For HeadPoseNet and CameraPoseNet, we report the Mean Absolute Error (MAE):

$$MAE(R, \hat{R}) = \frac{1}{N_f}\frac{1}{N_e}\sum_{f=1}^{N_f}\sum_{e=1}^{N_e}|R_e^f - \widehat{R_e^f}|,$$

where R_e^f is the predicted angle at frame f for Euler angle element e, and $\widehat{\ }$ is the ground truth. N_f and N_e indicate the number of frames and number of elements respectively.

The average inference time (ms) of each neural network is measured with a NVIDIA RTX 2080 GPU in order to demonstrate the real-time performance of our system.

3.7.2 Accuracy Results

BodyPoseNet

We evaluate our approach on the synthetic and the real-world test sets. We didn't establish a comparison with state of the art monocular egocentric human pose estimation methods because of non-available public source code [144] and the large network structures these methods use that do not fit our concept [135].

Table 3.1 shows the accuracy results of BodyPoseNet. Despite heavy lens distortion and occlusion, BodyPoseNet shows 43.6mm of MPJPE for the synthetic dataset. This model shows 84.9mm of MPJPE for the real-world dataset. We can see that the motion categories that are composed of static movements with less occluded body parts have higher accuracy. Note that we achieve this performance without fine-tuning BodyPoseNet with the real-world dataset and these results show that

our proposed network has good generalization capacity.

	MonoEye dataset	Real-world dataset								
Category	All	Walking	Sitting	Crawling	Crouching	Boxing	Waiving	Stretching	Kicking	All
Upper body	**24.6**	75.3	69.7	108.6	69.6	86.6	130.7	133.7	77.6	**94**
Lower body	**62.5**	71.1	71.3	88.6	66.2	75.1	77.8	83.5	72.0	**75.7**
Average	**43.6**	73.2	70.5	98.6	67.9	80.9	104.3	108.6	74.8	**84.9**

Table 3.1: Results of BodyPoseNet's raw predictions on the synthetic and real-world test sets. Metric: MPJPE and PA-MPJPE (mm).

Interestingly, we observed that the results for the lower body and the upper body obtained from BodyPoseNet have opposing behavior to each other with regards to accuracy for each dataset. In order to discover the reason for this, we analyzed the reconstruction error of per joint type for the real-world dataset in Table 3.2.

Joint	Error	Joint	Error
Neck	79.8	Pelvis	88.4
Shoulder	67.9	Knee	67.2
Elbow	71.4	Ankle	66.5
Wrist	154.1	Toe	80.7

Table 3.2: Average reconstruction error per joint, evaluated on the real-world test set. Metric: MPJPE (mm).

We can see that the error in the wrist part is significantly higher than other joints. We presume the reason for this high error is as follows. Firstly, since our synthetic dataset doesn't contain hand animations, we obtained weaker results from this network in real-world images which contain various hand movements. Secondly, hands have a large range of motion and it leads to a high scale of distortion in the fisheye lens. The error distribution between the upper body and the lower body is uniform except for the wrist. This is because the distance from the chest mount camera to each joint is relatively uniform, as described above.

Third-person Pose Estimation Pipeline

	Walking	Waving	Dribbling	Avg
Before fine-tuning	66.1	51.8	75.0	64.2
After fine-tuning (Ours)	**59.1**	**43.7**	**64.5**	**55.8**

Table 3.3: Results of our third-person pose estimation pipeline. Metric: PA-MPJPE (mm).

Table 3.3 shows the comparison results between before and after fine-tuning the model on the real-world dataset. This model shows 55.8mm of MPJPE for the real-world dataset, and this proved the effectiveness of fine-tuning.

HeadPoseNet

	Yaw	Roll	Pitch	Average
MonoEye dataset	4.4	4.5	3.3	**4.1**
Real-world dataset	16.9	11.3	11.3	**13.2**

Table 3.4: Results of HeadPoseNet's raw predictions on the synthetic and real-world test sets. Metric: MAE (°).

Since the landmark-based methods [77, 84] cannot estimate the head pose from the lower face image, we evaluated our network without comparison. The accuracy results of HeadPoseNet are shown in Table 3.4. The average Euler angle errors, that are evaluated through the synthetic and the real-world test sets, are 4.1° and 13.2° respectively. We presume that the reason for the accuracy gap between these datasets is that the trained network is not fine-tuned on real-world images and because the real-world test set mostly includes more extreme head movements, which are not included in our training set. Nevertheless, HeadPoseNet shows the comparable accuracy of general landmark-based methods [77, 84] with third-person view face images, and if the network can be fine-tuned with real-world images through a simple calibration procedure, higher accuracy can be achieved even if there is only a lower part of the face image available.

CameraPoseNet

Pitch	Roll	Average
3.4	2.4	**2.9**

Table 3.5: Results of CameraPoseNet's raw predictions on the synthetic testset. Metric: MAE (°).

Since the background images of the synthetic dataset consist of images taken in the real world, the evaluation of CameraPoseNet is performed with the test set of the MonoEye dataset, and the accuracy results are shown in Table 3.5. Because of this being a relatively easy problem, we have designed CameraPoseNet with a simple structure, and it solves this problem easily with high accuracy even with the simple network structure. According to these results, we demonstrated that camera pose estimation can be implemented with a simple network by using MonoEye's chest-mounted camera. This implementation was difficult due to the limited environment view of the top-down view-based methods.

3.7.3 Inference Time

	BodyPoseNet	HeadPoseNet	CameraPoseNet
Average Inference Time	12.3	5.9	2.7

Table 3.6: Results of an inference time benchmark for each neural network. Metric: average inference time (ms).

Table 3.6 shows the average inference time of each network. BodyPoseNet which has the largest structure can run at 80Hz, and HeadPoseNet and CameraPoseNet, which are composed of relatively light network structures, can run at 165Hz and 370Hz or higher, respectively. Since each network is designed to be lightweight in consideration of real-time performance, even if all networks are simultaneously used, they can all run above 47 Hz without optimization. Thus, our approach can be applied to many applications in which real-time performance is critical such as motion control in virtual reality and real-time interaction applications. In addition to this, it is possible to improve the target performance of the application by selecting the neural networks to be used according to the purpose of the application. Recently, various neural networks for mobile devices and training methods for them have been proposed. Thus, we expect that MonoEye's networks can be ported to mobile devices and can then be used as a fully portable system.

3.8 Qualitative Results

MonoEye's simple hardware setup enables multimodal motion capture in everyday life without restrictions on location and time. Different from conventional methods, our system uses the ultra-wide chest-mounted camera that captures not only the user's 3D pose but also the user's viewport and the surrounding environment in which there are various activity cues.

We can see that our method estimates accurate 3D pose that is comparable with third-person view-based monocular motion capture methods. The combination of RGB's images and deep learning methods provide stable results even in outdoor environments (see Figure 3.13). We can estimate 3D human pose with camera orientation information as illustrated in Figure 3.14, by combining the prediction results of CameraPoseNet and BodyPoseNet. We can distinguish the motions that have the same position and different camera directions, that previous portable motion capture systems are not able to distinguish.

We show some qualitative results of our third-person pose estimation on images captured indoor and outdoor in Figure 3.15. We observe that our method estimates accurate 3D poses comparable to monocular motion capture methods based on third-person viewpoints, despite the strongly distorted fisheye images of the input. Our system is able to estimate the other person's position and pose even at a distance of 1 m from the user (Figure 3.15.a) and with the person on both sides (Figure 3.15.b), which is difficult to do with a normal egocentric camera. Since we use RGB images as input, our method can be used outdoors (Figure 3.15.c) or indoor (Figure 3.15.d), unlike infrared camera-based methods.

As shown in Figure 3.16, the viewport estimation method using HeadPoseNet accurately captures the user's viewport. We believe that if the fisheye camera has a high resolution, it can replace the existing action camera or glasses type camera that can capture viewport.

MonoEye enables the capture of various multimodal information using only a single camera without the restrictions of environment and time. Therefore, the system can be used not only for simple motion capture related applications but also for various interactive applications. We also note that the system does not require calibration or any complicated post-optimization.

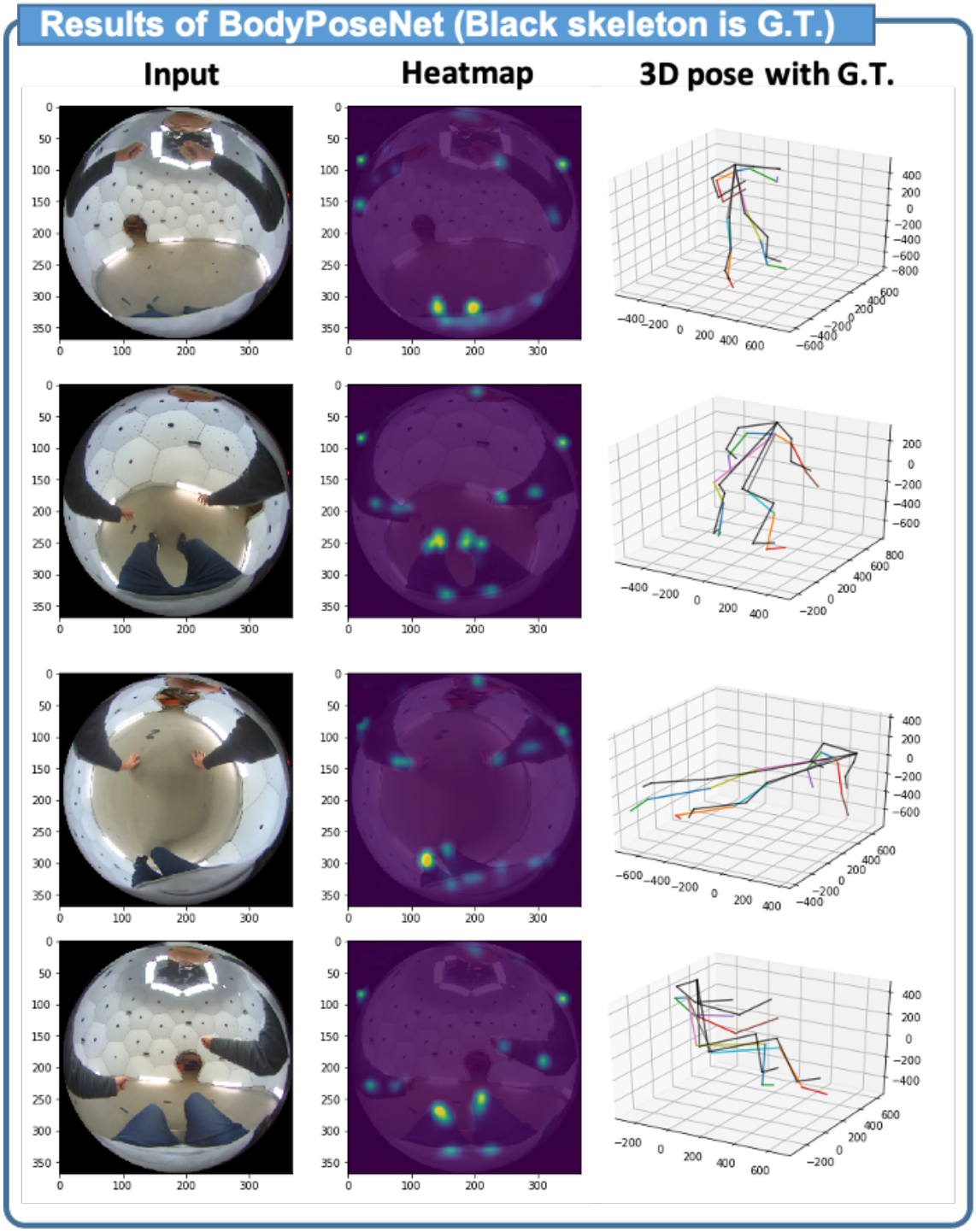


Figure 3.13: The 2D heatmap and 3D pose results from a side view, the black skeleton is the ground truth acquired using the motion capture system in CMU Panoptic studio [73]

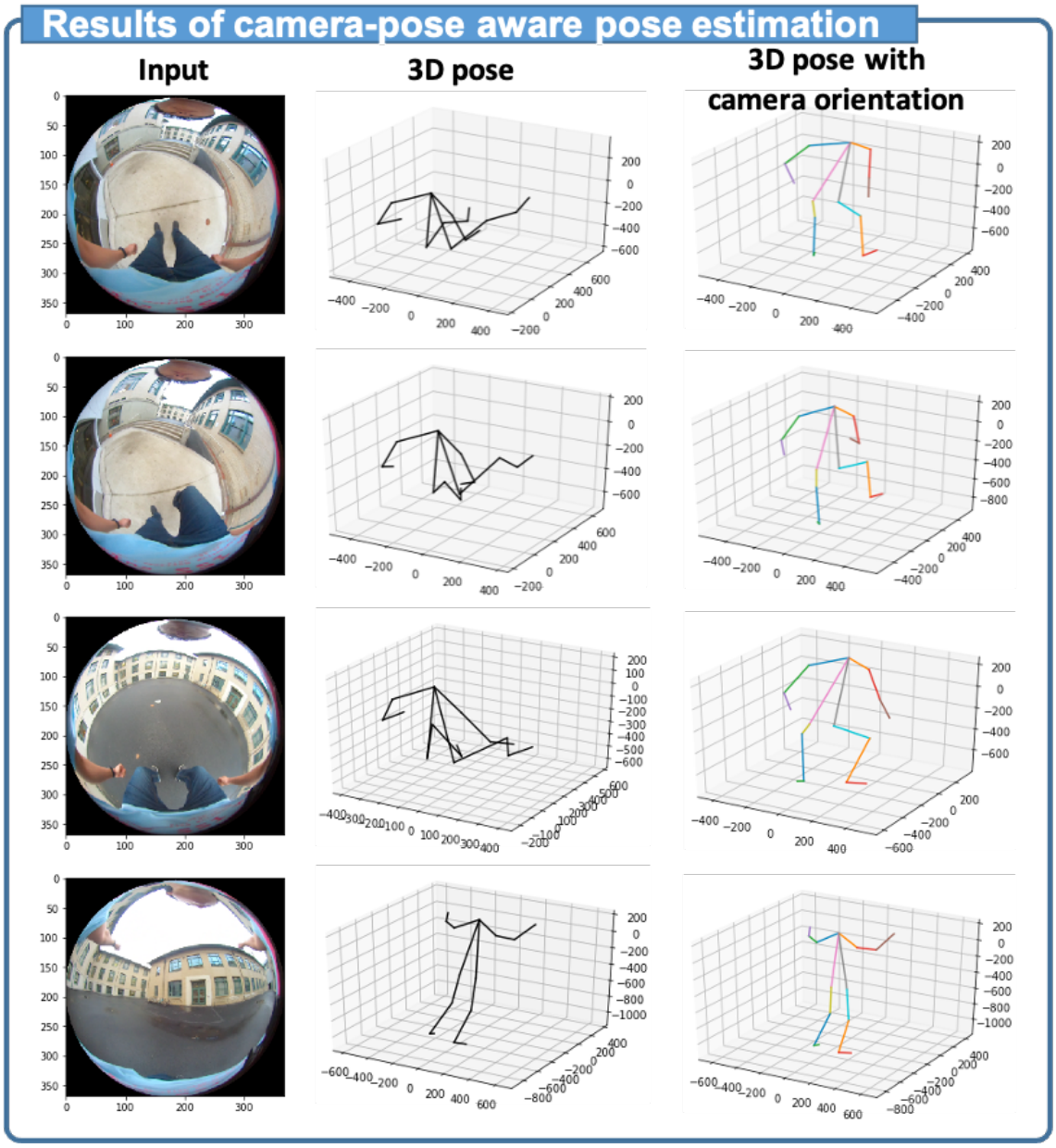


Figure 3.14: Comparison between camera-relative 3D pose and camera orientation-aware 3D pose.

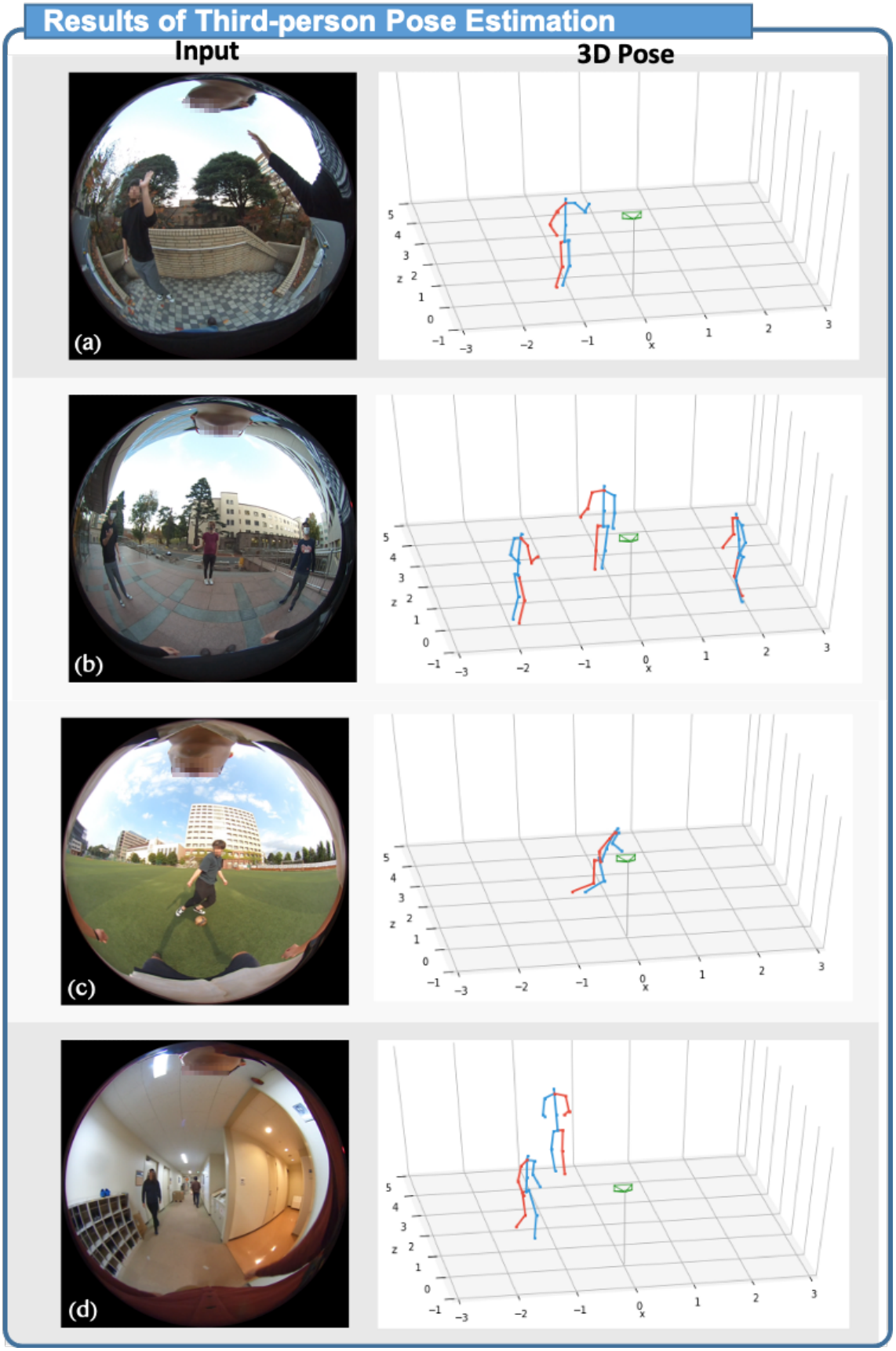


Figure 3.15: Results of our third-person pose estimation. Input fisheye images (left) and our absolute camera-centered global poses results (right). The position of the camera is indicated with a green symbol.

Figure 3.16: Results of the viewport estimation.

3.9 Applications

In this section, we have demonstrated several applications to show the capability of MonoEye to realize a variety of human motion and context-aware interactions.

3.9.1 Portable Motion Capture

Our system's lightweight hardware configuration enables the user to capture their motion anytime, anywhere. Since the system doesn't require any calibration procedure, the user can capture their motion by simply wearing the device (see Figure 3.17.a). The proposed method, consisting of an RGB camera and deep neural networks, can be operated regardless of indoors or outdoors and therefore, it is possible to record body pose, viewport, and the surrounding environment seen by the user in daily life. The system also enables the user to control a virtual character with their full-body motions, instead of typical physical controllers, and it allows the user to interact with the virtual environment and helps the user to become more immersed in the virtual environment.

3.9.2 Activity Highlighter

MonoEye can be used as a lifetime logger that records user activities due to the characteristics of the camera worn on the body. It is difficult to find moments the user wants to check from the long duration video sequences. An application to scan first-person videos based on various cues has been proposed [61] in order to solve this problem. It is possible to highlight specific activities from the entire video sequence since our system also captures various multimodal cues. We can specify the timeline where a specific object is detected by applying the object detection network to our estimated viewport sequences as shown in Figure 3.17.b. In addition to this, the frame that detected the intersection between the 2D hand's position and the object can be determined as the moment when the user and a specific object interact.

3.9.3 Context-aware Voice Control

The intention of the user sight is usually aimed at the object with which the user is interacting and MonoEye's viewport estimation enables intuitive real-world objects interaction using the image context. For example, objects of the same category are classified by giving them an alias during the existing voice recognition-based object interaction. On the other hand, even if there are many objects within the

same category, our viewport-based method can acquire image features of the object. Therefore, the user can interact with the specific object by simply placing eyesight on the target object and making a simple voice command to, for example, a smart home device as shown in Figure 3.17.c.

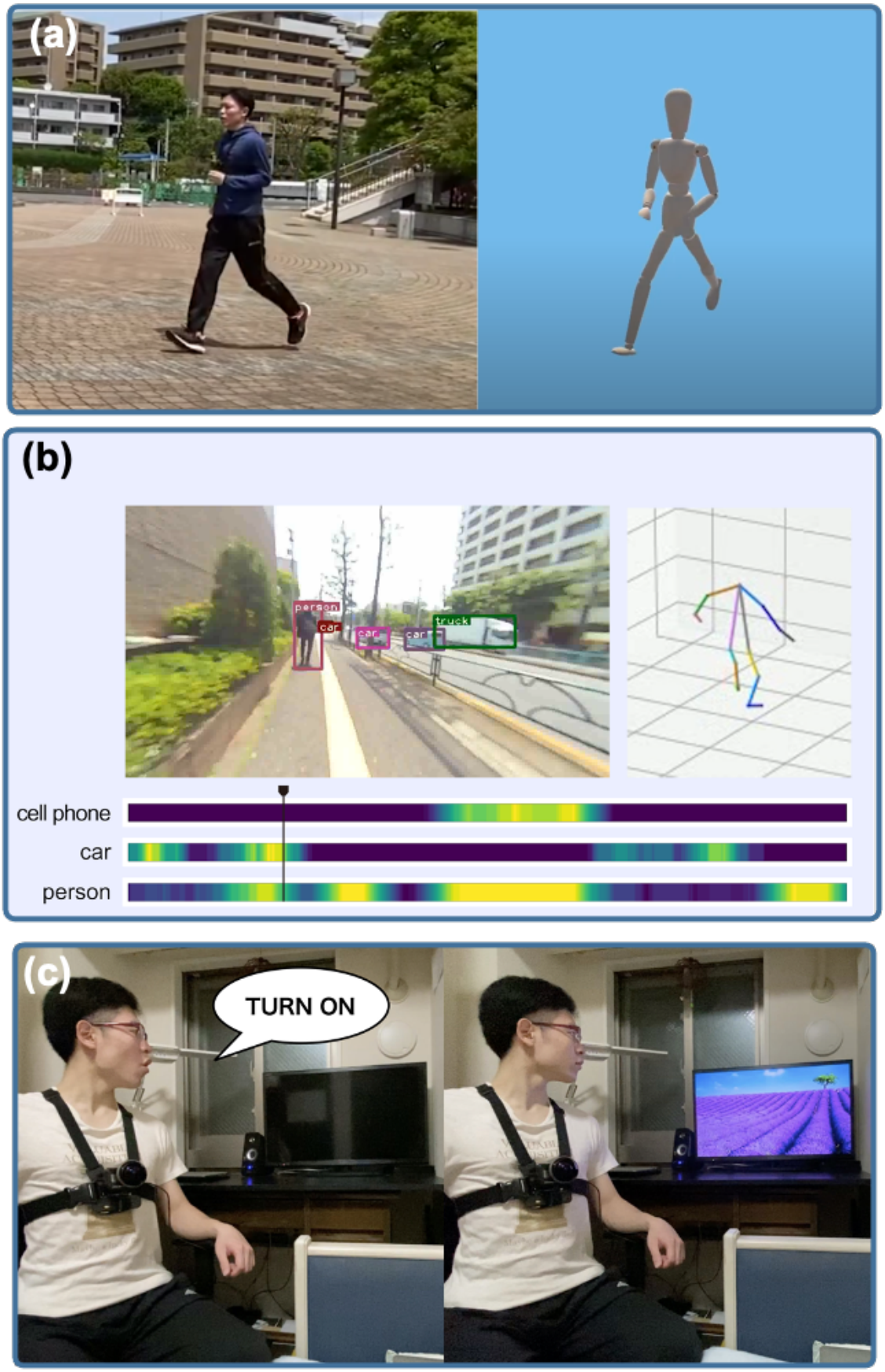


Figure 3.17: Examples of applications of MonoEye. Our system can be utilized from (a) portable motion capture to intuitive interactions in everyday life such as (b) activity highlighter and (c) context-aware voice control.

3.10 Discussion and Future Work

MonoEye is able to capture multimodal human motion using a single wearable camera. We believe that the ultra-wide view based on the wearable RGB camera enables various interactions without time and environment restrictions. Therefore, in this work, we explore the feasibility of this new concept through the implementation of a proof-of-concept prototype. In this section, we will discuss the main challenges and limitations confronted by the current prototype system and we will present future work to address these limitations.

3.10.1 Social Acceptance

The core part of the MonoEye prototype hardware is an ultra-wide-angle lens that allows the camera to acquire the limbs of the wearer and scenery. Because of limitations of current optical technology, the size of the ultra-wide-angle lens is large, and consequently, our prototype device is relatively bulky. However, we believe that

Figure 3.18: Future blueprint of the chest-mounted camera: the camera will be miniaturized and developed into various everyday accessories.

in the near future advanced optical technologies will achieve miniaturization of an ultra-wide-angle lens and as illustrated in Figure 3.18, cameras will take a variety of forms like everyday life accessories such as necklaces, brooches, tie pins, and sports gear which are highly accepted by users. To support this opinion, a necklace-type small camera with a wide-angle lens has been already released[2]. In the future, MonoEye can be realized with an advanced small wearable camera. We will design efficient, lightweight deep neural networks that can run on small devices within the scope of future works.

[2]https://www.insta360.com/product/insta360-go

Since our hardware uses a chest-mounted camera, the camera captures not only the user but also people in the surrounding environment, which can lead to privacy problems. Some people believe [64] that privacy perspectives may change as wearable cameras become more popular, nevertheless, we will consider how to delete information from a large number of unspecified people captured by the camera to address this issue (e.g. blurring out faces).

3.10.2 Mount Position of Wearable Device

Most of the wearable camera-based motion capture systems use head-mounted camera-based top-down views because they are easy to attach on virtual/augmented reality devices such as HMDs and because they acquire the entire body image of the user. However, even if the head-mounted camera is miniaturized, the camera needs to be installed on mount equipment such as a cap, headband, or HMD. Thus, there is the problem of low usability and social acceptance in everyday life as we discussed above. On the contrary, miniaturized wearable devices worn on the chest can be utilized in the form of various everyday accessories. We believe that users can wear and use the device in everyday life without social discomfort. In addition, a chest-mounted camera has the great advantage that it is able to acquire its surroundings. Our prototype system uses this advantage in order to demonstrate the feasibility of multimodal motion capture that can acquire not only the user's 3D pose information but also the viewport and camera pose. In the future, MonoEye can be utilized in various fields like gaze estimation, 3D reconstruction of people and environments, and sports broadcasting from novel viewpoints by using various deep neural networks.

3.10.3 Computer Vision Challenges

Despite the power of deep neural networks, the unique image domain of MonoEye faces much more complex challenges than a typical camera image domain. For example, estimating 3D information from a 2D image which contains limited visible body parts or face information is a highly under-constrained problem. Furthermore, the nonlinear distortion of the ultra-wide lens makes this problem more difficult. Even though MonoEye's networks are designed to restore invisible information from visible information, there is still a gap in accuracy compared to top-down view methods. In addition, despite the tricks applied when training the network, there is still a performance gap between real-world and synthetic datasets because of the domain

gap between real-world and synthetic images. We plan to improve the accuracy of MonoEye's networks by transferring knowledge of a stable pose latent space for efficient information restoration and making a photo-realistic dataset in order to reduce the domain gap.

3.11 Conclusion

We proposed a novel multimodal human motion capture system with a single wearable camera. Our lightweight deep neural networks estimate multimodal motion which includes 3D human pose, head pose, and camera pose from a single RGB image, and we can capture 3D human pose with camera orientation and viewport of the wearer by combining these estimation results. The MonoEye dataset, a new large scale synthetic dataset, has been presented to train the neural networks with this special image domain where it is difficult to get ground truth data in the real-world. We demonstrated the interactive potential of MonoEye via several application scenarios that utilize multimodal motion captured by our proposed system. We hope that this work will act as a trigger for egocentric camera-based interaction in everyday life.

Chapter 4

Lightweight 3D Human Pose Estimation with Teacher-Student Learning

4.1 Overview

We aim to estimate 3D human pose in real-time. Recently human pose estimation has achieved great progress and is being used for sports analytics, body and gesture motion capture in the AR (Augmented Reality) or VR (Virtual Reality) environment. As VR headset display technology becomes mature, various applications including entertainment, education, and telecommunication are getting released to the market. AR receives even more attention since AR does not require any additional equipment. Nevertheless, creating AR/VR content often requires special settings and devices. We believe that mobile-based marker-less motion capture system will accelerate the advance of AR/VR application market.

Conventional motion capture systems are marker-based relying on wearable suits with sensors and multiple cameras or need depth cameras (e.g. Microsoft Kinect[1], Intel RealSense[2]) to obtain human joint locations. These methods usually require expensive and specialized devices or are restricted to be used in the indoor environment due to calibration procedures and specific light sources required.

Recently, by leveraging the power of deep neural networks, human pose estimation technology with RGB images has been remarkably progressed. However, performance gains of deep learning-based models accompanies high deployment costs due to very deep and wide layers [132]. This leads to increased FLOPS (FLoating point

[1]https://developer.microsoft.com/en-us/windows/kinect
[2]https://software.intel.com/en-us/realsense

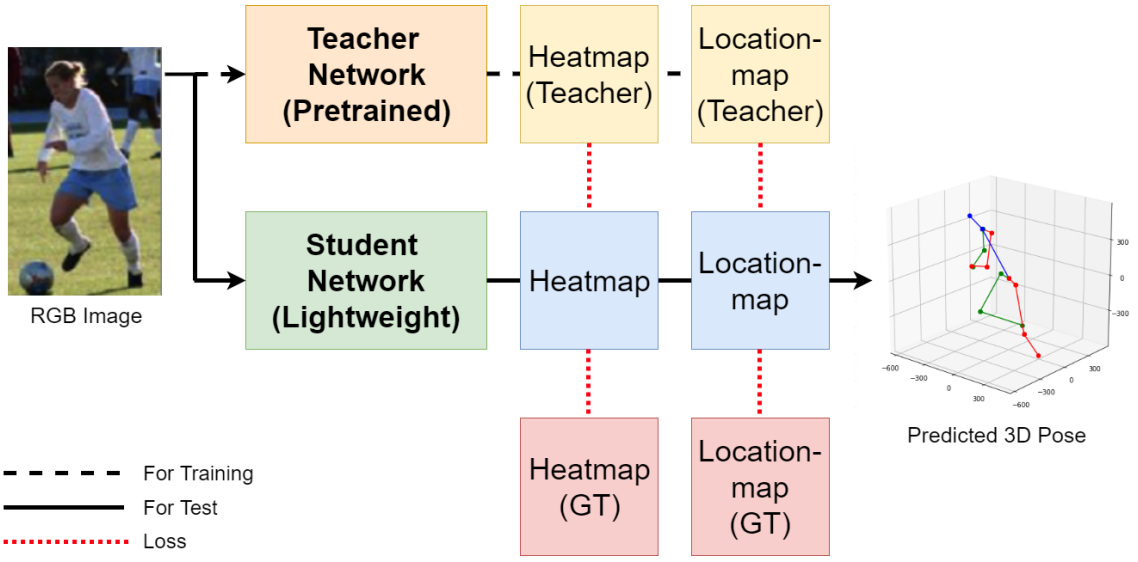


Figure 4.1: An overview of our proposed method. In order to train lightweight 3D human pose estimation model efficiently, we adopt the basis of knowledge distillation: (1) First, we train a teacher model, which consists of a large number of neural network layers. (2) Then, we train the lightweight model with extra supervision of the teacher model via mimicry loss functions for 3D pose knowledge transfer. The trained lightweight network does not depend on the teacher model and can perform efficient 3D human pose estimation.

OPerations per Second), which is not suitable for devices with limited computing resources such as smartphones or embedded systems. In order to reduce the number of FLOPS, a lightweight model is usually designed with a smaller number of parameters and with efficient operations such as depthwise convolutions. However, the significantly reduced amount of parameters affect the accuracy of the model. Methods using binarized convolutional neural network (CNN) or quantization [19, 21] often suffer from a lack of generalization capacity.

In this paper, we propose an efficient learning method for 3D human pose estimation model with minimal performance loss while reducing the number of parameters. We extend the 2D human pose estimation model learning method based on teacher-student learning [155] to 3D, and through designing and implementing MoVNect, a lightweight 3D pose estimation model. We observed that the lightweight model trained with the proposed approach achieves higher accuracy than the model trained with the vanilla method. In addition, we compare the inference time of our model with previous methods running on smartphones and develop an AR application with our model to show the effectiveness of the proposed method.

In summary, our contributions include:

- We design MoVNect, a lightweight 3D human pose estimation model that can

run in real-time on hardware with limited resources such as smartphones.

- We propose a method to efficiently train lightweight 3D human pose estimation with teacher-student learning (Figure 4.1). The proposed method shows an accuracy improvement of 14% than the vanilla training method on the Human3.6M test set.
- The inference time of various methods on smartphones is evaluated, and the feasibility of the proposed model to be used on various hardware is verified.
- We develop a real-time mobile application of 3D avatar with our proposed model to show the practicality of our approach.

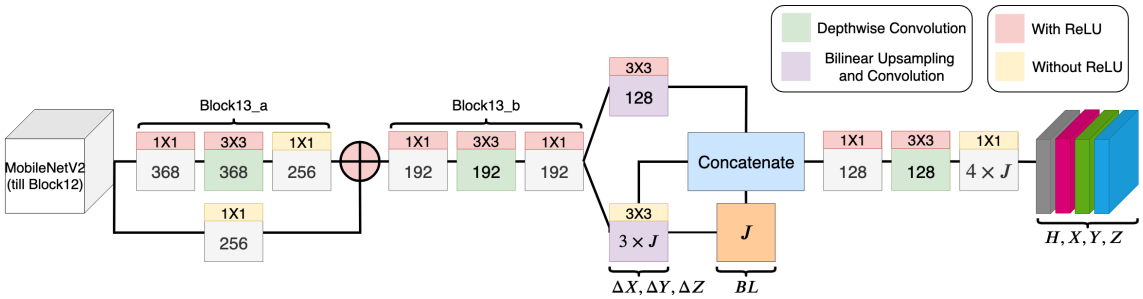


Figure 4.2: Network structure of MoVNect: a single RGB image is fed into the base network (MobileNetV2 till block12), and pointwise and depthwise CNN based structures are used for efficient feature extraction. The intermediate features, ΔX, ΔY, and ΔZ, are used for bone length-features, auxiliary cue to estimate root-relative 3D human pose. The network predicts heatmaps H and root-relative 3D joint location maps X, Y, Z.

4.2 MoVNect: Lightweight 3D Human Pose Estimation Network

In 3D human pose estimation, we estimate the 3D pose P^{3D} from a given RGB image I. $P^{3D} \in R^{3\times J}$ represents the root-relative 3D positions of the J body joints. We assume our network runs on low power devices (e.g. smartphone, embedded system). Therefore, the network estimates 15 joints ($J = 15$), a minimum requirement for the motion of 3D full-body characters.

4.2.1 CNN based 3D Pose Regression Network Architecture

Among previous 3D estimation approaches, the model proposed by Mehta et al. [98] has a good balance between accuracy and inference time. It is easy to apply knowledge distillation because the location map used in the model is 2D spatial information similar to the 2D heatmap. Therefore, we design a lightweight network architecture based on Mehta et al's approach [98] with the model search procedure (session 4.3.3) as shown in Figure 4.2. Our network produces the heatmaps and location maps for all joints $j \in 1..J$. We use the till of block 12 of MobileNetV2 [121] as the base network and adopt additional depthwise CNN layers for efficient computation (See Appendix A.2.1). We add the bone length-features to the network for an explicit clue to guide the prediction of root-relative location maps as:

$$BL_j = |\Delta X_j| + |\Delta Y_j| + |\Delta Z_j|$$

ΔX_j, ΔY_j, and ΔZ_j are intermediate features from our network. For efficient calculation, bone length-features are calculated using L1 distance instead of L2 distance-based equation proposed by Mehta et al. [98]. The calculated features are concatenated with other intermediate features and utilized to calculate the final output.

Inference: we use cropped images based on the person's bounding box when training our network. This makes our network performance affected by the size of the image at runtime. In order to address this issue while maintaining a real time processing on mobile devices, we acquire a bounding box based on the human keypoint K, found in the initial few frames of 2D heatmaps with a buffer area $0.2\times$ the height vertically and $0.4\times$ the width horizontally. We then track it continuously using previous frames with a momentum of 0.75. In order to normalize scale, a cropped image based on the bounding box is resized to 256×256 and used as an input to the network.

4.2.2 Extra Supervision based on Teacher-Student Learning

A brief outline of the proposed training method is shown in Figure 3.1. Most previous approaches with knowledge distillation are designed for object classification with softmax cross-entropy loss [12, 62] and not suitable to transfer pose knowledge (See Appendix A.2.2 for details). We design mimicry loss functions for 3D pose knowledge transfer based on the method of Zhang et al. [155]. The network is trained with heatmap loss function $\mathcal{L}_{HM}$ and location map loss function $\mathcal{L}_{LM}$ as

$$\mathcal{L}_{HM} = \frac{1}{J}\sum_{j=1}^{J}\{\alpha||H_j - H_j^{GT}||_2 + (1-\alpha)||H_j - H_j^{T}||_2\}$$

$$\mathcal{L}_{LM} = \sum_{j=1}^{J}\{\alpha||H_j^{GT} \odot (L_j - L_j^{GT})||_2 + (1-\alpha)||H_j^{GT} \odot (L_j - L_j^{T})||_2\}$$

where H_j and H_j^{GT} specify the heatmaps for the jth joint predicted by the model and ground truth, respectively. $\odot$ is the Hadamard product and L_j specify the location maps for the jth joint predicted by the model . GT and T indicate ground truth and predicted results by the teacher model, respectively. α is the blending factor between the ground truth and teacher model's loss terms and set to 0.5. The

teacher-student learning is conducted in each mini-batch and throughout the entire training process. After the training, we only use the student model, already learned with the teacher's knowledge.

Figure 4.3: 3D character control. The processed output can be easily utilized for handling a virtual avatar.

4.2.3 Post-processing

Our model performs CNN based per-frame pose estimation, which leads to a small jitter, an unacceptable artifact in graphics applications. In order to reduce this temporal jittering, we apply the 1 Euro filter [25] to the predicted 2D keypoint and use the filtered keypoint K to refer to the value of the location map. The acquired 3D pose is also filtered to reduce the temporal noise of the prediction results of the continuous images.

The root-relative 3D pose acquired from the cropped image with the bounding

box loses the global position information. In order to restore the global position P_G^{3D}, we use the following simple but effective global pose estimation equation [96]

$$P_G^{3D} = \frac{\sqrt{\sum_1^J ||P_{[xy]}^j - \overline{P}_{[xy]}||_2}}{\sqrt{\sum_1^J ||K^j - \overline{K}||_2}} \begin{pmatrix} \overline{K}_{[x]} \\ \overline{K}_{[y]} \\ f \end{pmatrix} - \begin{pmatrix} \overline{P}_{[x]} \\ \overline{P}_{[y]} \\ 0 \end{pmatrix}$$

where $\overline{P}$ and $\overline{K}$ are the 3D, 2D mean over all joints. $P_{[xy]}$ is the x, y part of P^{3D} and single subscripts indicate the particular elements. f is the focal length of the camera.

Since predicted 3D pose is the root-relative 3D position of each joint, it cannot be applied directly for character animations. Hence, inverse kinematics are applied to convert the 3D position of each joint into the orientation and these orientation values are also filtered with the 1 Euro filter [25].

In addition, since our model does not have explicit knowledge of the joint angle limits of the human body, our network does not explicitly decline physically invalid poses. In order to address this problem, we apply the anatomical joint rotation limits to the calculated angles of each joint to ensure bio-mechanical plausibility. Through the post processing, our approach exports data directly in a format suitable to 3D character control in real-time as shown in Figure 4.3.

4.3 Experiments

4.3.1 Experiment Setup

We evaluate our model using two measurements:

Accuracy: In order to measure the accuracy of the model, we use the Human3.6M [68] dataset, currently the largest 3D pose dataset. This dataset contains 15 actions performed by 11 subjects. We employ the commonly used evaluation protocol #1: subject 1, 5, 6, 7, and 8 for training and subject 9 and 11 for testing. Mean Per Joint Position Error (MPJPE) is calculated with the root-relative 3D joint positions from our network.

Inference Time: In order to confirm the applicability of the proposed lightweight model in the actual mobile environment, we measure inference time on smartphone devices (Apple iPhone series, See Table 4.1) with a variety of computing hardware specifications (CPU, GPU, NPU). We use the Apple Core ML[3] framework to convert neural network models to mobile ones and to run these models on the smartphone.

Device	iPhone 7	iPhone 8	iPhone X	iPhone Xs
SoC	A10 Fusion	A11 Bionic	A11 Bionic	A12 Bionic
RAM	2GB	2GB	3GB	4GB
NPU	X	X	O	O

Table 4.1: Mobile device comparison chart used for inference time benchmark.

4.3.2 Training Details

Since most 3D human pose datasets consist of indoor images only, the network, trained with only the existing 3D pose dataset, has a lack of generalizability for in-the-wild scenes. Therefore, following Mehta et al.'s method [96], we first pre-train the 2D pose estimation using LSP [72] and MPII datasets [11], and train the 3D pose estimation through Human3.6M [68] and MPI-INF-3DHP [96] datasets. Frames of 3D datasets are sampled with at least one joint movement by >200mm between them and cropped using the bounding box of the person. For the MPI-INF-3DHP dataset, the background augmentation is performed using the Places365 dataset [158], and finally 95k of MPI-INF-3DHP training samples and 100k of Human3.6M training samples are prepared.

We use the Keras [32] framework with the TensorFlow backend for training the network. Some random scaling (0.7-1.0) and gamma correction are performed on training. RMSProp optimization algorithm [134] with learning rate to 2.5×10^{-4}

[3]https://developer.apple.com/documentation/coreml

is used for 2D pose training and Adam optimization algorithm [78] with the same learning rate is used for 3D pose training. Mini-batch size is set to 4. We use the pre-trained base network with ImageNet [41] and batch normalization [67] before each non-linear activation.

4.3.3 Model Search

Network	Network Structure		Upsampling Method
	Block13_a	Block13_b	
Type A	368, 368, 256	192, 192, 128	Bilinear + Conv2D
Type B	368, 368, 256	192, 192, 128	TransposedConv2D
Type C	512, 512, 512	256, 256, 128	Bilinear + Conv2D

Table 4.2: Specification for our prototype MoVNect models. Sequential numbers on Network Structure column denote the number of CNN layers, which make up each block.

In order to find a suitable model, which has a good balance between accuracy and inference time, we design and train various types (Type A, B, C) of models that have a different number of layers on Block13_a, Block13_b, and upsampling method (Bilinear upsampling + Convolution, Transposed Convolution). See Table 4.2 for specification of our prototype MoVNect networks.

Network	Network Structure		# Param	MPJPE
	Block13_a	Block13_b		
Type A	368, 368, 256	192, 192, 128	**1.03M**	113.3
Type C	512, 512, 512	256, 256, 128	2.69M	**108.2**

Table 4.3: Performance analysis with the number of layers. Metric: average MPJPE(mm). M:10^6.

First, we measure the performance and inference time correlation with the number of layers. we design MoVNect-Small (Type A), MoVNect-Large (Type C) and measure the average MPJPE on the test set of Human3.6M as shown in Table 4.3. Because of the deep neural network's suboptimal trade-off between the representation capability and the computational cost, Type C has no significant improvement in accuracy (about 5mm improvement), even though the number of parameters in the network is twice that of Type A.

Next, we measure the change in accuracy according to the upsampling method which increases the resolution of the network output. As shown in Table 4.4, we compare Type A and Type B, which use bilinear upsampling with convolution and transposed convolution, respectively. According to the results, although transposed

Network	Upsampling Method	# Param	MPJPE
Type A	Bilinear + Conv2D	**1.03M**	**113.3**
Type B	TransposedConv2D	1.13M	126.7

Table 4.4: Performance analysis with upsampling methods. Metric: average MPJPE(mm). M:10^6

convolution method (Type B) requires more parameters, accuracy was lower than resize-convolution method (Type A). We presume that while transposed convolution method has a unique entry for each output window, resize-convolution method is implicitly weighted in a way that it reduces the high frequency artifacts. Based on these results, we finally choose Type A network for MoVNect.

4.4 Results

4.4.1 Accuracy Results on Human3.6M Dataset

Our results on Human3.6M are shown in Table 4.5. Our model shows competitive accuracy compared with other methods. In particular, the model trained with teacher-student learning (marked with †) shows significantly improved accuracy (14% average MPJPE reduction). Even though our model consists of a very small number of parameters, it has cost-effective accuracy. These results show that our proposed training approach has good generalization capability in yielding cost-efficient 3D pose estimation models.

We compare the computation amounts of networks in Table 4.6 (see column 2). Compared with the teacher model [98], our model only requires 7.1% (1.03M / 14.6M) parameters and 18.5% (1.35M / 7.3M) computational amount but achieves 82.7% (97.3 / 80.5) accuracy in average MPJPE. When compared with the best performer [81], our model with 3% (1.03M / 34M) parameters and 9.6% (1.35M / 14M) computational amount achieves 53.2% (97.3 / 51.8) accuracy. Our model with the proposed method has cost-effectiveness advantages over other alternative models. Note that we apply the teacher-student learning method without changing any network structure. Based on the results in Table 4.3, we presume that several times more parameters with additional layers are required to overcome the performance gap without our teach-student learning. This design choice is quite critical and inevitable in real-time applications.

In Figure 4.4, we show qualitative results on Human3.6M and MPII datasets to demonstrate the generalization of our network to general scenes.

Methods	Direct.	Discuss	Eating	Greet	Phone	Photo	Pose	Purch.	Sitting	SittingD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.	# Param
Zhou et al. [159]	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0	-
Du et al. [44]	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	1120.0	117.7	137.4	99.3	106.5	126.5	-
Park et al. [104].	100.3	116.2	90.0	116.5	115.3	149.5	117.6	106.9	137.2	190.8	105.8	125.1	131.9	62.6	96.2	117.3	-
Mehta et al. [96]	52.6	63.8	55.4	62.3	71.8	52.6	72.2	86.2	120.6	66.0	79.8	64.0	48.9	76.8	53.7	68.6	-
Martinez et al. [93] w/ SH	51.8	56.2	58.1	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9	19.3M
Mehta et al. [98]	62.6	78.1	63.4	72.5	88.3	63.1	74.8	106.6	138.7	78.8	93.8	73.9	55.8	82.0	59.6	80.5	14.6M
Pavlakos et al. [106]	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2	-
Yang et al. [147]	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6	-
Kocabas et al. [81]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**51.8**	34M
Ours	80.6	96.3	92.2	90.4	116.1	82.1	110.9	188.4	224.6	106.9	123.2	98.9	90.4	117.3	80.5	113.3	**1.03M**
Ours†	72.4	83.4	76.9	82.1	101.9	70.4	91.8	156.5	193.0	92.8	108.4	85.1	76.8	97.2	70.5	97.3 **(14%↓)**	**1.03M**

Table 4.5: Results of our network's raw CNN predictions. All frames of subject 9 and 11, cropped with the ground truth bounding box, were used for evaluation. † means the model trained with the proposed teacher-student learning method. Metric: MPJPE(mm). M:10^6.

4.4.2 Inference Time Benchmark Results on Mobile Devices

Methods	Cost-Effectiveness			Inference Time on Devices									
	MPJPE	# Param	FLOPS	iPhone7		iPhone 8		iPhone X			iPhone XS		
				CPU	GPU	CPU	GPU	CPU	GPU	NPU	CPU	GPU	NPU
Mehta et al. [98]	80.5	14.6M	7.3M	275	175	215	140	270	120	120	200	110	17
Martinez et al. [93] w/ SH	62.9	19.3M	22M	750	200	300	160	350	160	160	270	120	20
Kocabas et al. [81]	**51.8**	34M	14M	500	220	210	210	230	160	160	200	125	50
Ours	97.3	**1.03M**	**1.35M**	**48**	**56**	**40**	**33**	**37**	**28**	**28**	**32**	**22**	**6**

Table 4.6: Comparison of networks' cost-effectiveness and inference time on mobile devices with various hardware configurations. Metrics: average MPJPE(mm), the number of parameters, FLOPS, and average inference time(ms). M:10^6.

Table 4.6 (see column 3) shows the inference time benchmark results on mobile devices. Throughout all the devices we test, our model's inference time outperforms other networks. Even with low-end devices (iPhone 7 with CPU), our model runs out over 20fps and with high-end devices (iPhone XS with NPU), the throughput reaches over 160fps. Note that except our model, which has low FLOPS and memory consumption, there is no other method that can perform over 10fps on CPU and GPU. Compared to Mehta et al.'s model [98], which is used as the teacher model, our model performs at least 283% (iPhone XS with NPU) and up to 730% (iPhone X with CPU) faster throughput. Compared to the best performer [81], our model shows at least 393% (iPhone 7 with GPU) and up to 1042% (iPhone 7 with CPU) faster inference time.

As the device processing power increases, the difference in throughput between networks decreases. In particular, in the case of utilizing a dedicated neural network accelerator such as NPU, all models of the comparison group are able to process more than 20 fps. However, different from other networks, our network does not have a huge gap across different processing unit types. Hence, if the model inference is done by CPU in low-end devices and by NPU in high-end devices, the GPU could be fully utilized for graphic rendering, and this is a great advantage for CG applications. Furthermore, most users do not have smartphones equipped with a dedicated processor for neural networks. Our proposed approach is expected to contribute to the spread of deep learning-based interactive applications until high-end devices are deployed broadly.

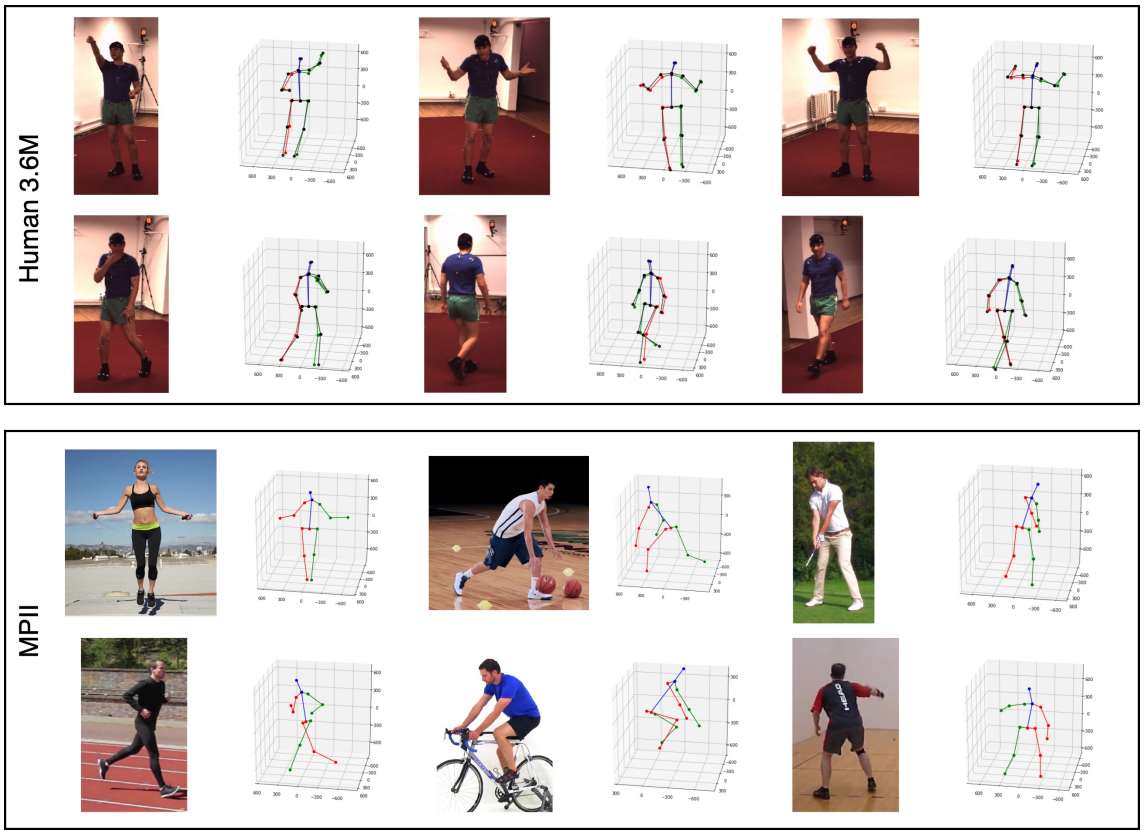


Figure 4.4: Qualitative results on the test set of Human3.6M(3D) and MPII(2D) datasets. Left: the input images; Right: the results of 3D pose prediction from a different viewpoint, the black skeleton is the ground truth of the Human3.6M dataset.

4.4.3 Applications

Our proposed network can be applied for various interactive mobile applications because it can provide motion data in a format suitable to 3D avatar control in real-time on mobile devices. In addition, since our network has low inference time, enough times remains for CG rendering for the application.

Augmented and Virtual Reality: Smartphone, which has built-in camera, inertial measurement unit sensor, and display, is the best portable device for AR and VR applications. Our method enables applications that provide the user with immersive content through a virtual avatar of the user exactly mimicking the user's real pose using a single RGB camera as shown in Figure 4.5. It also enables a real time interaction in body gesture capturing applications.

Motion Capture Simply Accessible: Our lightweight network can be used in a variety of devices that have low-computation power. Without communication with a high-performance server for processing algorithms, the network can be deployed and run directly on various mobile IoT devices in our daily life and can be applied in various real-life scenarios such as interactions with objects through body gestures, healthcare, and so on. For example, our algorithm can be used to recognize body language or to analyze walking postures of the elderly with common smart home devices.

Figure 4.5: AR-based real time 3D avatar mobile application. Our lightweight network can be utilized for interactive applications, which provide immersive experiences to users.

4.5 Discussion and Future Work

In order to the best of our knowledge, our training approach is the first knowledge transferring method for 3D human pose estimation networks. MoVNect achieves a well-balanced performance between accuracy and inference time. Nevertheless, it still has certain limitations that can be addressed in future work. In this paper, we transfer the knowledge to the lightweight network based on the location map, thanks to the similar output type to the 2D pose network. Furthermore, because the mimicry loss function is very simple, we envision that we can easily apply our knowledge transfer method to various 3D human pose networks.

We have focused on estimating the 3D pose of a single person, which can run in real-time across various devices. Currently, latest high-end smartphones tend to be equipped with dedicated accelerators such as NPU. The proposed fully-convolutional network could be scaled to multiple persons if such devices have enough computational capacity.

In order to reduce the resource and power consumption, most mobile deep learning frameworks do not fully support recurrent architectures. We also design our network based on per-frame prediction and this may lead to some temporal instability, similar to previous per-frame prediction approaches. We believe that our post-processing method should reduce temporal jitters enough to be usable and practical in various fields. Furthermore, in the near future, mobile devices will have more processing power and we will be able to expand per-frame to video processing levels using a recurrent approach.

In addition, to reduce inference time, our network uses a single scale of the cropped image. Processing each frame inference with multiple scales of the image (scale-space search) makes it difficult to guarantee real-time performance on low power devices. For applications that require a better accuracy for the pose, two different scales (like 0.7 and 1.0) of the cropped image can be used.

A few failure cases are illustrated in Figure 4.6. Our location map-based approach relies on 2D heatmap detection results and our lightweight model is not robust enough to occlusion. As future work, we will apply a pose encoding-decoding scheme [97], robust to occlusion, to our network. Despite these limitations, we observe that our method proposes an initial step in the direction of a training method for efficient lightweight 3D motion capture.

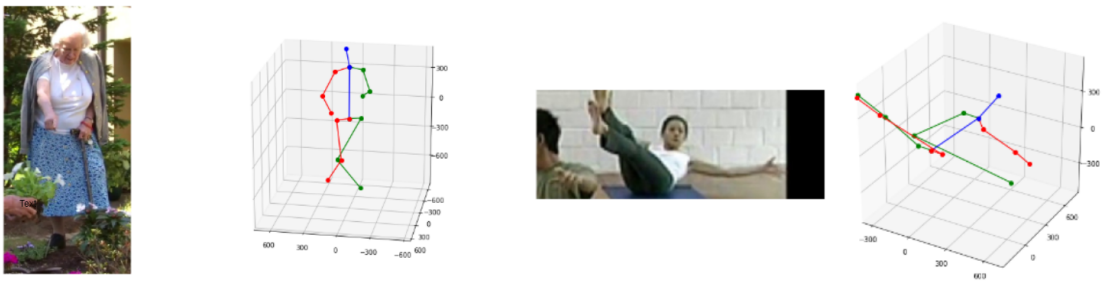


Figure 4.6: Failure cases of our model. Left: knees are crossed because of body part occlusion. Right: the position of the right hand is mislocated to the left hand because the right hand is occluded with the extreme pose.

4.6 Conclusion

In this paper, we propose MoVNect, a lightweight 3D human pose estimation model, and an efficient training strategy based on teacher-student learning. We make the step from existing 2D pose estimation with knowledge distillation to 3D pose estimation. Moreover, we present extensive evaluations on human pose and inference time benchmarks. Based on the results, we observe that our proposed teacher-student learning method significantly improves the accuracy of the model, and our network trained with the proposed method achieves very fast inference time with reasonable accuracy on various devices from low-end to high-end. We demonstrate these advantages on real mobile devices with an AR-based 3D avatar application. We hope that this work would act as an ignition of efficient training methods for lightweight neural networks in 3D human pose estimation.

Chapter 5

Synthesizing Pseudo-2.5D Mixed Reality Content from Monouclar Videos

5.1 Overview

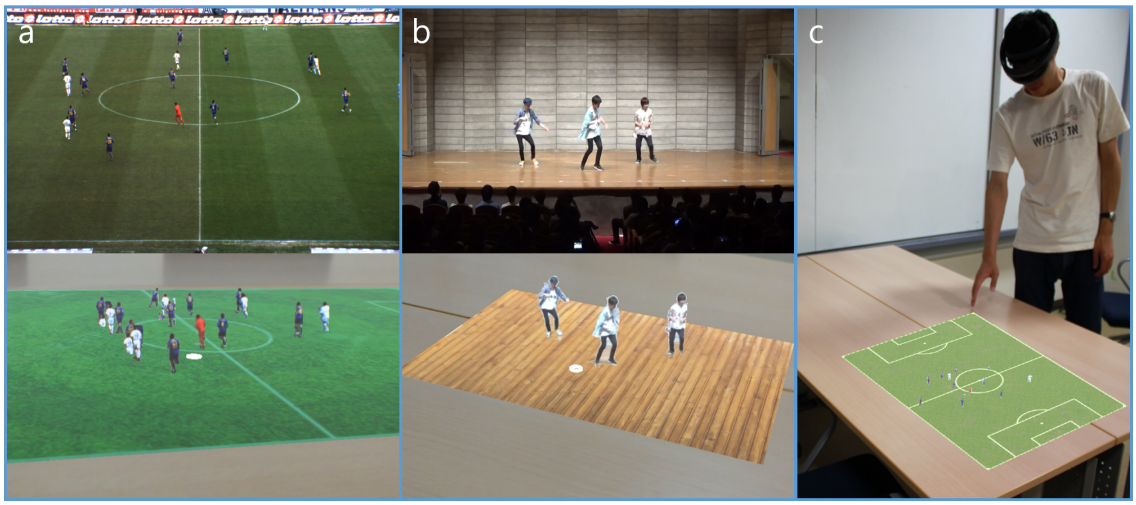


Figure 5.1: MonoMR enables users to easily synthesize pseudo-2.5D mixed reality content from monocular videos uploaded on the Internet or taken with common imaging equipment such as smartphones and cameras. With the MonoMR system, the user can create and experience immersive mixed reality content from various monocular videos, such as (**a**) sports broadcasting videos and (**b**) entertainment videos. (**c**) The synthesized content can be displayed in the real world through a mixed reality head-mounted display.

Mixed reality (MR) head-mounted devices (HMDs) are display devices that can overlay virtual content in the real world, and a user can watch it on a free viewpoint. Specifically, compared with 2D media, such as photos and videos, 3D content

synthesized from real-world objects has higher immersiveness. Hence, many methods for synthesizing MR content from real-world objects have been proposed [37, 103], and several systems are already commercialized [1, 2]. A common method in creating MR content is to place multiple monocular RGB or depth cameras around an object, synchronously capture the images, and then reconstruct 3D shapes of the object from the captured images. The content synthesized using this method is accurate and impressive and is utilized in various industry fields, such as sports broadcasting and entertainment. However, since most previous systems based on this method require multiple cameras and synchronization devices, configuring the system is complex, and the operating environment is restricted [53, 149, 103]. Thus, these factors make it difficult for end-users to use these systems. In addition, estimating the intrinsic and extrinsic parameters of the cameras from prerecorded videos is a challenging task. Some methods try to estimate the parameters using landmarks in images [109]. However, this limits the types of video that the method can process.

Nowadays, we can record monocular videos through smartphones and digital cameras, and a large number of various monocular videos, such as sports, entertainment, and daily life, have been uploaded to the Internet. If there is an easy way for end-users to create MR content from these videos, various MR content can be provided without any special equipment. Moreover, the created content can be shared freely, similar to the present video-sharing websites on the Internet. To explore the usability and feasibility of MR content made from monocular content, we propose MonoMR, a system for synthesizing MR content from monocular videos. MonoMR is an end-to-end system (Figure 5.1) that uses simple yet effective methods, different from the previous complex systems. In order to create MR content, first, the system detects people from a monocular video through a deep neural network (DNN), then calculates the homography matrix between real-world and image distances using an interactive user interface, and estimates pseudo-3D positions of the detected people. Next, person textures are extracted and placed at the estimated positions. Finally, the MR content is synthesized on these elements. The content synthesized by the proposed system is played on Microsoft HoloLens; the user can freely place the content in the actual world and view it from a free viewpoint. The paper's outline is summarized as follows:

- We propose MonoMR, a system to synthesize MR content from single or multiple monocular videos.
- We evaluate the quantitative performance of our system.

- We assess the impact of the synthesized content through a user study.
- We develop suitable sample applications using the proposed system.

5.2 MonoMR System

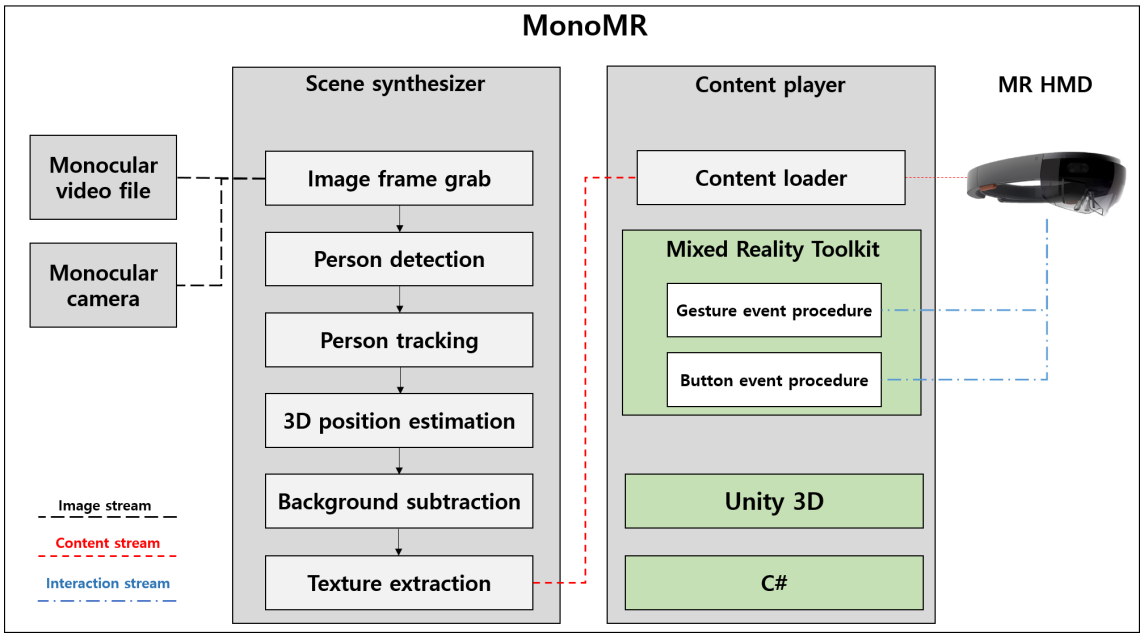


Figure 5.2: Configuration diagram of the MonoMR system.

As shown in Figure 5.2, the MonoMR system consists of a personal computer-based scene synthesizer to synthesize content and a HoloLens-based client player to play the content. This section describes how the scene synthesizer generates MR content from monocular videos, and the client player displays the generated scene.

5.2.1 Person Detection and Tracking

As the first step of the scene synthesizer, people in video frames are detected. This procedure was a very challenging problem in computer vision until a few years ago. However, recent dramatic advances in DNNs allow accurate person detection and their body keypoints in a monocular image. We use OpenPose [24], one of the state-of-the-art person detectors to detect persons and body keypoints from video frames.

Our system provides normal mode (656×368) and precision mode (1312×736) according to the input resolution of the network. In the normal mode, the network detects normal-sized human bodies. However, small people in the video cannot be detected because of the low input resolution. In precision mode, the network accurately detects small-sized human bodies with slow inference speed. Then, the bounding boxes are defined based on the detected keypoints.

Even if the system uses a state-of-the-art person detector, the results may contain false positives and false negatives depending on the quality of the video. In order

to detect persons robustly, each detected person should be continuously tracked in subsequent frames. In tracking an object, the system should solve the $X \in \mathbb{R}^{2\times k}$ assignment problem. Therefore, the following equation is minimized using the Kuhn-Munkres algorithm [83].

$$E = \sum_{m}^{M}\sum_{n}^{N} \|X_t^m - X_{t-1}^n\| \cdot C_{mn} \tag{5.1}$$

$$C_{mn} = \begin{cases} 1 & \text{Person } m \text{ assigned to person } n \\ 0 & \text{Otherwise.} \end{cases}$$

Here, X is a set of k-joints and M and N are the number of detected people in time t and $t-1$. After the assignment procedure, the moving average filter is applied to the coordinates of each object's bounding box to remove the jitter of each tracking object's trajectory. Moreover, the filter can estimate the undetected person's position based on previously observed values.

5.2.2 Pseudo-3D Position Estimation

In order to capture the depth information of humans in the real world, multi-view stereo vision, depth cameras, and DNNs have been used in existing systems. However, these systems require a complicated configuration or special equipment and are difficult to use. In this study, we propose a simple depth estimation method using the homography matrix [57], calculated based on the detected person's ankle position and minimal user interaction, as illustrated in Figure 5.3. The system receives the four vertices and approximate distance of the real world between the vertices in the first frame of a video from users. Then, a homography matrix H for mapping the image coordinate system i, j into the real-world x, z coordinate system is calculated. Then, the pseudo-3D position on the real world $X_t(x, z)$ is calculated using the following equation.

$$X_t(x, z) = H \cdot \mathcal{A}(X_t) \tag{5.2}$$

where $\mathcal{A}$ is the average position of ankles i, j in the X_t set. The moving average filter is applied to the calculated pseudo-3D position to remove the noise.

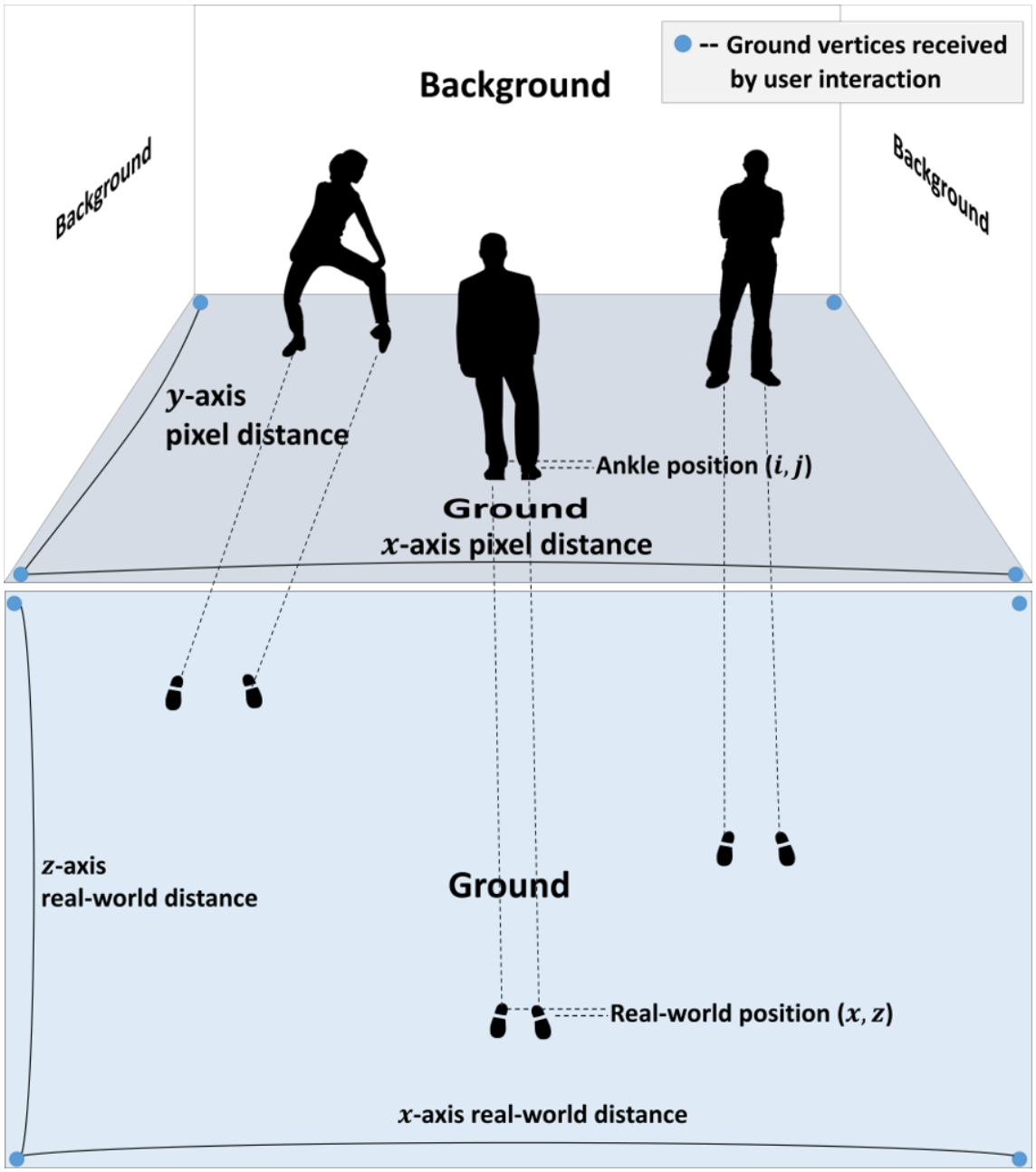


Figure 5.3: Proposed method for estimating the pseudo-3D position of a person in the image. The ankle position detected in the image coordinate system $(i,\ j)$ is mapped using a homography matrix to estimate the real-world coordinate system $(x,\ z)$.

5.2.3 Extracting Person Texture Using Background Subtraction

A segmentation procedure is performed to extract the texture of the detected person in the video. Graph cuts [16] and mask R-CNN [59] are the standard algorithms for the segmentation, however these algorithms have high computational complexity. We propose a simple method to extract the person's texture using a background subtraction algorithm for efficient texture extraction.

Given that most videos constantly change foreground and background, the foreground objects extracted with the vanilla background subtraction algorithm based on image difference do not have sufficient quality. We use a k-nearest neighbor (KNN)-based background subtraction method [162], one of the Gaussian mixture model (GMM)-based methods. GMM-based algorithms are robust to repeated, slow motion, and constantly changing lighting conditions. Thus, these algorithms can be used in most videos that have constantly changing foregrounds and backgrounds. Notably, the KNN-based background subtraction algorithm automatically updates the parameters in real-time and selects only the components required by each pixel. Therefore, the processing time of the KNN-based algorithm is reduced compared with the existing GMM algorithms, even with better quality. In addition, the background image without foreground objects can be acquired with this algorithm, and this image can be utilized in the generated content.

The foreground image extracted by the KNN method may contain noises and holes. In order to remove them, morphological operations are applied to the foreground image. After noise removal, only the moving objects' textures are extracted by masking with the bounding boxes. Figure 5.4 shows the results of the background subtraction procedure.

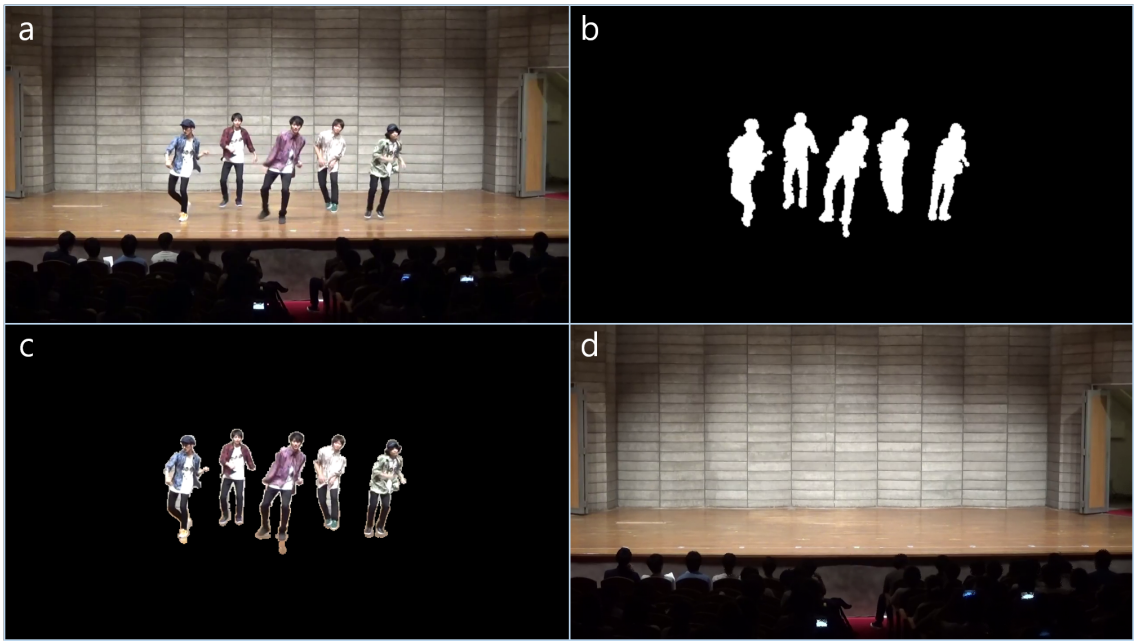


Figure 5.4: Results of the texture extraction procedure. (**a**) Input image, (**b**) foreground mask, (**c**) foreground segments, and (**d**) background image.

5.2.4 Texture Size Correction Using Weak-Perspective Projection and Content Synthesis

If the extracted texture is directly applied to the scene, then the size of the same person is different according to the position of the perspective. To minimize this perspective distortion, we use a weak-perspective projection-based correction method. First, the pixel per meter in the real world at the corresponding position of the person is calculated from the homography matrix H and the position of the image coordinate system $X_t(i, j)$. Then, the texture size is recalculated, and the distortion is corrected as shown in Figure 5.5.

Finally, the corrected textures are placed in the 3D world based on the pseudo-3D position of each texture, the extracted background image or a custom image is set to the ground texture, and then the MR content is synthesized.

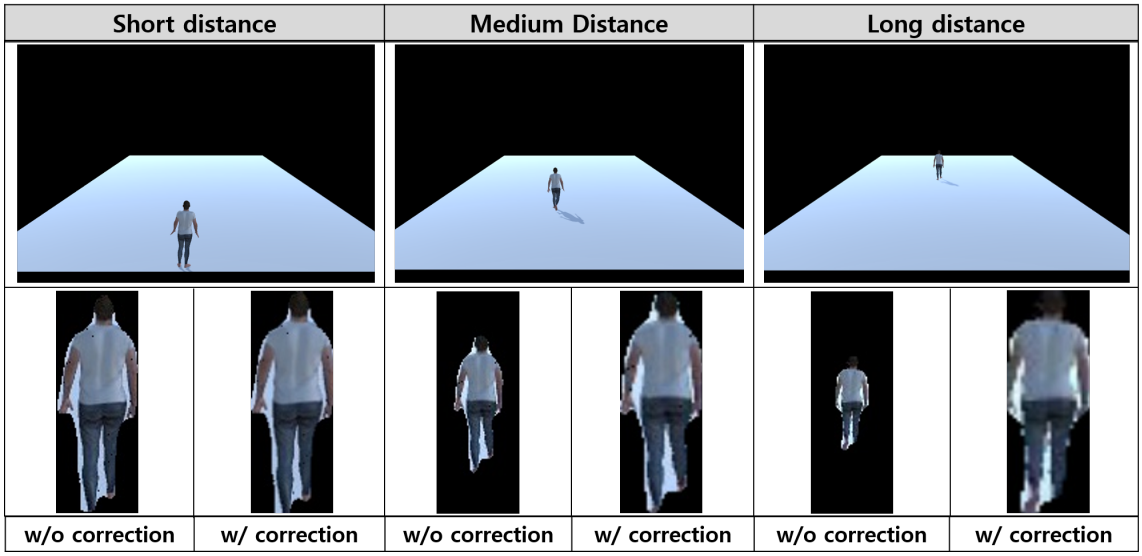


Figure 5.5: Result of the texture size correction.

5.2.5 Billboard Rendering

The MR content is synthesized by extracting the textures from image frames of a monocular video. Therefore, the camera's viewpoint is fixed, and we cannot obtain the information not captured in the original video (e.g., an information loss on the part not facing the camera), and the user notices the unnaturalness when the viewpoints of the camera and user are different (See Figure 5.6a).

In order to address this problem, the system has a function that applies billboard rendering [6] to every texture, as shown in Figure 5.6b. Billboard rendering is a simple technique in which the textures are rotated toward the user's viewpoint,

thereby reducing the unnaturalness of placing 2D textures in a 3D space. With this function, even if the user changes their viewpoint from the original camera's viewpoint, they notice less unnaturalness.

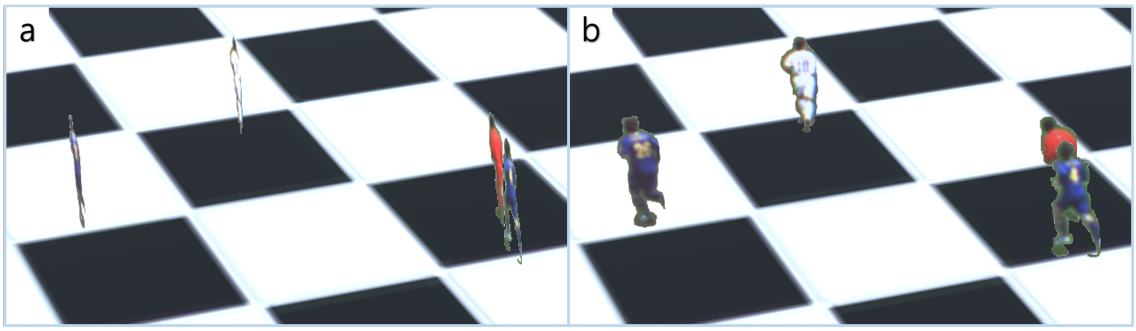


Figure 5.6: Result of billboard rendering. (**a**) Billboard rendering disabled and (**b**) billboard rendering enabled.

5.2.6 Playing Synthesized Content on MR HMDs

The generated content is played using a client player application. The client player is based on Unity and Mixed Reality Toolkit [1] and runs on the HoloLens MR HMD. The content is displayed in the real world, and users can control the playback, pause, and billboard rendering functions and activation/deactivation through buttons.

Given that the content consists of a minimal number of polygons, rendering and producing multiple textures and polygons in 30 fps even using a standalone MR HMD, which has limited processing power, is possible. The user can enjoy the MR content while changing their position and viewpoints freely, and the content can be placed or resized in the real world through user gestures, as shown in Figure 5.7.

[1]https://github.com/Microsoft/MixedRealityToolkit-Unity

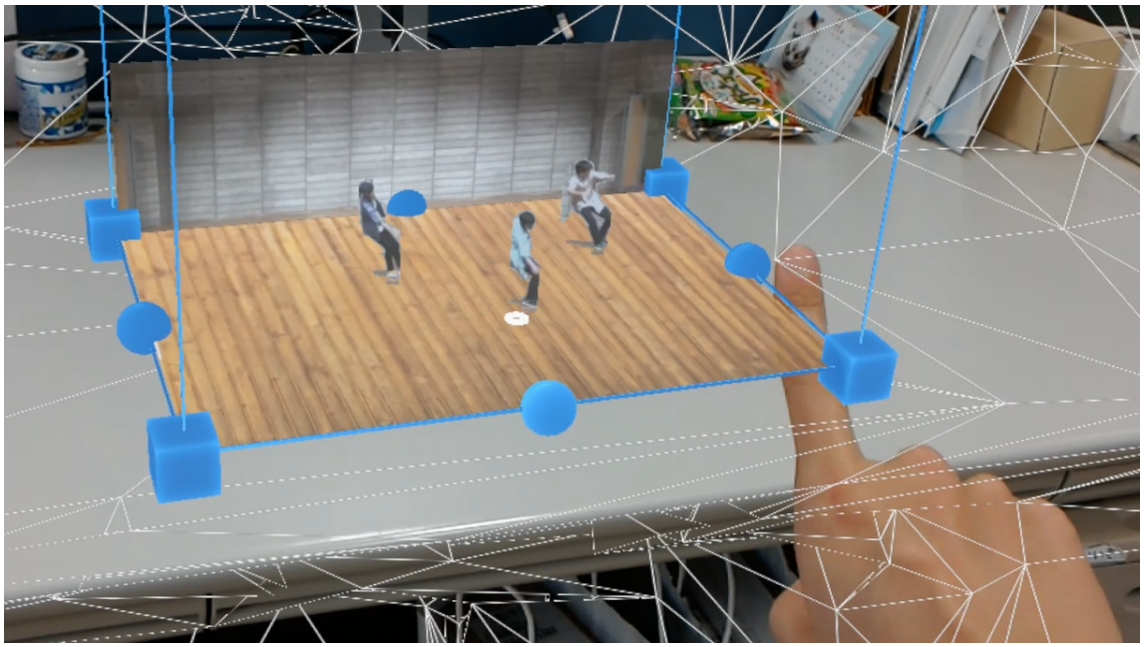


Figure 5.7: Display of the generated content in the real world.

5.3 Performance Evaluation

In this section, we performed performance assessments on the depth estimation accuracy, quality of textures, and processing speed, factors that directly affect generated content's quality.

5.3.1 Accuracy of Depth Estimation

First, we evaluated the depth accuracy of the proposed method. We perform performance evaluations based on two capturing scenarios (small space and large space).

In the case of a small space, the subject freely walked in a square, with a space of approximately 1.7 m in width and approximately 3 m in height, and a Kinect was used to obtain the ground truth data. Five subjects (two female) were the participants in the preparation of the ground truth set. We obtained 600 frames of full HD images and depth information for each subject. The mean absolute error between ground truth and estimated results is 24.57 cm, and the results for each subject are shown in Figure 5.8.

In the case of a large space, it's not easy to acquire the accurate ground truth from a real-world scene. Therefore, we rendered some synthetic ground truth composed of 3000 frames of images and the depth information using computer graphics. The mean absolute error between ground truth and estimated results is 76.04 cm, and the results for each distance section are shown in Figure 5.9.

At a distance of less than 20m, depth errors were less than 1m. As shown in Figure 5.9, the error increased as the model moved away from the camera because as the distance of the ground truth image increases, the number of pixels that represent the same distance decreases. Thus, the error increases as the quantization error increases.

Figure 5.8: Mean absolute error of depth estimation for each subject in a short-range space.

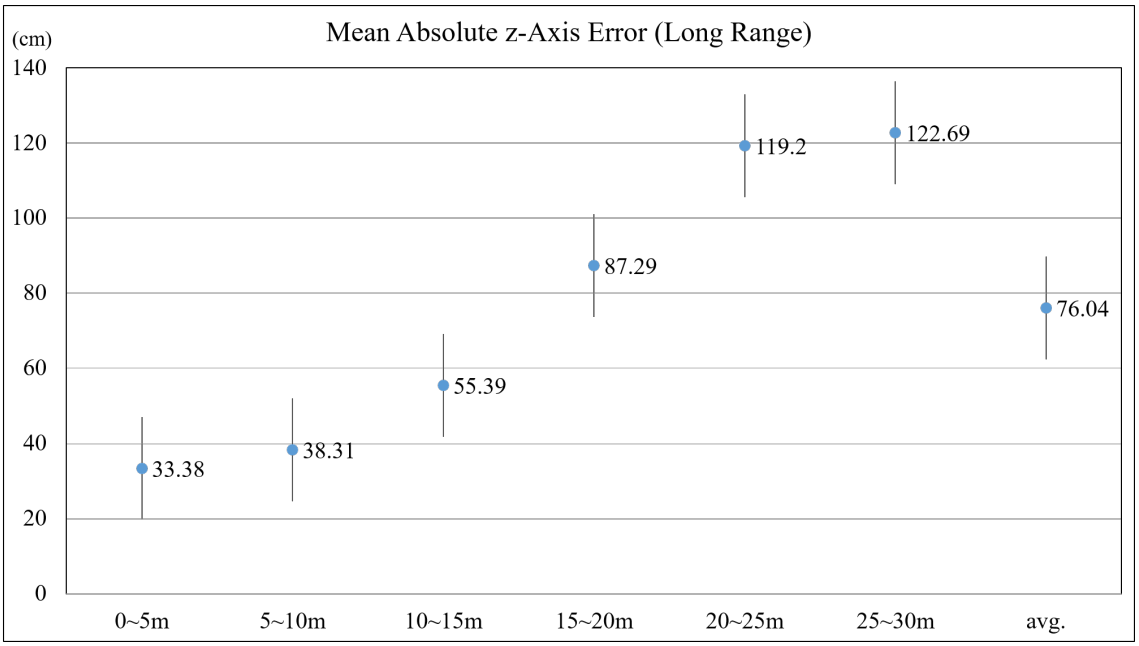


Figure 5.9: Mean absolute error of depth estimation for each distance section in a long-range space.

5.3.2 Accuracy of Texture Extraction

We evaluated the accuracy of the person texture extraction method applied to MonoMR. We created ground truth data for 500 images through the mask R-CNN [59], one of the state-of-the-art algorithms of segmentation, and measured the mask intersection over union (IoU). Consequently, the obtained average mask IoU is 0.72 (sd = 0.05). Figure 5.10 is the visualization result of extracting texture through the mask R-CNN and the proposed method.

The mask image obtained through our method generally does not have significant artifacts. However, compared with the mask obtained through the mask R-CNN, the results of our method may contain noise such as shadow and missing body parts. Nevertheless, given that the proposed method has more advantages than the mask R-CNN regarding processing speed, we applied the background extraction-based texture extraction method in this study to enhance the possibility of real-time processing.

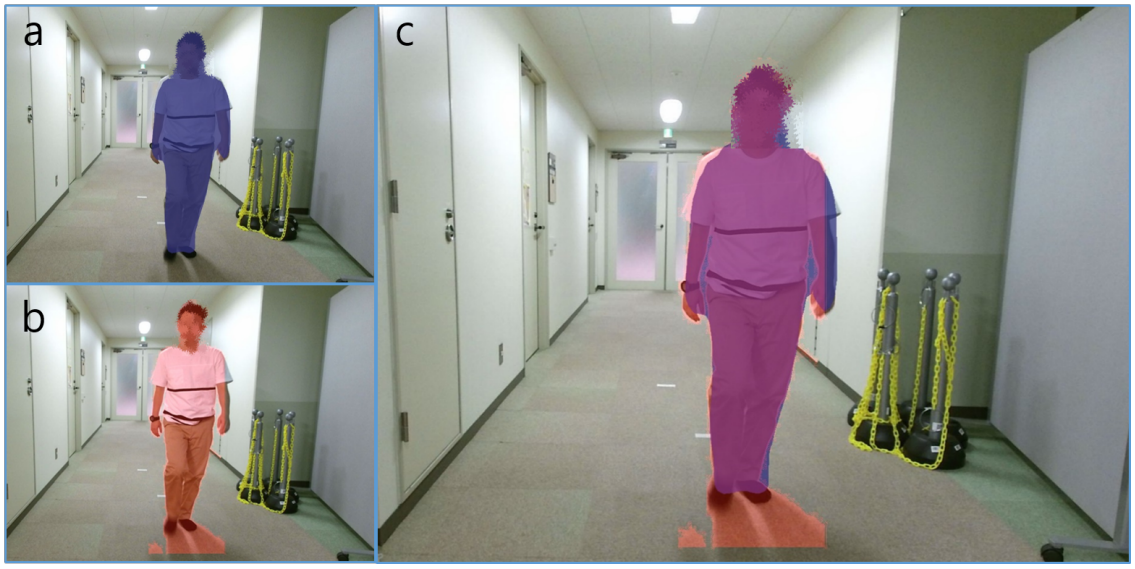


Figure 5.10: Visualization results of texture extraction methods. (**a**) Mask R-CNN (blue region), (**b**) ours (red region), and (**c**) overlapped two methods (purple region is the intersection area).

5.3.3 Processing Speed

We measured the processing time of MonoMR to synthesize the content. All experiments were conducted on a desktop with a Core i7 processor (4-core, 8-thread), 16 GB RAM, and GTX 1080 with 6 GB memory. We used the video sequence (full HD resolution) of ISSIA-CNR [42] (soccer video dataset) as the synthesizing target and measured the average processing time to generate the content from 500 frames of the video. The results are shown in Table 5.1.

The system consumed approximately 179 ms per frame to generate the content in normal mode; hence, we can confirm that the proposed system generates images at a processing speed of approximately 5fps (the precise mode for detecting small people has a performance of two frames per second). In addition, the processing speed of the mask R-CNN and proposed texture extraction method was measured, and the results are listed in Table 5.2. The proposed method extracts textures at a very high speed compared with mask R-CNN, and it can increase the possibility of real-time processing. Based on these results, we confirmed that MonoMR could generate MR content at a relatively fast speed because all the procedures are not mainly composed of DNNs but use a combination of common image processing algorithms.

Procedure	**Time (ms)**
Person detection with normal / precision modes	137.83 / 406.94
Person tracking	0.07
Pseudo-3D position estimation	0.03
Texture extraction with background subtraction	41.43
Total processing time with normal / precision modes	**179.36 / 448.47**

Table 5.1: Processing time for each procedure.

Procedure	**Time (ms)**
Texture extraction with background subtraction (Ours)	**41.43**
Texture extraction with mask R-CNN	2545

Table 5.2: Processing time of two texture extraction methods.

5.4 Small-scale User Study

In order to measure the effectiveness of the synthesized content, we conducted a user study with a two-by-three design with the two independent factors content types and display types. Twelve participants (three females; mean age=26, SD=9.23) volunteered in our experiment, and they had no experience using MR and Virtual Reality (VR) devices.

5.4.1 Experiment Design

We attempted to confirm the effectiveness of the content by qualitatively evaluating the following items:

- Depth perception: How much of a stereoscopic degree the user feels in the content.
- Immersiveness: How immersed is the user in the content.
- Attractiveness: How interested is the user in the content.

We used two types of content, sports broadcasting (soccer) and entertainment (dancing), for the experiment. The comparison conditions are as follows:

- C1: Monocular videos displayed on a flat-panel display.
- C2: Monocular videos displayed on a MR HMD.
- C3: Synthesized content displayed on a MR HMD.

Category	Question
Depth perception	Q1. It was easy to recognize a visual-depth of the sports content.
	Q2. It was easy to recognize a visual-depth of the entertainment content.
Immersiveness	Q3. It was easy to immerse in the sports content.
	Q4. It was easy to immerse in the entertainment content.
Attractiveness	Q5. It was attractive to watch the content with this system.

Figure 5.11: Questions used in the user study.

Each subject experienced each comparison condition in a random order, and the evaluation was performed using a 5-point Likert-based questionnaire sheet (where 1 = Strongly Disagree to 5 = Strongly Agree). The detailed questions are listed in Figure 5.11.

5.4.2 Results

Depth perception

First, we performed the Friedman test [38] for multiple comparisons to assess the differences between the group means in the experimental results. As presented in Table 5.3, the test result shows a significant difference in the subject's assessment depending on the method. We conducted the Wilcoxon signed-rank test [141] as post hoc analysis to identify which factors have a significant difference, and the result is illustrated in Figure 5.12. In order to verify whether the randomization controls order effects, the two-way ANOVA test [47] was used to check whether there were significant differences in assessment results between the ordering groups. The ANOVA test shows p-value $= 0.163$ for sports broadcasting and p-value $= 0.589$ for entertainment content, which means that the randomization works well because no significant differences were detected across the ordering conditions.

Based on the post hoc test result, the user can recognize the spatial information through the proposed method C3 more easily than C1 and C2 ($p \leq 0.001$). Hence, we assumed that the content generated by the proposed system allows the subjects to perceive the depth information of the content based on positive feedback. This result showed that the user could feel an improved stereoscopic effect on the content created by the proposed system compared with other experimental conditions.

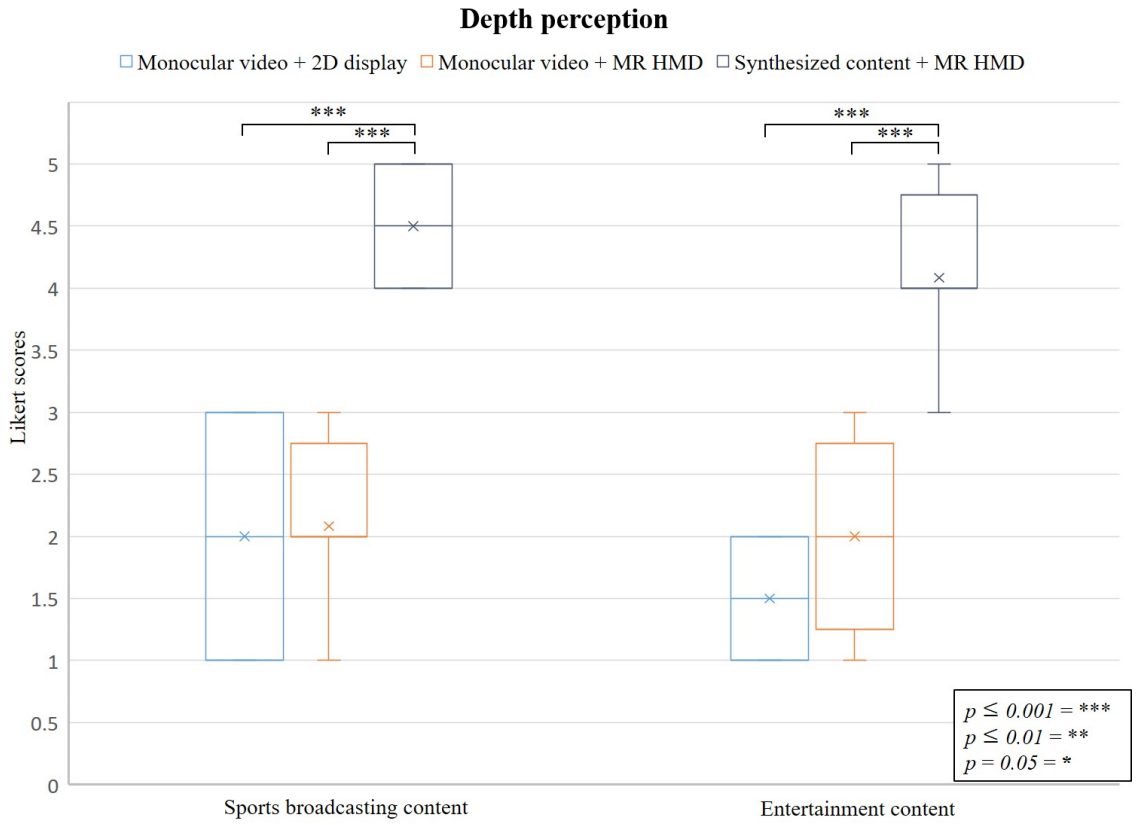


Figure 5.12: Evaluation of the depth perception for each condition.

Source	Sum Sq.	d.f.	Mean Sq.	Chi Sq.
Method (*)	315.65	2	157.82	46.87
Content	3.56	1	3.56	2.95

*: $p \leq 0.05$

Table 5.3: Friedman test table of subjective score with depth perception.

Immersiveness

The results of the Friedman test and Wilcoxon post hoc analyses are shown in Table 5.4 and Figure 5.13, respectively. We can observe a significant difference in the results of the method. The ANOVA test shows p-value = 0.605 and 0.389 for sports broadcasting and entertainment content, respectively, which means that the randomization is valid.

Specifically, a meaningful difference also exists between C3 and other conditions and between C2 and C1. The users responded that C2 was more immersive than C1($p \leq 0.01$) because the size of the virtual screen of C2 was more significant than the physical screen of C1. The subjects evaluated the proposed method (C3) as the most immersive method ($p \leq 0.001$). We assumed that the content generated by MonoMR could be watched at a free-viewpoint, and this feature affects the immersiveness of users. Based on these results, we confirmed that the proposed system could increase the immersiveness of monocular content.

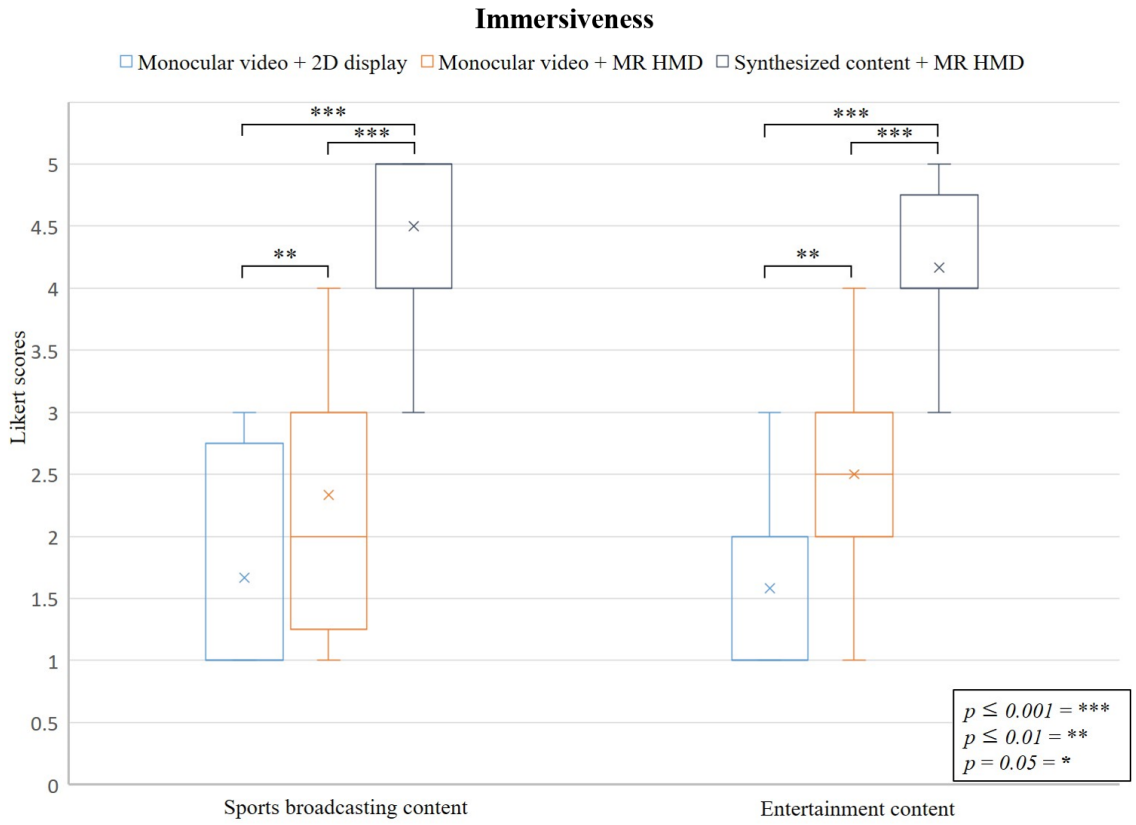


Figure 5.13: Evaluation of the immersiveness for each condition.

Source	Sum Sq.	d.f.	Mean Sq.	Chi Sq.
Method (*)	166.54	2	83.27	24.17
Content	0.13	1	0.13	0.1

*: $p \leq 0.05$

Table 5.4: Friedman test table of the subjective score with immersiveness.

Attractiveness

Attractiveness was comprehensively evaluated regardless of the content type. The Friedman test and Wilcoxon post hoc analysis results are shown in Table 5.5 and Figure 5.14, respectively. The ANOVA test shows p-value = 0.319, and the ordering groups don't affect assessment results.

Subjects responded that C2 was more interesting than C1 ($p \leq 0.01$). Based on the users' verbal feedback, we confirmed that the reason is that the video player's size in MR HMD can be freely adjusted according to the context of the content.

The subjects reported that C3 provided the most exciting experience among the methods ($p \leq 0.001$). Apart from the questionnaire, we asked the subjects which factor most increased the attractiveness of the synthesized content. Seven subjects answered the improved depth perception, and five subjects answered the advantage of free viewpoint.

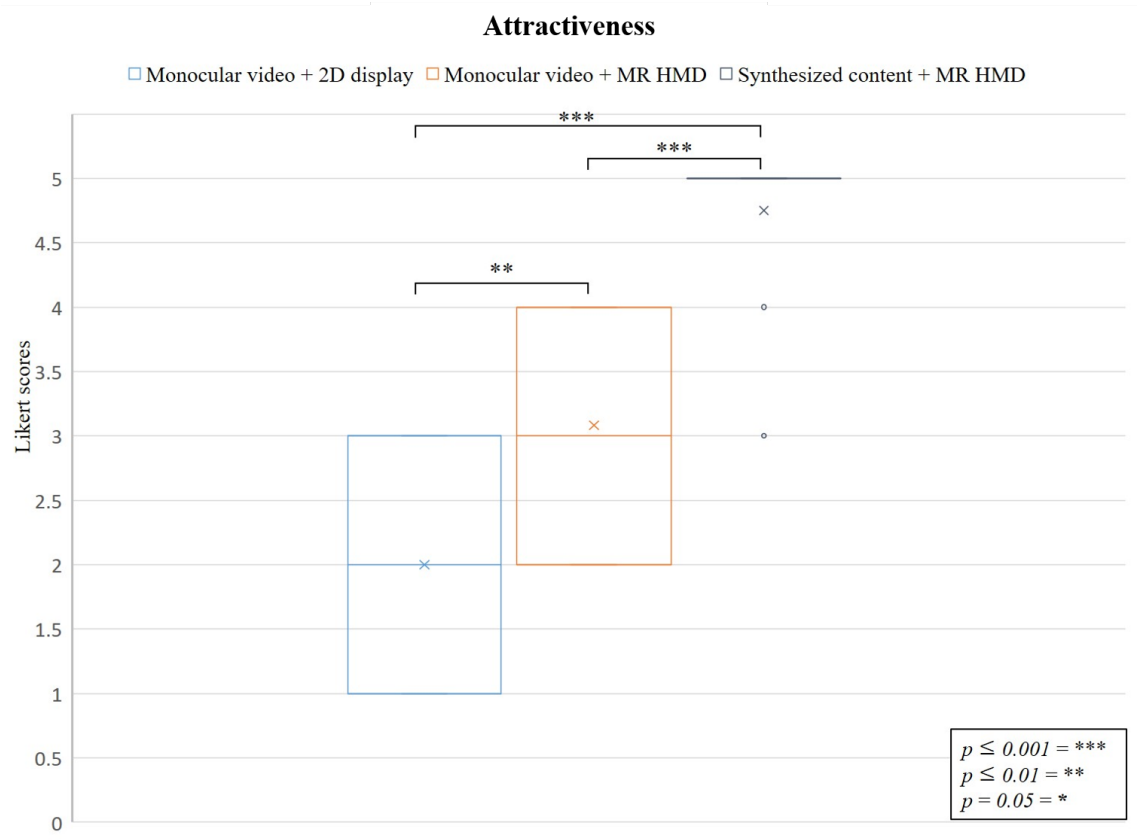


Figure 5.14: Evaluation of the attractiveness for each condition.

Source	Sum Sq.	d.f.	Mean Sq.	Chi Sq.
Method (*)	166.54	2	83.27	24.17

*: $p \leq 0.05$

Table 5.5: Friedman test table of the subjective score with attractiveness.

5.5 Applications

In order to demonstrate the applications of MonoMR, we synthesized prototype content from various monocular videos. Because the proposed system can generate content from monocular videos with small environmental constraints, it can be applied to various fields, as shown in Figure 5.15.

5.5.1 Immersive Sports Broadcasting

Sports broadcasting is one of the areas where free-viewpoint video systems are most actively applied. Users who watch sports broadcasts want to watch the game from a different viewpoint; hence, systems such as Eye Vision [3] have been applied to sports broadcasting to meet these demands. However, most of the existing methods are difficult to set up because these methods require many cameras to be placed around the target and are complicated to use for users who are not experts.

Because the MonoMR system can easily synthesize the MR content from a single monocular camera, the user can create sports content from 2D videos, view the content with a free-viewpoint, and recognize the 3D positions of players intuitively (Figure 5.15a). In addition, given that the system can generate a single scene from multiple monocular videos, capturing a large stadium, which is difficult to capture using a single camera, is possible by dividing the capture area into multiple cameras and merging the videos into a single content.

Figure 5.15: Application examples of the proposed system. (**a**) Sports broadcasting and (**b**) entertainment content provide improved stereoscopic effect and an immersive feeling to users than original monocular videos. (**c**) Surveillance systems based on MonoMR allow users to easily recognize situations and spatial information of multiple cameras.

5.5.2 Dynamic Entertainment Content

Entertainment content, such as performance and theater, is also one of the areas where the proposed system can be utilized. In the case of performance and theater content in DVD and Blu-ray media, the videos have been recorded from various viewpoints. Hence, it allows the user to select and enjoy scenes from a specific viewpoint. As described in the experiment section, the entertainment content created by our system is more attractive than a monocular video displayed on a flat-panel display. Therefore, with our system, the user can enjoy the content while freely changing their viewpoint (Figure 3.17b).

5.5.3 Effective Surveillance System

The existing surveillance systems display videos through a single display with divided windows or multiple displays. However, these methods are complex for a user to observe the plurality of screens simultaneously. Mainly, recognizing the place displayed on the monitor is not intuitive. In order to address this problem, a system [43] has been proposed where multiple camera images are synthesized into a single 360-degree image and displayed to a VR HMD. However, the entire scene cannot be recognized within a limited field of view of the HMD, and it's hard to recognize the depth of the target object.

Constructing an efficient surveillance system with the MonoMR is possible because the proposed system can synthesize a single scene from several monocular cameras. We synthesized the four surveillance videos of CMUSRD [58] into a single scene using MonoMR as the demo application of the surveillance system (Figure 5.15c). The user can intuitively perceive a place where a specific target is located. In addition, even if the target moves from the camera viewpoint to another camera's viewpoint, the user can track the moving target intuitively. Since the system can synthesize content from various kinds of videos, there are many other potential applications besides the ones proposed.

5.6 Discussion

We present the MonoMR system that generates the pseudo-2.5D MR content from monocular camera videos. The system can render MR content from a large number of previously captured videos of various types. Our system does not require special imaging equipment, such as multiple monocular cameras or depth cameras, similar to most conventional systems and complicated settings, such as the synchronization among the cameras. In addition, because the proposed system can generate MR content using a single monocular camera with minimal user interaction, this has higher usability than any previously proposed systems. Therefore, users without expert knowledge can easily create MR content.

Our system consists of not only a DNN but also uses typical image processing algorithms. Based on the experimental results, we confirmed that our proposed method has reasonable performance for depth information estimation and texture extraction required for producing MR content from monocular videos. In addition, the proposed method is processed relatively high speed, except for the DNN process. Therefore, if a high-performance GPU is used and parallel optimization for image processing is applied, the proposed system can reach real-time performance.

We conducted the small user study, and the results show the feasibility of converting existing monocular videos into more exciting and immersive content with the proposed system. Although the MonoMR system has good performance and usability, we describe some technical challenges and limitations of the system based on the conducted experiment and system implementation.

Limited camera posture. MonoMR does not use a global motion compensation algorithm and the camera posture estimation method using specific landmarks [148, 109] because of high computation and low generalization capacity. Therefore, the input video's viewpoint, which is converted into MR content, should be fixed.

Pseudo-3D position. The system estimates the person's x and z positions based on the correlation between the ankle position of the human and the ground. However, if the subject moves on the y-axis, such as jumping and tumbling, it's difficult to estimate the correct 3D position. To address this problem, we will attempt to apply the global depth estimation DNN [45, 51] to our system.

Texture quality. MonoMR extracts textures using a background subtraction algorithm to increase the entire processing speed. As can be observed from the previous experiments, MonoMR extracts person textures with acceptable quality. However, if the texture quality reduces due to a detection failure of the human

detector or drastic illumination changes in the capturing environment (Figure 5.16), then some artifacts could be observed.

Figure 5.16: Artifacts of the extracted textures and content caused by abrupt illumination change, non-detection of body parts, and overlapping people.

Mask R-CNN, the current state-of-the-art algorithm, shows excellent quality; however, as mentioned in Section 5.3.2, it has drastically reduced the system's entire processing speed. Therefore, this method has not been applied in this study. If a segmentation network with good accuracy and performance is proposed, then we will consider applying the network to our system.

2D texture. The system displays 2D textures instead of 3D mesh models. Hence, it's difficult for the human vision to recognize that the generated model is planar if the user views the content at a certain distance [123]. However, if the user views the content at a short-range, then they feel the unnaturalness. Therefore, we are considering applying methods to recover the information not captured by the camera, such as generative adversarial nets [52] and 3D mesh recovery network [31], as future work.

Large content size. The content consists of the human textures and location data of each frame. The size of the content is large because it does not use any compression method for the real-time operation in low I/O performance of the standalone HMD (e.g., the size of the 30 s of soccer content is approximately 200 MB). We expect that if the HMD's network bandwidth and I/O performance are increased and the texture compression method is applied, then real-time streaming could be applied without difficulty.

Small-scale user study. We conducted the user study with a small subject group. Because of the small number of subjects in the experiment, the statistical power is insufficient to prove significance. A post hoc power analysis revealed that the effect size and statistical power observed in the user study are 0.74 and 0.75,

respectively. Therefore, more than 14 participants would be required to obtain statistical power at the recommended 0.80 level [36]. We plan to conduct a user study with a large number of participants and perform accurate statistical analysis to prove the usability of the system.

Despite the limitations, MonoMR is a potentially powerful system in which anyone can easily convert monocular videos into immersive and exciting MR content. To the best of our knowledge, no system that can convert videos of various genres captured using a single monocular camera into MR content has been proposed yet. In addition, it will be possible for a DNN to be developed and applied to the system to produce better quality content when the hardware limitations have been addressed.

5.7 Conclusions

This paper presents MonoMR, the system synthesizing the pseudo-2.5D MR content from monocular videos for MR HMD. Our approach can generate MR content from only a single monocular camera or many different videos uploaded on the Internet. In addition, the system requires only minimal user interaction during content creation, and end-users without expertise can easily use this system. Users can enjoy the synthesized content at a free-viewpoint through MR HMD and freely arrange and adjust contents via hand gestures.

We confirmed that the generated content is more immersive and attractive than the original monocular video through user studies. Based on these evaluations, we believe that the proposed system converts a lot of existing monocular content into MR HMD optimized content. We hope that the proposed system will contribute to the distribution of MR content regarding the increase in content demands owing to the commercialization and expansion of MR HMD.

Chapter 6

Conclusions

6.1 Summary

This thesis presents three contributions that capture human motion and generate visual content from monocular videos. We argue that a solution for capturing and visualizing various human motion information is by exploiting multiple cameras and specialized machine learning algorithms. It is hard to escape from the "Curse of Cameras" with this conventional approach.

We directly tackle the "Curse of Cameras" by leveraging new concept hardware, machine learning techniques, and interactive methods. We propose the system that captures multimodal motion information such as the 3D motion of the camera wearer and surrounding people and the user's viewport using an ultra-wide fisheye camera (Chapter 3). We demonstrate the lightweight motion capture network that can operate in real time on devices with limited computing resources and an efficient training strategy (Chapter 4). Finally, we build the end-to-end system that generates pseudo-2.5D mixed reality content and visualize it (Chapter 5). Our contributions have realized the framework that enables 3D content creation with a single monocular camera by replacing the existing complex 3D content creation framework, promoting user-based XR content creation.

6.2 Future Work

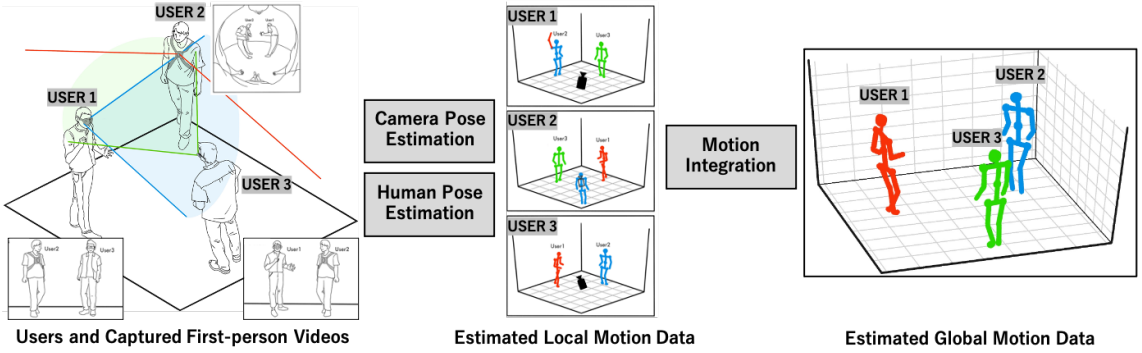


Figure 6.1: Conceptual diagram of the virtual motion capture system on the real world.

As mentioned in the introduction, the ultimate goal of this study is to establish a virtual motion capture studio. In order to realize this goal, numerous monocular cameras in the real world should be connected to others, and the captured motion information is shared with each device to provide a metaverse experience to users. We called this concept "Hyper Connectivity." In future work, each motion capture and visualization method proposed as an individual component in this thesis is connected to create a virtual motion capture studio. In order to achieve our objective, key future directions are summarized below.

6.2.1 Estimating the camera's position and pose in the world coordinate system

Implement an algorithm to calculate the position and pose information of each monocular camera in the world coordinate system. We estimate each camera's location and geometry information using SLAM, a method of estimating camera location and geometry information from images in real time. In order to synchronize each camera's estimated information, timestamp data of the GPS embedded on the devices is used. By matching each synchronized topographical information into a single map, the poses and positions of the cameras in the world coordinate system are calculated.

6.2.2 Estimating human motion with various camera parameters

We design and implement an optimal DNN that reflects the characteristics of various camera lenses and estimate 3D human motion and the relative distance from the camera. The network is designed based on a heatmap regression method widely used for human pose estimation and can determine the presence of each body joint as a probabilistic score. In the case of a camera with a wide field of view, the network is designed to estimate the pose of others and the camera wearer's pose. Since it is inefficient to train the network corresponding to a new lens parameter from scratch every time, meta-learning is used to enable the network to adapt to new lens parameters with a tiny number of samples. It takes a lot of time and cost to create a dataset by combining various lenses, environments, clothes, and motions in the real world. Therefore, in this study, multiple types of data required for DNN training are synthesized using computer graphics, and the synthesized dataset is used for training the network.

6.2.3 Integrating local human motion information

Since the user's motion information estimated by the DNN is on the local coordinate system, we implement an algorithm that converts the estimated motion data through each camera into the world coordinate system and then integrates it. The motion data on the local coordinate system is converted into the world coordinate system using the estimated position and pose of each camera. Since converted motion data may have duplicates, the probability score for each joint of the network and various features obtained from multiple viewpoints are composed in a graph, and then duplicated motion information is removed through dynamic programming. Finally, we get integrated motion information.

Because the proposed concept can overcome the limited capture capacity of a single device through interaction between multiple devices, this enables global motion estimation and 3D mesh estimation of real-world objects based on semantic segmentation and social signal sensing using other media such as audio.

Appendix A

Background Knowledge

Our methods use a fisheye lens model, depthwise separable convolution, and knowledge distillation. In this chapter, we describe the background knowledge about these.

A.1 Camera Model

This section presents essential knowledge of an ultra-wide fisheye camera model related to our proposed methods.

A.1.1 Ultra-wide Fisheye Camera Model

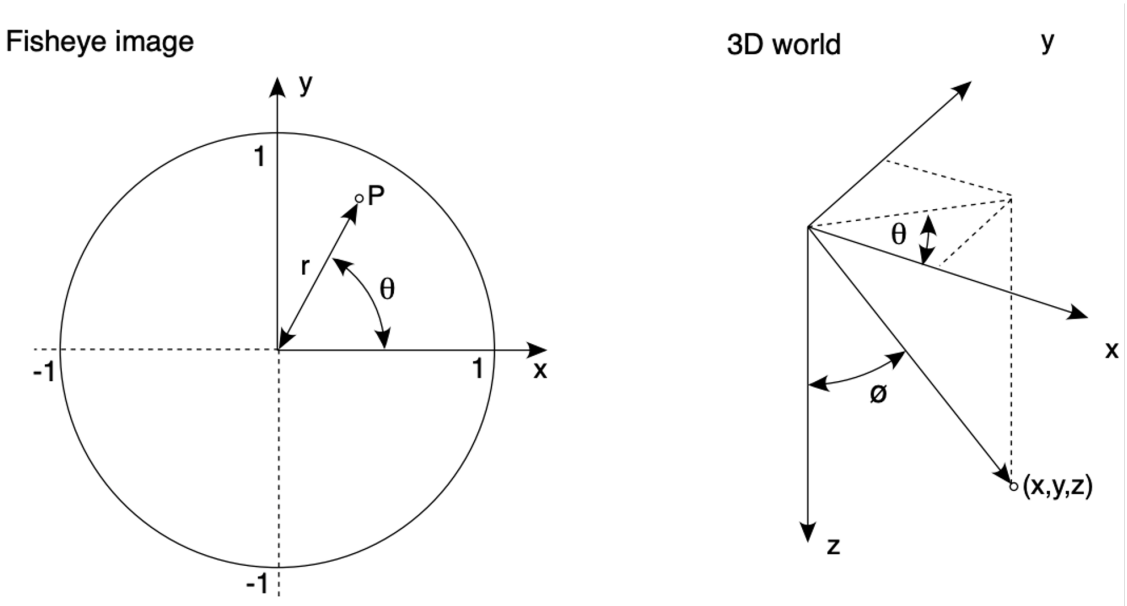


Figure A.1: Illustrate of the linear fisheye model.

The method proposed in chapter 3 uses an ultra-wide camera. We briefly describe mathematics for the fisheye lens here. In a linear (mathematical) fisheye, any point

P defines a longitude and latitude angle, and therefore a 3D vector into the world as illustrated in Figure A.1. A 1D correction polynomial can be used to convert any real fisheye into a linear fisheye as the following equation.

$$\begin{aligned} \theta = longitude = atan2(P.y, P.x) \\ \phi = longitude = r\phi_{max}/2 \\ X = sin(\phi)cos(\theta) \\ Y = sin(\phi)sin(\theta) \\ Z = cos(\phi) \end{aligned} \tag{A.1}$$

ϕ means the field of view of the fisheye lens.

However, most of the fisheye lenses have a non-linear relationship between r and ϕ. This non-linearity is most noticeable towards the fisheye's periphery, resulting in a compression artifact. The data points relating r to latitude are fitted with a suitable polynomial to correct the non-linear relationship. For the latitude ϕ, a general function could be

$$\phi(r) = a_0 + a_1 r + a_1 r^2 + ... + a_n r^n \tag{A.2}$$

Because the fisheye is assumed to be radially symmetric, and $r = 0$ denotes the fisheye's center, which corresponds to a latitude of 0, a_0 is 0. The highest-order polynomial with a fourth-order (n=4) is required in practice. Therefore the polynomial for a least-squares fit as the following equation.

$$\phi(r) = a_1 r + a_2 r^2 + a_3 r^3 + a_4 r^4 \tag{A.3}$$

A normalized radius r on the fisheye image gives the true latitude ϕ. This equation is used when mapping individual points from the fisheye image into the 3D point, and Figure A.2 shows the fitted polynomial of the Entaniya M12 280 fisheye lens, which is used in our research.

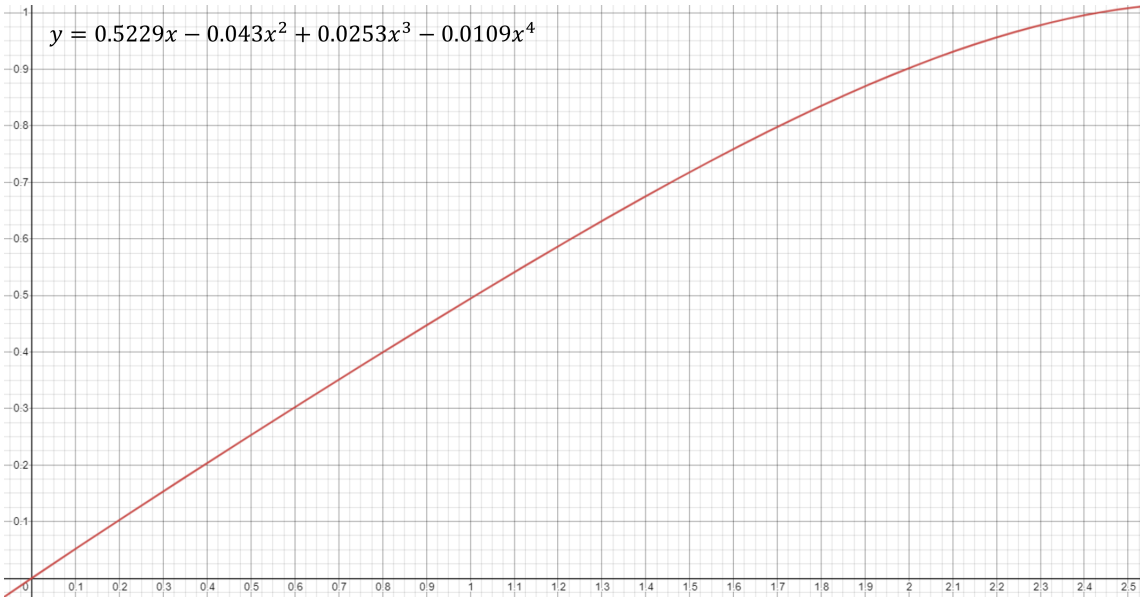


Figure A.2: Fitted polynomial of the Entaniya M12 280 fisheye lens.

A.2 Deep Neural Networks

A.2.1 Depthwise Separable Convolution

Even though CNN requires fewer parameters and computations than MLP, it is still not suitable for mobile devices with limited computational capability. In order to address this problem, a depthwise separable convolution layer has been proposed [33]. In depthwise separable convolution, two steps are generally applied to extract an activation map as illustrated in Figure A.3. The first step is a depth-wise convolution, and it is performed over each channel of a tensor. The second step is a point-wise convolution, and it projects the output tensor by the depth-wise convolution onto a new channel space. For the point-wise convolution, 1×1 kernel is usually used.

Here is an example showing how efficient depthwise-separable convolution is compared to conventional convolution through convolution arithmetic. To extract a single activation map from a common convolution layer, $KernelSize^2 \times InputChannel \times OutputChannel$ matrix multiplications are required. In depth-wise separable convolution, $KernelSize^2 \times InputChannel$ matrix multiplications are required for a depth-wise convolution, and $InputChannel \times OutputChannel$ matrix multiplications are required for a point-wise convolution. Therefore, we can dramatically reduce the computation and memory requirements for the convolution operation.

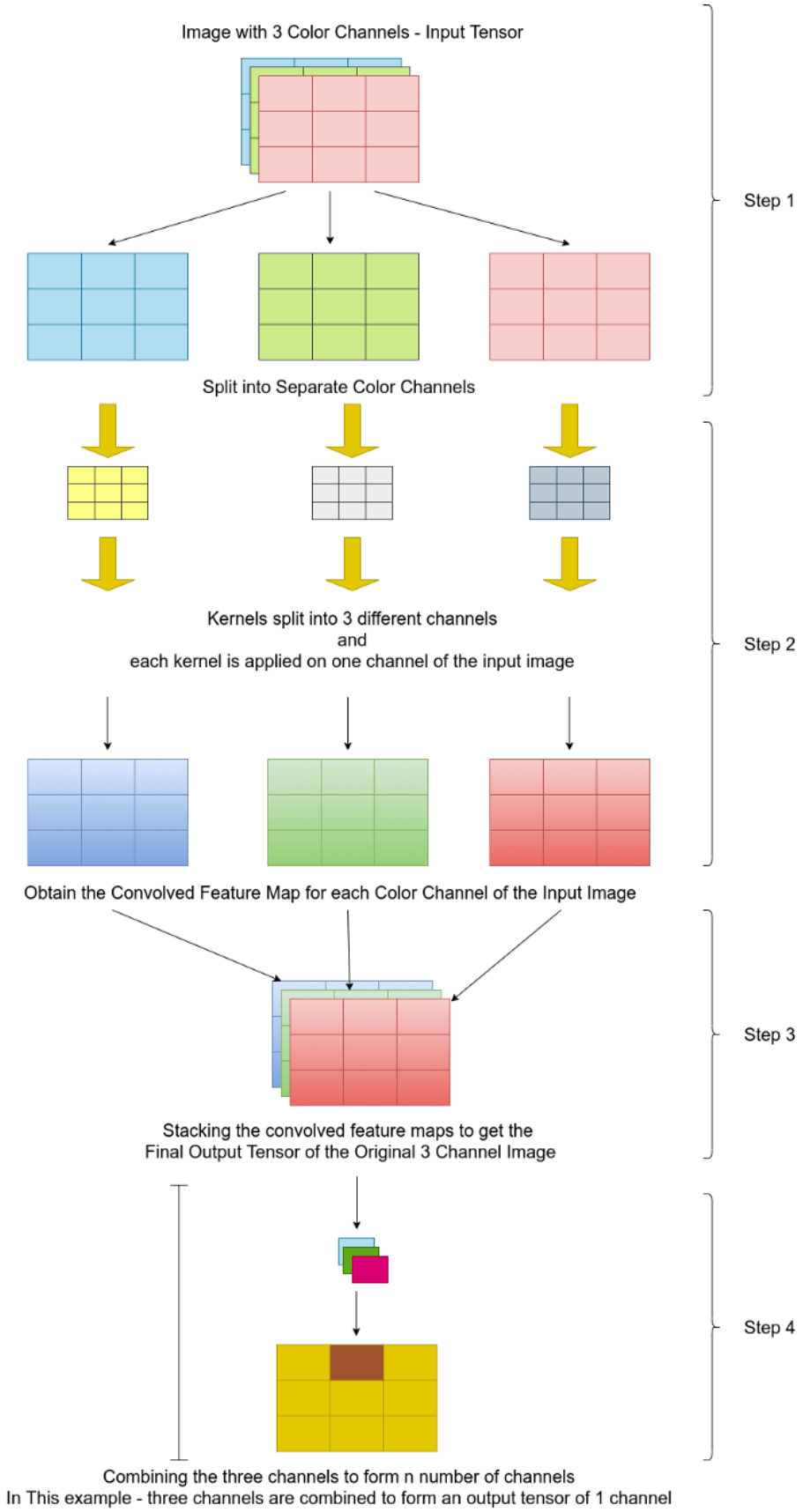


Figure A.3: Illustration of Depthwise Separable Convolution [122].

A.2.2 Knowledge Distillation

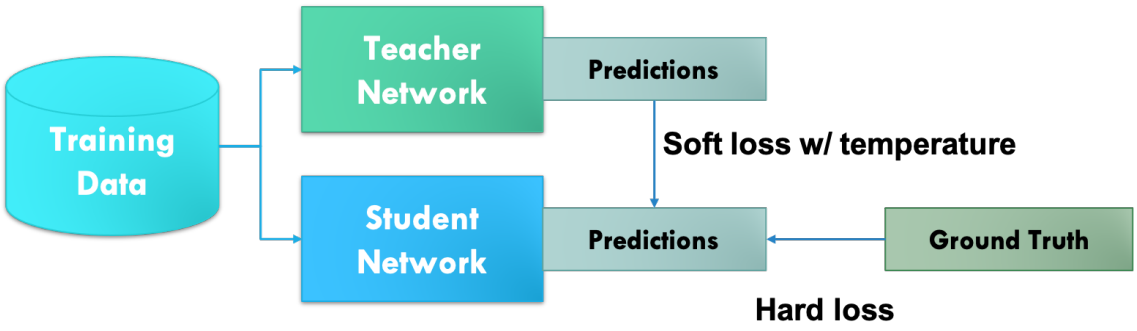


Figure A.4: Illustration of Knowledge Distillation.

Knowledge distillation is a method for transferring knowledge from a large model to a smaller model (also called teacher and student, respectively.). Generally, large models have higher knowledge capacity compared with small models. However, in most cases, the relationship between the increase in the model parameters and the increase in performance is not linear. We can assume that the model's capacity is not fully utilized in this case.

If we can fully utilize the capacity of the smaller model, higher performance can be achieved without using the larger model, and the small model can be deployed on devices with limited computational power, such as mobile devices. In order to realize this concept, Hinton et al. [62] proposed a knowledge distillation method.

In supervised learning, the hard loss between the model predicted value and the ground truth is calculated, and an optimizer backpropagates it to the model to update parameters. In knowledge distillation, in addition to the hard loss, the soft loss between the model prediction and the larger model's prediction is calculated and used for backpropagation. The soft loss for classification is following equation,

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)} \tag{A.4}$$

Here q_i is a probability value for the i th class among j classes, and T is a parameter called temperature. If T is set to 1, it is the same as a standard softmax. Higher temperature values have the effect of generating a softer distribution of probabilities among the classes.

For training, the soft loss is fused with the hard loss through the following loss function.

$$LOSS = \sum_{x,y \in \mathbb{D}} L_{KD}(M_s(x,T), M_t(x,T)) + \lambda L_{CE}(\hat{y}, y_{M_s}) \tag{A.5}$$

x and y means an image and a label in dataset $\mathbb{D}$, respectively. L_{KD} is a soft loss between a teacher model M_t and a student model M_s with temperature T. L_{CE} indecates a common cross entropy loss function between a prediction of a student model y_{M_s} and a ground truth $\hat{y}$, and λ is a blend factor.

References

[1] Canon announces development of the free viewpoint video system virtual camera system that creates an immersive viewing experience. Available online at: `https://global.canon/en/news/2017/20170921.html` (accessed on 16 August 2021).

[2] Intel® true view – see more game than ever. Available online at: `https://www.intel.com/content/www/us/en/sports/technology/true-view.html` (accessed on 24 July 2021).

[3] Takeo kanade collection: Envisioning robotics: Virtualized reality and eye vision. `http://diva.library.cmu.edu/Kanade/kanadeeye.html`. (Accessed on 01/02/2019).

[4] Carnegie mellon university - cmu graphics lab - motion capture library, 2001.

[5] Karan Ahuja, Chris Harrison, Mayank Goel, and Robert Xiao. Mecap: Whole-body digitization for low-cost vr/ar headsets. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 453–462, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-Time Rendering*. A. K. Peters, Ltd., Natick, MA, USA, 3rd edition, 2008.

[7] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Detailed human avatars from monocular video. In *2018 International Conference on 3D Vision (3DV)*, pages 98–109, 2018.

[8] Thiemo Alldieck, Gerard Pons-Moll, Christian Theobalt, and Marcus Magnor. Tex2shape: Detailed full human body geometry from a single image. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2019.

[9] Sikandar Amin, Mykhaylo Andriluka, Marcus Rohrbach, and Bernt Schiele. Multi-view pictorial structures for 3d human pose estimation. In *Procedings of the British Machine Vision Conference 2013*. British Machine Vision Association, 2013.

[10] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014.

[11] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE CVPR 2014*, 2014.

[12] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2654–2662. Curran Associates, Inc., 2014.

[13] Andreas Baak, Meinard Muller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *2011 International Conference on Computer Vision*. IEEE, nov 2011.

[14] Luca Ballan, Gabriel J. Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Trans. Graph.*, 29(4):87:1–87:11, Jul 2010.

[15] Alexander Bogomjakov, Craig Gotsman, and Marcus Magnor. Free-viewpoint video from depth cameras. In *Proceedings of the International Workshop on Vision, Modeling and Visualization (VMV)*, pages 89–96, 2006.

[16] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient n-d image segmentation. *International Journal of Computer Vision*, 70(2):109–131, Nov 2006.

[17] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*. IEEE Comput. Soc, 1998.

[18] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *ACM KDD 2006*, KDD '06, pages 535–541, New York, NY, USA, 2006. ACM.

[19] Adrian Bulat and Georgios Tzimiropoulos. Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In *IEEE ICCV 2017*, 2017.

[20] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017.

[21] Adrian Bulat, Georgios Tzimiropoulos, Jean Kossaifi, and Maja Pantic. Improved training of binary networks for human pose estimation and image recognition. *arXiv preprint arXiv:1904.05868*, 2019.

[22] Magnus Burenius, Josephine Sullivan, and Stefan Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2013.

[23] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012.

[24] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, July 2017.

[25] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 € filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2527–2530, New York, NY, USA, 2012. Association for Computing Machinery.

[26] Y. Cha, T. Price, Z. Wei, X. Lu, N. Rewkowski, R. Chabra, Z. Qin, H. Kim, Z. Su, Y. Liu, A. Ilie, A. State, Z. Xu, J. Frahm, and H. Fuchs. Towards fully mobile 3d face, body, and environment capture using only head-worn cameras. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2993–3004, 2018.

[27] Liwei Chan, Chi-Hao Hsieh, Yi-Ling Chen, Shuo Yang, Da-Yuan Huang, Rong-Hao Liang, and Bing-Yu Chen. Cyclops: Wearable and single-piece full-body gesture input devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3001–3009, New York, NY, USA, 2015. ACM.

[28] Feng-Ju Chang, Anh Tuan Tran, Tal Hassner, Iacopo Masi, Ram Nevatia, and Gerard Medioni. FacePoseNet: Making a case for landmark-free face alignment. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. IEEE, oct 2017.

[29] Jiawen Chen, S. Paris, J. Wang, W. Matusik, M. Cohen, and F. Durand. The video mesh: A data structure for image-based three-dimensional video editing. In *2011 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, April 2011.

[30] Yu Chen, Chunhua Shen, Xiu-Shen Wei, Lingqiao Liu, and Jian Yang. Adversarial PoseNet: A structure-aware convolutional network for human pose estimation. In *IEEE ICCV 2017*, 2017.

[31] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *European Conference on Computer Vision (ECCV)*, 2020.

[32] François Chollet et al. Keras. https://keras.io, 2015.

[33] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] Chia-Jung Chou, Jui-Ting Chien, and Hwann-Tzong Chen. Self adversarial training for human pose estimation. In *APSIPA ASC 2018*. IEEE, 2018.

[35] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *IEEE CVPR 2017*. IEEE, 2017.

[36] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.

[37] Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. High-quality

streamable free-viewpoint video. *ACM Trans. Graph.*, 34(4):69:1–69:13, Jul 2015.

[38] W.W. Daniel. *Applied Nonparametric Statistics*. Duxbury advanced series in statistics and decision sciences. PWS-KENT, 1990.

[39] Edilson de Aguiar, Carsten Stoll, Christian Theobalt, Naveed Ahmed, Hans-Peter Seidel, and Sebastian Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics*, 27(3):1–10, aug 2008.

[40] Daniel F. Dementhon and Larry S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1-2):123–141, jun 1995.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE CVPR 2009*, 2009.

[42] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. L. Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, AVSS '09, pages 559–564, Washington, DC, USA, 2009. IEEE Computer Society.

[43] Ruofei Du, Sujal Bista, and Amitabh Varshney. Video fields: Fusing multiple surveillance videos into a dynamic virtual environment. In *Proceedings of the 21st International Conference on Web3D Technology*, Web3D '16, pages 165–172, New York, NY, USA, 2016. ACM.

[44] Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. Marker-less 3d human motion capture with monocular image sequence and height-maps. In *ECCV 2016*, pages 20–36, 2016.

[45] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014.

[46] A. Elhayek, E. de Aguiar, A. Jain, J. Tompson, L. Pishchulin, M. Andriluka, C. Bregler, B. Schiele, and C. Theobalt. Efficient ConvNet-based marker-less motion capture in general scenes with a low number of cameras. In *2015 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, jun 2015.

[47] R. A. Fisher. *Statistical Methods for Research Workers*, pages 66–70. Springer New York, New York, NY, 1992.

[48] Max Fleischer. Method of producing moving-picture cartoons. *brevet US*, 1917.

[49] Takashi Fukuda, Masayuki Suzuki, Gakuto Kurata, Samuel Thomas, Jia Cui, and Bhuvana Ramabhadran. Efficient knowledge distillation from an ensemble of teachers. In *Interspeech 2017*. ISCA, 2017.

[50] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87(1-2):75–92, nov 2008.

[51] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6602–6611, July 2017.

[52] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[53] P. Goorts, S. Maesen, M. Dumont, S. Rogmans, and P. Bekaert. Free viewpoint video for soccer using histogram-based validity maps in plane sweeping. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 3, pages 378–386, Jan 2014.

[54] O. Grau, A. Hilton, J. Kilner, G. Miller, T. Sargeant, and J. Starck. A free-viewpoint video system for visualization of sport scenes. *SMPTE Motion Imaging Journal*, 116(5-6):213–219, May 2007.

[55] O. Grau, G. A. Thomas, A. Hilton, J. Kilner, and J. Starck. A robust free-viewpoint video system for sport scenes. In *2007 3DTV Conference*, pages 1–4, May 2007.

[56] Marc Habermann, Weipeng Xu, Michael Zollhoefer, Gerard Pons-Moll, and Christian Theobalt. A deeper look into deepcap. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1–1. IEEE, 2021.

[57] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[58] K. Hattori, H. Hattori, Y. Ono, K. Nishino, M. Itoh, V. Boddeti, and T. Kanade. Image dataset for researches about surveillance camera-cmusrd (surveillance research dataset). Technical report, Carnegie Mellon University, 11 2014.

[59] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct 2017.

[60] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[61] Keita Higuchi, Ryo Yonetani, and Yoichi Sato. Egoscanning: Quickly scanning first-person videos with egocentric elastic timelines. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 6536–6546, New York, NY, USA, 2017. Association for Computing Machinery.

[62] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[63] Michael B. Holte, Cuong Tran, Mohan M. Trivedi, and Thomas B. Moeslund. Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of Selected Topics in Signal Processing*, 6(5):538–552, sep 2012.

[64] Jason Hong. Considering privacy issues in the context of google glass. *Commun. ACM*, 56(11):10–11, November 2013.

[65] Youichi Horry, Ken-Ichi Anjyo, and Kiyoshi Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 225–232, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[66] Naho Inamoto and Hideo Saito. Free viewpoint video synthesis and presentation of sporting events for mixed reality entertainment. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, ACE '04, pages 42–50, New York, NY, USA, 2004. ACM.

[67] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML 2015*, ICML'15, pages 448–456. JMLR.org, 2015.

[68] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[69] Karim Iskakov, Egor Burkov, Victor Lempitsky, and Yury Malkov. Learnable triangulation of human pose. In *IEEE ICCV 2019*, 2019.

[70] Hao Jiang and Kristen Grauman. Seeing invisible poses: Estimating 3d body pose from egocentric video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[71] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Procedings of the British Machine Vision Conference 2010*. British Machine Vision Association, 2010.

[72] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC 2010*, 2010. doi:10.5244/C.24.12.

[73] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015.

[74] Y. Kameda, T. Koyama, Y. Mukaigawa, F. Yoshikawa, and Y. Ohta. Free viewpoint browsing of live soccer games. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, volume 1, pages 747–750 Vol.1, June 2004.

[75] T. Kanade, P. Rander, and P. J. Narayanan. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1):34–47, Jan 1997.

[76] Junya Kashiwakuma, Itaru Kitahara, Yoshinari Kameda, and Yuichi Ohta. A virtual camera controlling method using multi-touch gestures for capturing free-viewpoint video. In *Proceedings of the 11th European Conference on Interactive TV and Video*, EuroITV '13, pages 67–74, New York, NY, USA, 2013. ACM.

[77] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014.

[78] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[79] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[80] Itaru Kitahara, Yuichi Ohta, Hideo Saito, Shinji Akimichi, Tooru Ono, and Takeo Kanade. Recording of multiple videos in a large-scale space for large-scale virtualized reality. *Kyokai Joho Imeji Zasshi/Journal of the Institute of Image Information and Television Engineers*, 56(8):1328–1333, 8 2002.

[81] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Self-supervised learning of 3d human pose using multi-view geometry. In *IEEE CVPR 2019*, 2019.

[82] T. Koyama, I. Kitahara, and Y. Ohta. Live mixed-reality 3d video in soccer stadium. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 178–186, Oct 2003.

[83] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[84] Amit Kumar, Azadeh Alavi, and Rama Chellappa. KEPLER: Keypoint and pose estimation of unconstrained faces by learning efficient h-CNN regressors. In *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, may 2017.

[85] Claudia Kuster, Tiberiu Popa, Christopher Zach, Craig Gotsman, and Markus H. Gross. Freecam: A hybrid camera system for interactive free-viewpoint video. In *VMV*, 2011.

[86] Tobias Langlotz, Mathäus Zingerle, Raphael Grasset, Hannes Kaufmann, and Gerhard Reitmayr. Ar record & replay: Situated compositing of video content in mobile augmented reality. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*, OzCHI '12, pages 318–326, New York, NY, USA, 2012. ACM.

[87] Sijin Li and Antoni B. Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *ACCV 2014*, pages 332–347. Springer, 2014.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing, 2014.

[89] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Smpl: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6), October 2015.

[90] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *IEEE CVPR 2018*, 2018.

[91] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[92] Etienne-Jules Marey. Movement, 1972.

[93] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3d human pose estimation. In *IEEE ICCV 2017*, 2017.

[94] K. Matsumoto, C. Song, F. de Sorbier, and H. Saito. Free viewpoint video synthesis using multi-view depth and color cameras. In *IVMSP 2013*, pages 1–4, June 2013.

[95] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 Fifth International Conference on*. IEEE, 2017.

[96] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose

estimation in the wild using improved CNN supervision. In *IEEE 3DV 2017*, 2017.

[97] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Mohamed Elgharib, Pascal Fua, Hans-Peter Seidel, Helge Rhodin, Gerard Pons-Moll, and Christian Theobalt. Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. *arXiv preprint arXiv:1907.00837*, 2019.

[98] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics*, 36(4), July 2017.

[99] Thomas B. Moeslund, Adrian Hilton, Volker Krger, and Leonid Sigal. *Visual Analysis of Humans: Looking at People*. Springer Publishing Company, Incorporated, 2013.

[100] Peter Mohr, David Mandl, Markus Tatzgern, Eduardo Veas, Dieter Schmalstieg, and Denis Kalkofen. Retargeting video tutorials showing tools with surface contact to augmented reality. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 6547–6558, New York, NY, USA, 2017. ACM.

[101] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV 2016*, pages 483–499. Springer, 2016.

[102] Evonne Ng, Donglai Xiang, Hanbyul Joo, and Kristen Grauman. You2me: Inferring body pose in egocentric video via first and second person interactions. *CVPR*, 2020.

[103] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 741–754, New York, NY, USA, 2016. ACM.

[104] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *Lecture Notes in Computer Science*, pages 156–169. Springer International Publishing, 2016.

[105] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[106] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *IEEE CVPR 2018*, 2018.

[107] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *IEEE CVPR 2017*, 2017.

[108] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting multiple views for marker-less 3d human pose annotations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[109] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[110] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *IEEE CVPR 2019*, 2019.

[111] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, jan 2019.

[112] Konstantinos Rematas, Ira Kemelmacher-Shlizerman, Brian Curless, and Steve Seitz. Soccer on your tabletop. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[113] Helge Rhodin, Christian Richardt, Dan Casas, Eldar Insafutdinov, Mohammad Shafiei, Hans-Peter Seidel, Bernt Schiele, and Christian Theobalt. EgoCap. *ACM Transactions on Graphics*, 35(6):1–11, nov 2016.

[114] Helge Rhodin, Nadia Robertini, Christian Richardt, Hans-Peter Seidel, and Christian Theobalt. A versatile scene model with differentiable visibility applied to generative pose estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015.

[115] Nadia Robertini, Dan Casas, Helge Rhodin, Hans-Peter Seidel, and Christian Theobalt. Model-based outdoor performance capture. In *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, oct 2016.

[116] Gregory Rogez, James S. Supancic, and Deva Ramanan. First-person pose recognition using egocentric workspaces. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[117] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR 2015*, 2015.

[118] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2018.

[119] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[120] Chris Russell, Rui Yu, and Lourdes Agapito. Video pop-up: Monocular 3d reconstruction of dynamic scenes. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 583–598, Cham, 2014. Springer International Publishing.

[121] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE CVPR 2018*, 2018.

[122] Arjun Sarkar. Understanding depthwise separable convolutions and the efficiency of mobilenets, Jun 2021.

[123] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM.

[124] Takaaki Shiratori, Hyun Soo Park, Leonid Sigal, Yaser Sheikh, and Jessica K. Hodgins. Motion capture from body-mounted cameras. In *ACM SIGGRAPH 2011 papers on - SIGGRAPH '11*. ACM Press, 2011.

[125] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*. IEEE, jun 2011.

[126] Leonid Sigal, Alexandru O. Balan, and Michael J. Black. HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1-2):4–27, aug 2009.

[127] Leonid Sigal, Michael Isard, Horst Haussecker, and Michael J. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, 98(1):15–48, sep 2011.

[128] Ernie Smith. This is what 1970s motion capture tech looked like, Mar 2017.

[129] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *2011 International Conference on Computer Vision*. IEEE, nov 2011.

[130] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE CVPR 2019*. IEEE, 2019.

[131] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019.

[132] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[133] Bugra Tekin, Pablo Marquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *IEEE ICCV 2017*, 2017.

[134] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop, coursera: Neural networks for machine learning. *University of Toronto, Technical Report*, 2012.

[135] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xR-EgoPose: Egocentric 3d human pose from an HMD camera. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[136] Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2014.

[137] Raquel Urtasun, David J. Fleet, and Pascal Fua. Temporal motion models for monocular and multiview 3d human body tracking. *Computer Vision and Image Understanding*, 104(2-3):157–177, nov 2006.

[138] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017.

[139] T. Watanabe, I. Kitahara, Y. Kameda, and Y. Ohta. 3d free-viewpoint video capturing interface by using bimanual operation. In *2010 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4, June 2010.

[140] Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics*, 31(6):1–12, nov 2012.

[141] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

[142] Jianxiong Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012.

[143] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015.

[144] Weipeng Xu, Avishek Chatterjee, Michael Zollhofer, Helge Rhodin, Pascal Fua, Hans-Peter Seidel, and Christian Theobalt. Mo2cap2 : Real-time mobile 3d motion capture with a cap-mounted fisheye camera. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):2093–2101, may 2019.

[145] Xiangyu Xu and Chen Change Loy. 3d human texture estimation from a single image with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13849–13858, October 2021.

[146] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. FSA-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019.

[147] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *IEEE CVPR 2018*, 2018.

[148] Qiang Yao, Hiroshi Sankoh, Keisuke Nonaka, and Sei Naito. Automatic camera self-calibration for immersive navigation of free viewpoint sports video. *2016 IEEE 18th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2016.

[149] G. Ye, Y. Liu, Y. Deng, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Free-viewpoint video of human actors using multiple handheld kinects. *IEEE Transactions on Cybernetics*, 43(5):1370–1382, Oct 2013.

[150] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE CVPR 2017*, 2017.

[151] Haruka Yonemoto, Kazuhiko Murasaki, Tatsuya Osawa, Kyoko Sudo, Jun Shimamura, and Yukinobu Taniguchi. Egocentric articulated pose tracking for action recognition. In *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, may 2015.

[152] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. Humbi: A large multiview dataset of human body expressions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[153] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Computer Vision – ECCV 2018*, pages 763–778. Springer International Publishing, 2018.

[154] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time PD control. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[155] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *IEEE CVPR 2019*, 2019.

[156] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE CVPR 2018*, 2018.

[157] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *IEEE CVPR 2018*, 2018.

[158] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[159] Xiaowei Zhou, Menglong Zhu, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *IEEE CVPR 2016*, 2016.

[160] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. *CoRR*, abs/1511.07212, 2015.

[161] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.

[162] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7):773–780, May 2006.