論文 / 著書情報
Article / Book Information

題目(和文)	デュアル間接特徴を用いた人間姿勢予測と 技能習得への応用
Title(English)	Human Pose Prediction using Dual Indirect Features and its Application in Skill Acquisition
著者(和文)	WU ERWIN
Author(English)	Erwin Wu
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12248号, 授与年月日:2022年9月22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,篠田 浩一,岡崎 直観
Citation(English)	Degree:Doctor (Academic), Conferring organization: Tokyo Institute of Technology, Report number:甲第12248号, Conferred date:2022/9/22, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctor Dissertation

Human Pose Prediction using Dual Indirect Features and its Application in Skill Acquisition

Tokyo Institute of Technology

School of Computing

Department of Computer Science

Name : Erwin WU

Main Supervisor : Professor Hideki KOIKE

Submission Date : 2022/7/6

Abstract

In this paper, we proposed an indirect feature-based real time pose prediction system and introduced several applications for skill acquisition using the proposed system.

Different from conventional direct feature-based pose estimation, the proposed system try to make use of those features which are indirect related with human posture (e.g. estimate full body pose from feet pressure).

The proposed network consists of two parts: a FuturePoseNet which aims to extract temporal indirect features from the input video sequences and a InvisiblePoseNet which finds out the spatial indirect relationship within each image. For each network, a special indirect feature extraction module is developed to enhance the learning of an indirect feature. The performance of both networks is quantitatively and qualitatively evaluated in the experiment, and the results suggest that the proposed indirect feature-based prediction can achieve similar accuracy as the conventional methods, without observing the direct features.

For applications, three types of different skill acquisition are introduced: Skiing, Piano, and Table Tennis, which aims to study the results from three different perspectives. The Skiing is mainly focus on spatial indirect features while the piano requires temporal one. Table Tennis is the most well-studied application which includes both temporal and spatial indirect features.

Finally, the contribution and limitation of the current work is discussed. The proposed framework is compared with other existing methods to provide clues for future research. To the best of our knowledge, this work is the first real-time 3D pose prediction using a dual-module indirect feature-based network, which is proved to be useful in different types of skill training and might open a new way for using indirect features in training.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Motion Tracking in Skill Analysis

Nowadays, Human-Computer Interactions (HCI) technologies are widely used in analyzing advanced skills such as sports [2, 29, 48, 72], musical instruments [21], or even medical operations [43]. One of the most essential keys to understand an advance skill is to analyze its posture, from an overall body posture to a specific dexterous finger movement. This is to say, the development of motion capture systems [33, 49, 52, 56] are changing the researches of HCI. These technologies make it possible to analyze motions of an athlete and giving real-time feedback for pointing out their mistakes, improving their performances, or differentiate with other's motions. More importantly, a correctly recorded motion can be used to transfer a skill to others with less practice and training.

The methods of commercial motion capture system can be broadly divided into three categories based on their principles: optical marker-based methods [49], wearable sensor-based methods [52], depth camera-based methods [56, 80]. The first two ways have already been widely studied and therefore is well established, which can achieve a high precision in high speed. However, both methods requires special markers or sensors to be wore by the users, these markers are sometimes bulky and may disturb the users in performance, which is not suitable to be used in many situations such as real-time sports games. On the other hand, depth camera-based

Figure 1.1: Examples of motion capture technologies: optical marker-based Optitrack [49] (upper), IMU sensor-based Xsens [52] (lower left), and depth camera based Microsoft Kinect [80].

Figure 1.2: Examples of skill transfer using motion tracking.

methods are hardware-dependent and limited to environment, e.g. IR cameras cannot be used in the wild. Therefore, all these motion capture technologies introduce hardware restrictions which limit the target to profession.

1.1.2 Deep Learning-based Pose Regression

Under these circumstances, human pose estimation methods using deep neural network [8, 47, 58, 61, 65, 69] has been widely studied. Vision-based pose estimations use convolutional networks to extract visual human features from images or videos. Among them, regressing 3D posture from a single RGB image is an essential and challenging task. Single camera-based pose estimation enables markerless and in-the-wild motion capture, which can be applied to much wider field such as sports or dexterous skills. Especially for real-time estimations, which can not only be used in afterwards analysis but also provide real-time feedback or support training.

All of these above mentioned estimation shows the importance of pose understanding and motion analysis. However, estimation (which is a direct feature regression) is only the basis behavior of human brain, in terms of more advance-level

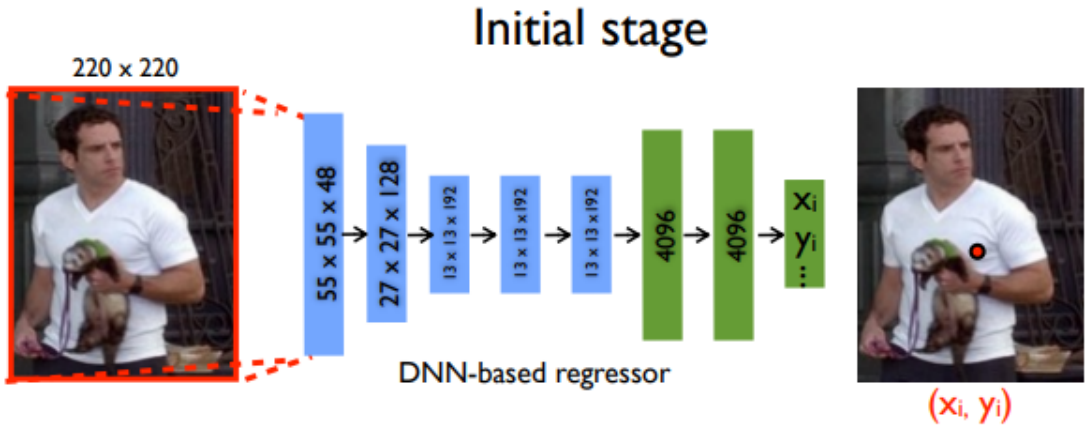


Figure 1.3: Deep Learning-based pose estimation [8, 58, 69].

skill acquisition, such as understanding one's thoughts or transfer one's experience, we need to go one step further to indirect feature regression.

1.1.3 Prediction from Indirect Features

Different from the ability of estimating or recognizing an object, the ability of prediction is more mandatory in some advance skill. For instance, a recent report from NTT Research [35, 36] tried to understand and shape a professional baseball athlete's brain by comparing his/her motion data with an amateur. The results showed that a professional batter reacts to a curve ball before the ball starts to change while an amateur reacts after.

Predictions, which can be also defined as indirect features regressions, which mean that human brain estimate from some indirect features which are unseen/unavailable for the present information. In this paper, we mainly focus on two types of indirect features: temporal indirect features and spatial indirect features.

An estimation using temporal indirect features, which are commonly known as future prediction, is an ability to predict information using experience from the past. In the field of deep neural network, recurrent neural networks [53] are trying to realize the same function. Yagi et al. [74], for example, developed a network to predict a pedestrian's future position.

On the other hand, predictions using spatial indirect features make it possible to let people estimate a whole target with partially information. For example, even if someone's body is occluded, our brain can make up the full posture based on our experience. Nowadays, more networks are developed to study these spatial indirect relationships, such as graph neural networks [55] and attention networks [68].

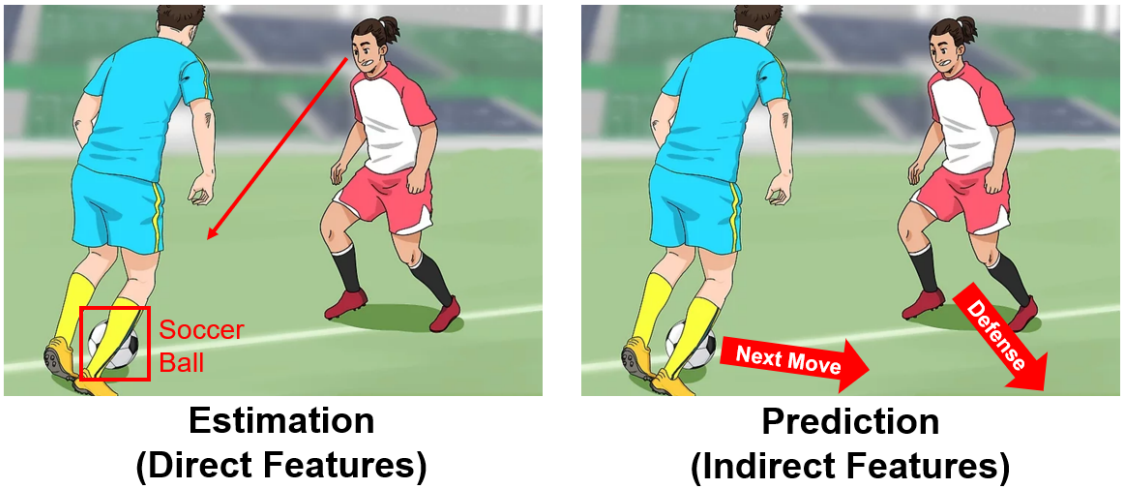


Figure 1.4: Difference between direct and indirect feature-based estimation

1.2 Research Motivation

With all the mentioned technologies, ideally, it should be possible to employ the data of experienced athletes, musicians, or doctors, and provide an intuitive instruction to a student who want to learn an advanced skill. However, there are two major problems of existing works: first, networks for different types of indirect features are lacking; second, there are currently few well-studied skill acquisition application using these indirect features.

Therefore, in this paper, we propose a novel indirect feature-based pose estimation network – IndirectPoseNet, to serve as a strong baseline which can estimate real-time 3D human posture from both temporal and spatial vision features. Our system uses a two-stream customized recurrent convolutional network (RCNN) to obtain the temporal movement of a specific posture and the spatial information for body regression. To enhance the temporal extraction, we developed a lattice optical flow algorithm to calculate the joint movement with less computation. On the other hand, to obtain the indirect spatial feature, we developed a graph-based model to study the hidden relationship between the extracted spatial features and the predicted posture.

1.2.1 Research Target

To the best of our knowledge, our system is the first to realize real-time indirect feature-based pose forecasting and apply it to skill acquisition. Compared to previous work, our system does not require users to wear special suits and can be used outdoors or in large environment since the motions can be captured by a single RGB camera, which leads to a higher usability and adaptability.

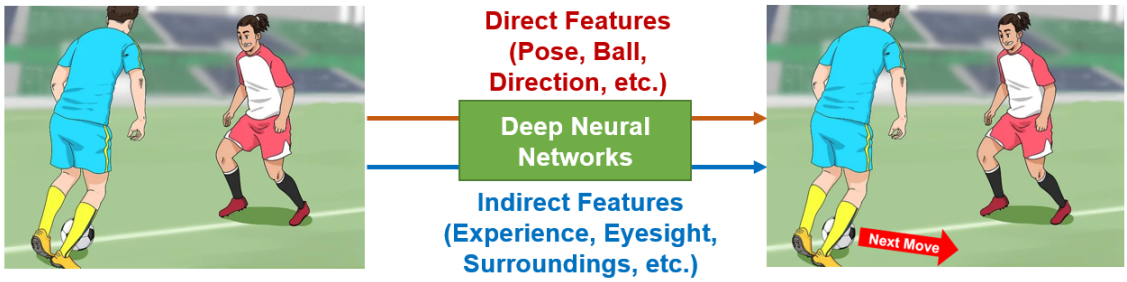


Figure 1.5: Our Target: using indirect spatial and temporal features to support advance skill acquisition.

1.3 Organization

The organization of the whole paper is arranged as follows:

1. In this thesis, the author have already introduced the background and innovation of current work in the introduction section.

2. Then, the related works about real-time pose estimations, indirect feature-based estimations as well as skill acquisition using these forecasting will be described . After that, the research proposal of this work will be introduced.

3. The next chapter after research proposal will be the introduction of the network architectures used in this paper.

4. Next, we will first explain how the proposed network architectures are used in predictions of temporal indirect features, which is also considered as future pose prediction.

5. Following the temporal one, the spatial indirect feature-based algorithm will also be proposed. Explaining how the proposed network is tuned to predict spatial features.

6. Detailed studies for the two indirect features will follow on each chapter. Quantitative evaluations are performed to show the accuracy and significance of this system.

7. As one of the most important part of this dissertation, several skill acquisition applications including sports and musical instruments are displayed in Chapter 7.

8. To proof the concept of training effects, user studies are conducted for all the applications. The experiments include several performance metrics in the corresponding skill and a detailed qualitative questionnaire to study user's experiences.

9. In the chapter of discussion, a summary including the result of the evaluations and the limitations of this system will be shown. And corresponding solutions to handle these disadvantages as well as the future vision of this project will be described.

10. Finally, a summary of this research as well as acknowledgement to related staffs will be given.

Chapter 2

Related Work

In this chapter, existing works related to this study will be introduced from 5 different perspectives: First, an overall summary of vision-based real-time 2D and 3D pose estimation is introduced. After that, current researches on temporal and spatial indirect feature-based estimation are compared. Next, we explain how current deep learning is related with skill acquisition. Finally, training feedback methods using VR/AR are shown.

2.1 Pose Estimation

2.1.1 Vision-based Real-time Pose estimation

Many deep learning-based real-time pose estimation from camera images are proposed during the last few years [8, 38, 46, 47, 58, 61, 63, 69, 76]. The OpenPose [8, 58, 69] represents the first real-time multi-person system to jointly detect human body, hand, facial, and foot keypoints (in total 135 keypoints) on a single image. They introduced the Part Affinity Fields (PAFs) to learn to associate body parts with individuals in the image, and their system is proved to achieve high accuracy and real-time performance, regardless of the number of people in the image (as shown in Figure 2.1). Even though their work is already published 3 years, it is still one of the best 2D pose estimation method.

In terms of 3D joints position, Recent works [8, 38, 47, 63, 76] are trying different approach to get one step further to reconstruct 3D postures from 2D by image

Figure 2.1: OpenPose

observations. Among them, Mehta et al.'s VNect [47] and the 3-D Reconstruction Module of OpenPose by Cao et al. [8, 58, 69] are the current state of the art methods for real-time 3D human pose estimation. The VNect combines a new convolutional neural network based pose regressor with kinematic skeleton fitting. Their fully-convolutional pose formulation regresses 2D and 3D joint positions jointly in real time and does not require tightly cropped input frames. As a result, their network provides a better accuracy for the 3D skeleton recognition with less computation and good real-time ability, even though it cannot be used in multi-person detection. Conversely, OpenPose learns the body parts associated with individuals and they can detect multiple people in a single image, while the inference time of it is greater

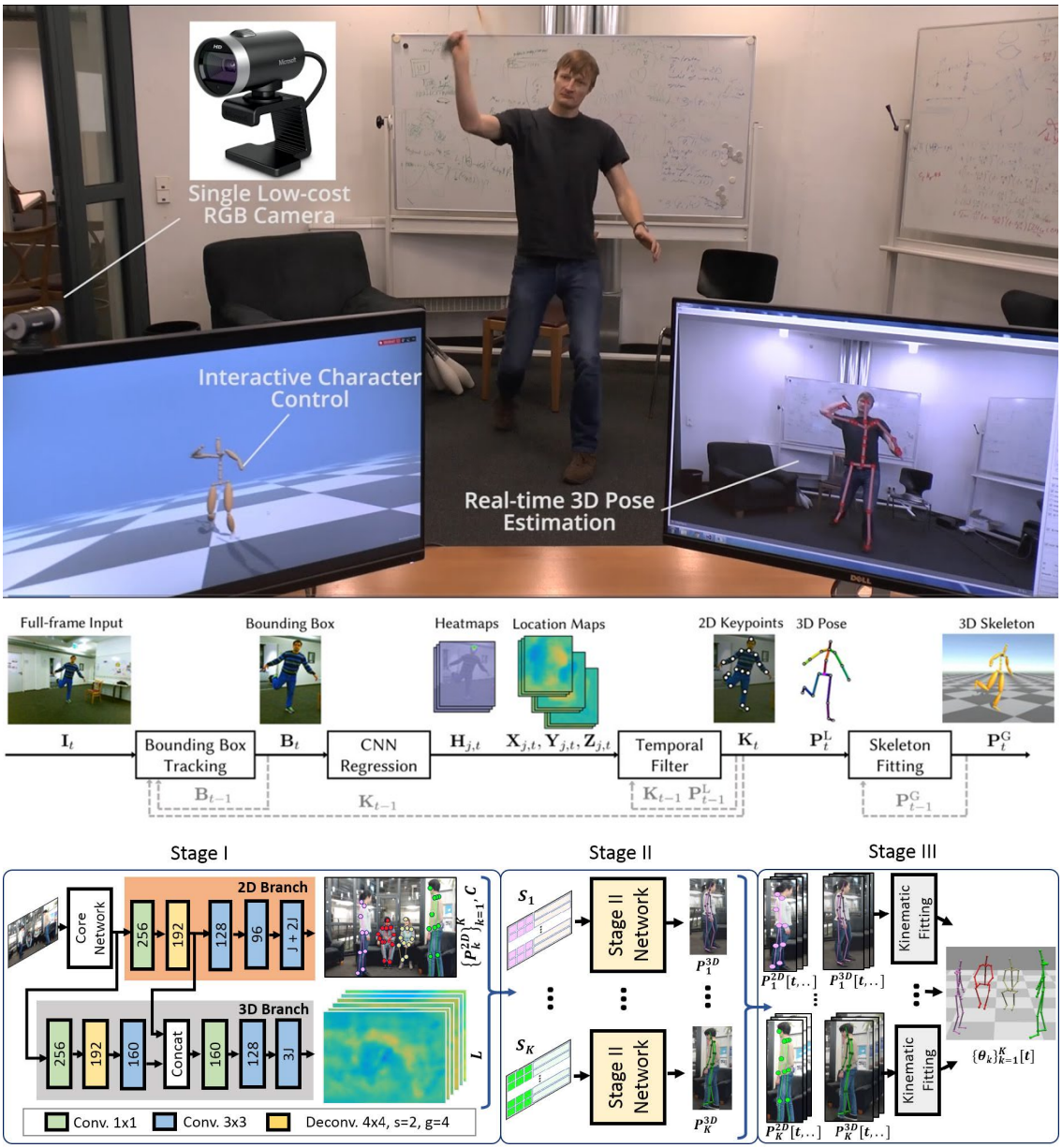


Figure 2.2: VNect & XNect

than VNect.

As a follow-up of the VNect, as well as the state-of-the-art real-time 3D pose estimation, Mehta et al. presented XNect [46], which realize multi-person 3D motion capture using a single RGB camera. They improved their previous network into a two-branch architecture, where 2D and 3D joint heatmaps are regressed separately.

Different from the method mentioned above, Martinez et al. [42] presented an effective network for 3D Pose Recovery using a simple and deep neural network with only two linear layers and two residual blocks (six linear layers in total). Their evaluation demonstrated that a 3D pose could be created from simple 2D joint positions and their method achieved acceptable results in both accuracy and real-time ability on the Human3.6M [9, 30] dataset.

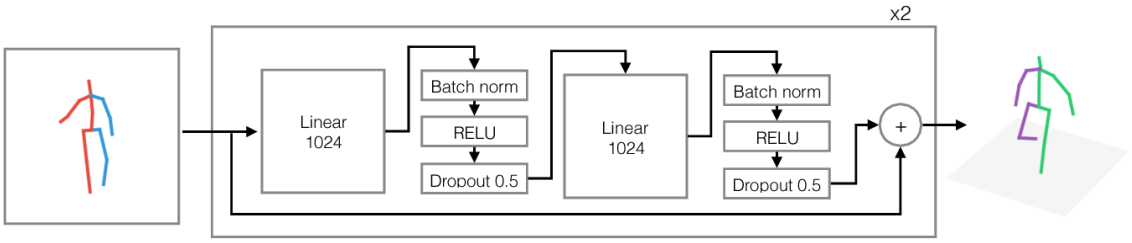


Figure 2.3: Martinez et al.

2.1.2 Temporal Indirect Feature-based Estimation

For temporal indirect feature-base estimation (or simply called future pose prediction), Chao et al. [10] proposed the 3D Pose Forecasting Network (3D-PFNet) as the first study on forecasting human dynamics from single RGB images. Their method of forecasting 2D skeletal poses and converting them into 3D space was shown to have quantitative results, with average joint position errors of approximately 87.6mm. However, 3D-PFNet is an off-line network requiring a large amount of computation, and is therefore difficult to use in the contents of sports which require immediate feedbacks.

Horiuchi et al. [28] forecast human body motions 0.5s (15 frames in 30 fps video) in advance using a five-layered neural network with motion data input taken by a Microsoft Kinect V2 camara [56, 80]; the maximum difference in the prediction was 7.9cm which was acceptable for their experiment. However, Kinect is a depth camera using IR sensors, as previously mentioned: therefore, it is not suitable to use in an outdoor environment or a large area. A five-layered neural network might

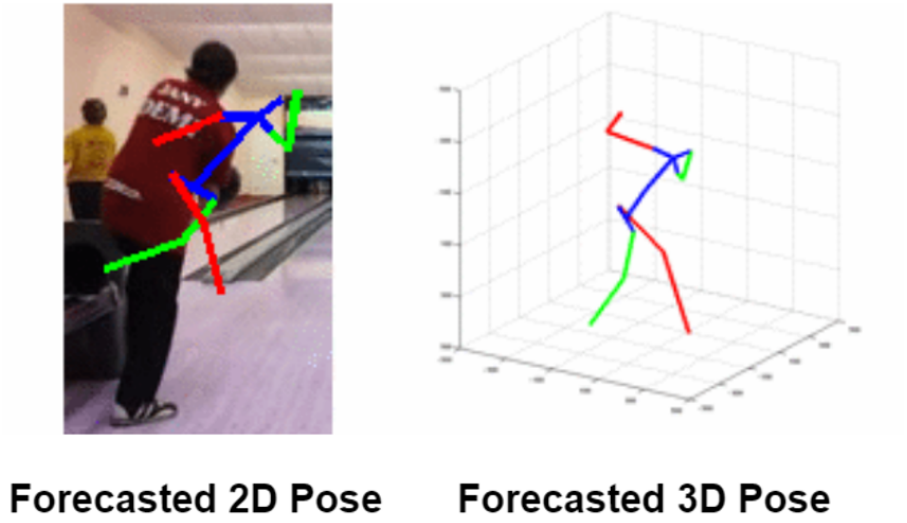


Figure 2.4: 3DPFNet

be enough for simple jumping actions, which was the case in their experiments, but not for more complicated athletic movement such as boxing, where temporal features is of great importance.

Yagi et al. developed a future person localization system [74] for estimating other pedestrian's walking trajectory from a first-person-view video using a three stream encoder-decoder network. Each stream extracts the location-scale, ego-motion, and the target person's posture from the past temporal sequences, respectively. Their final results outperforms some recurrent networks such as LSTM [27] in several first-person locomotion dataset.

The two above mentioned system can perform real-time future prediction, however, they either requires depth information or can only do future localization but not predict future posture, while our final target is to realize real-time future pose prediction using a single RGB camera.

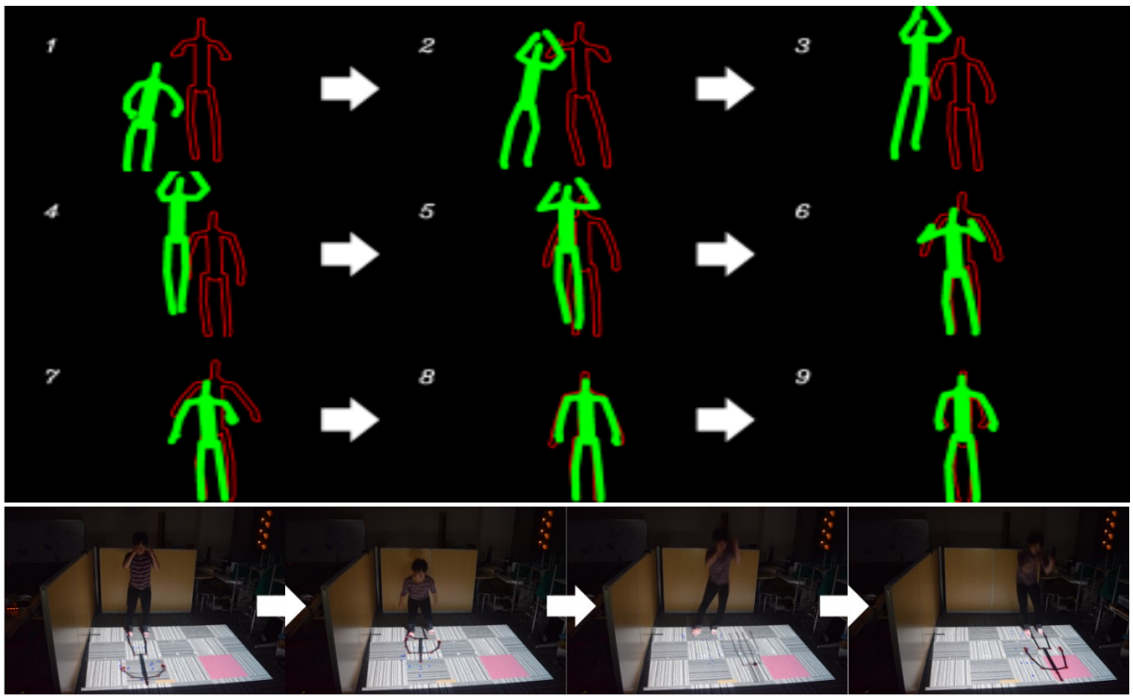


Figure 2.5: Computational Foresight

2.1.3 Spatial Indirect Feature-based Pose Estimation

Recent years, the research interest in the field of indirect spatial feature-based pose estimation is becoming popular. One of the most representative research is the egocentric pose estimation [64, 78, 79], which regress full body poses only from first-person-view video, as shown in Figure 2.6. For example, Yuan et al. [79] proposed the EgoPose Net using a proportional-derivative control based policy, which learns human motion directly from unsegmented egocentric videos. This kind of indirect relationship between egocentric videos and human postures are deeply related to the prediction behavior of human's brain and are often used for robotics operations.

Speaking of learning the relationship, graph neural networks [55] have being widely used to extract a feature graph instead of conventional direct regressions. Reddy et al. [51] proposed the Occlusion-Net which is the first graph networks aims to estimate keypoint from occluded images. In terms of body posture, Cai et al. [7]

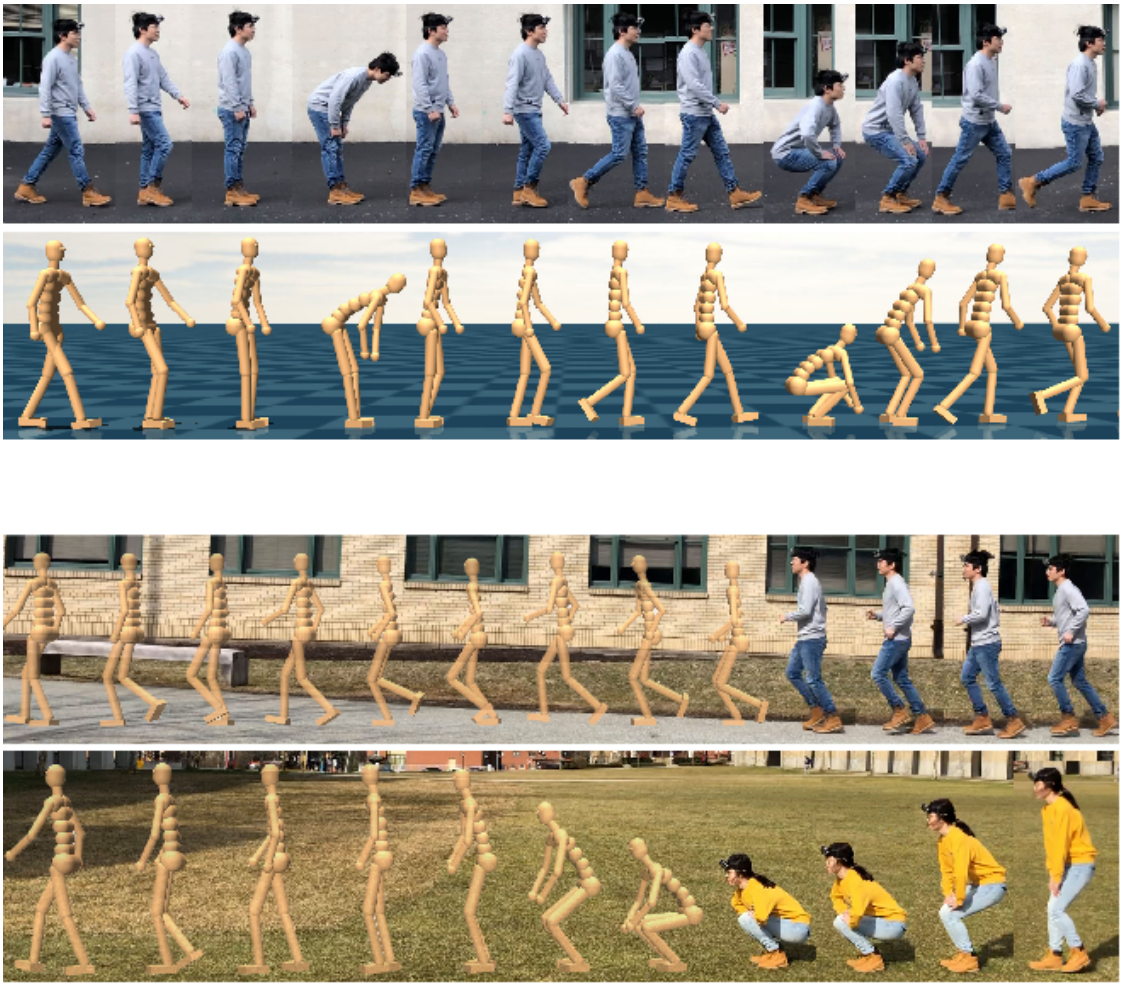


Figure 2.6: EgoPose

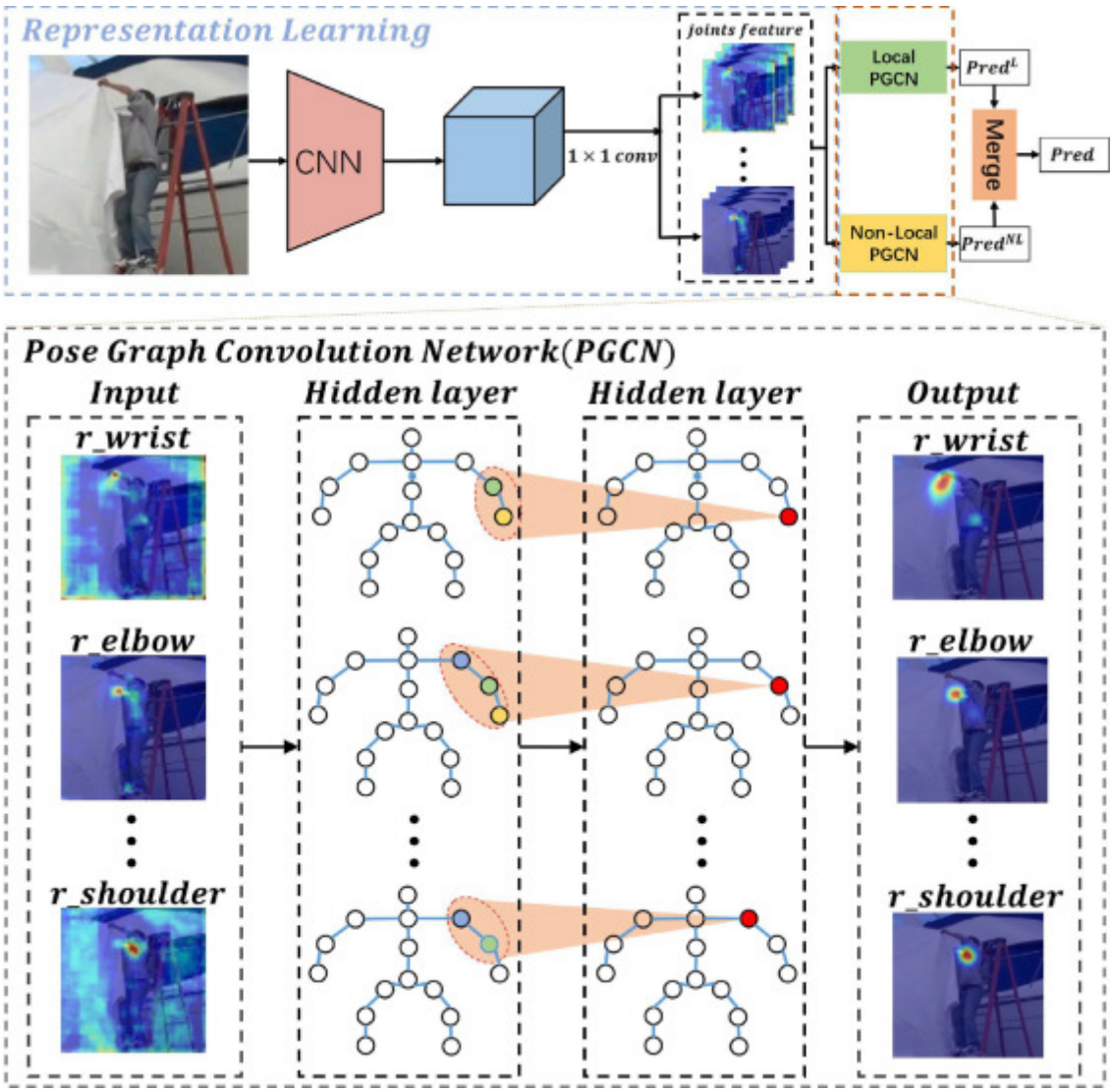


Figure 2.7: Graph Neural Network Pose

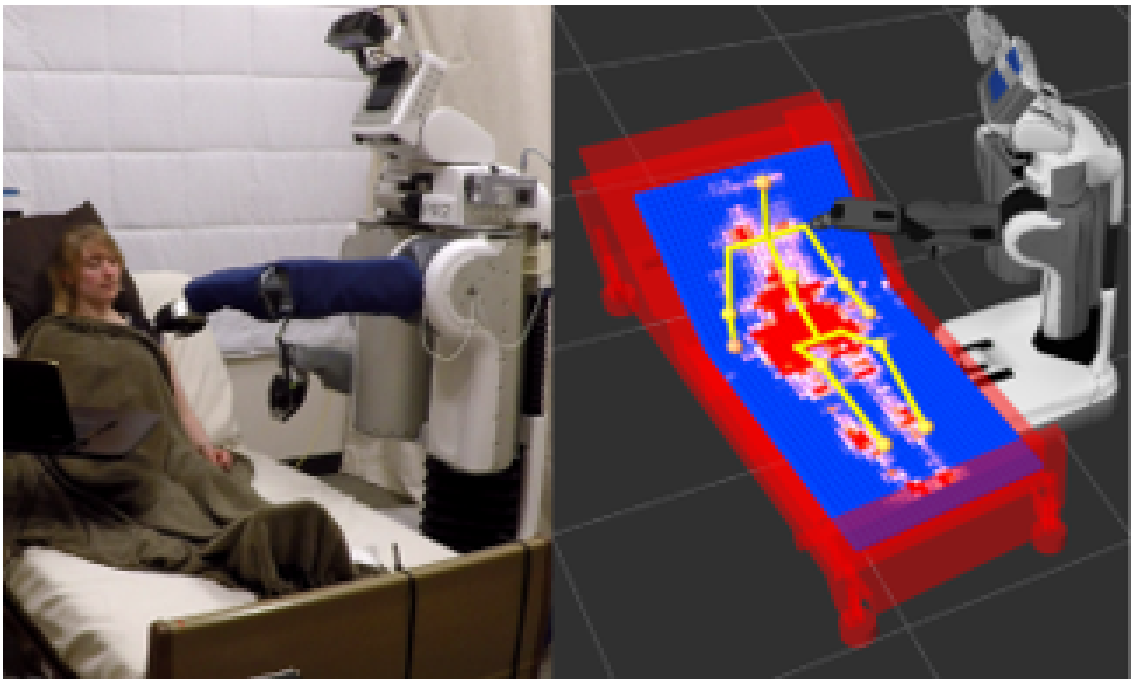


Figure 2.8: Pressure Bed

used graph convolutional networks (GCN) to realize 3D pose estimation from a short sequence of 2D joints positions. Similarly, Yang et al. [75] employ deep graph neural networks for learning dynamics in human motions. Their network fix the problem of missing occluded person of conventional pose estimation methods, as shown in Figure 2.7, and beat the current state-of-the-art pose estimation network in predicting human pose with high occlusion.

One of the most related idea is the pressure bed by Clever et al. [13], they used a configurable bed for medical healthcare and predict the real-time 3D body pose from the pressure image, which succeed in estimate the whole body posture even if part of the body is in the mid-air. Nevertheless, their work consists of a simple convolutional network and a limited kinematic model, which might be sufficient for simple motion on bed but not more complicated motion. Also, the bed pressure image almost cover the whole body, which make the task relatively straightforward, just like regression from a depth map.

All these works are using indirect spatial features to estimate human posture,

however, most of them are focusing on improving the precision of the networks or predicting less related targets. To the best of our knowledge, currently there are few works trying to apply these indirect spatial features to skill transfer.

2.2 Skill Acquisition

2.2.1 Skill Acquisition using Pose Estimation

In many motor skills such as sports or musical instruments, a correct posture is the most essential feature. Therefore, the very basic step for most beginners is to try to mimic a correct (or an ideal) posture and spend plenty of time to master it. However, during this process, the learner may face several difficulties.

One problem is the motivation. Studies [25] have already proved that repetitive and similar training may lower learner's motivation and reduce the learning effect. Chen et al. [12] visualize user's posture and provide scoring in tai-chi training, the results of their study suggested that user can maintain longer concentration when they can objectively see their growth. Also, Nozawa et al. [48] perform similar study on a ski simulator using different visual cues.

Another issue is also related with the motivation, the difficulty. Depends on the target skill, some of the "basic steps" may still be very difficult for beginners. Estimating a spin serve is such an example, Wu et al. [71] showed that many beginners can hardly understand the relationship between the spin type of a serve and the serve motion of the opponent, even after hundreds hours of training. Their study suggests that showing the server's posture and the spin ball simultaneously can support the speed of understanding. The last one is the weak self-understanding, it is very difficult for people to objectively observe their posture, which can be told by the study by Susan Higgins [25]. Because when you control your brain to mimic a pose, your brain is thinking that it's doing correctly although it looks totally different from others. That's why dancer always practice dancing in a mirrored

Figure 2.9: Skill Acquisition using Pose Estimation

room which can better compare themselves with others [26]. A feedback is always needed to objectively notify the learners how to correct their poses, which will be introduced in the next section.

2.2.2 Training Feedback using VR/AR

Since predicting 3D pose comes true nowadays, one of the best visualization method to show the 3D contents intuitively is the XR, such as augmented reality (AR), virtual reality (VR) or mixed reality (MR). Plenty of these types of artificial reality devices was developed [11, 16, 18]. Hämäläinen et al [22] are the first to bring artificial reality to martial arts, who introduce a game where the player fights virtual enemies. The player's motion was taken by real-time image processing and visualized on two large displays. However, their system is limited to single person and does not support person versus person, their virtual environment is pseudo-3D since the user is treated as a 2D plane within the 3D scene.

Ikeda et al. [29] proposed a method of replaying the motion of sports in mixed reality for golf training. They recorded the motion of an expert and replay the whole action on a MR HMD, which also use a special DP matching to tell the difference between the user and the experts. However, their system still require a recorded data which means it cannot work in real-time.

On the other hand, plenty of VR sports [20, 37] were developed recently, however, all of them require both of the players to wear a VR HMD and to take a pair of controllers, which changes the martial arts only into a game and therefore not suitable for martial training.

Figure 2.10: VR Sports

Chapter 3

Research Proposal

3.1 Problem of previous work

In this paper, we focus on how to benefit the training of skill acquisition. From the chapter of introduction and related work, it is clear that deep learning-based pose estimation is uniquely beneficial to skill acquisition. Also, human's ability of prediction can also be learned by the network, which can enhance the training. However, prediction from indirect features is challenging, current researches still face several problems which limit the use for skill transfer. Hereby we summarize these issues as follows:

1. Requirement of bulky equipment: Most of the pose estimation systems require some specific suits or sensors to be placed on user's whole body. This also results in some special environmental dependencies and might disturb the user frequently. Even though some marker-less approaches [8, 46, 47, 58, 69] have been proposed, most of them requires to be fixed at some place thus not suitable for many situations (such as outdoor sports).

2. Single type of features: The prediction of human beings are based on different clues, including temporal and spatial one. However, most of the existing pose estimation network only focus on one specific type of features. The recurrent graph convolutional network [50] mentioned before does focus on both temporal and graph branch, however, the inference time of their network is relatively heavy and their graph network only focus on self-occlusion but

not relationship body and other parts. The work from Clever et al. [13] is a successful example which regress 3D poses from the pressure bed, which we can refer. Nevertheless, their task is relatively simple because their pressure map covers the whole body, which is "less indirect".

3. Lack of good applications: This might be the most essential and critical point, which is directly related to the training effect. Even though some works [28, 70, 74] succeeded in predicting or forecasting 3D human pose in real-time, they don't have good method to visualize or feedback it to a learner for skill acquisition. Most of them still use screen or 2D projections to show the result or error [29], which are not making full use of the 3D information.

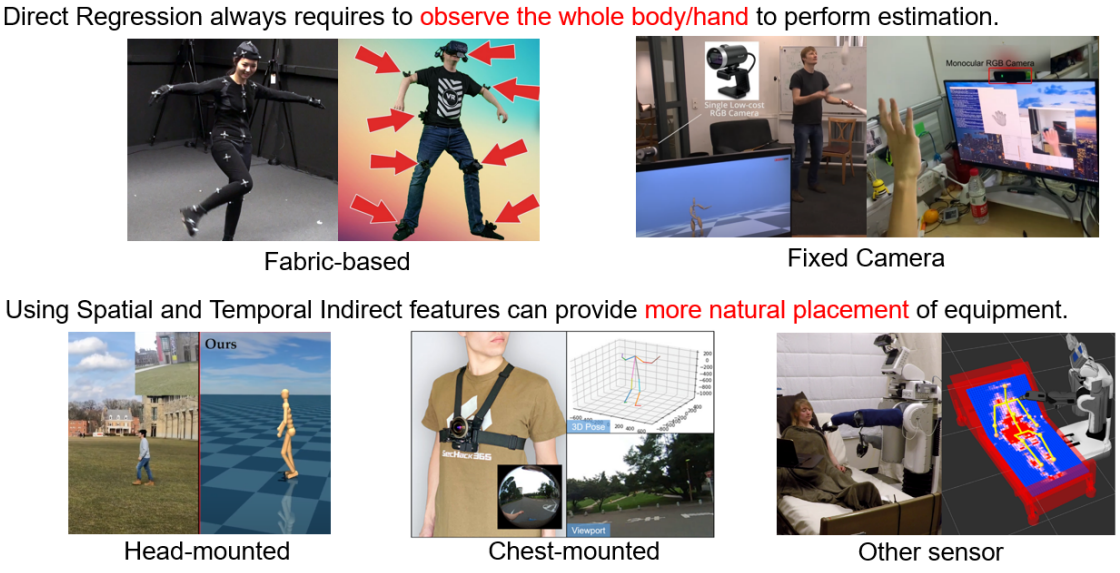


Figure 3.1: Problem of previous system

3.2 Research Approach

To solve the limitations of the related work mentioned before, we developed a novel indirect feature-based pose prediction network, Aligned with the three previ-

ous mentioned issues, our solution is also divided into three part. The solution are listed as follows:

3.2.1 Indirect Estimation for Natural Placement

For the issue of bulky equipment, it will be ideal if there is an almighty system that can estimate real-time 3D poses which is portable and does not disturb the user. However, such a device is technically not possible, as long as the user need to wear/carry extra devices, it has the potential to be bulky depends on what activity the user is doing. Under this situation, the current best solution is to put such devices At a more natural position that may not disturb the user based on the application.

For example, it is very difficult to perform real-time motion tracking for skiing. A fixed camera is out of the question since the position of a skier is changing rapidly, and fabric-based technologies are also less robust and have the potential to hurt the user while skiing. The only possible place to put such sensors are inside the ski boots. Therefore, a more "natural" and optimal solution will be using the feet pressure. Thanks to the development of tiny and long-lasting pressure sensor, nowadays there are many high precision feet pressure in-sole sensors. Also, previous work by Clever et al [13] already showed the possibility of regressing full body pose from pressure map, it is clear that such pressure value is indirectly related with human's posture. Similar approaches can be applied to many other skill transfers where only part of the body is able to be used for sensing.

3.2.2 Pose Estimation using Multiple Indirect Features

To improve the precision as well as the variety of applications, we want to build a dual module network which extract both spatial and temporal indirect features that can be adapted to different types of pose estimation. Puchert et al. [50] has already succeeded in combining a graph neural network with recurrent network to

extract two types of direct features, of which the network can be referred.

We propose two networks, FuturePoseNet for temporal feature regression and InvisiblePoseNet for spatial feature regression, each network consists of normal direct feature extraction layers with an indirect feature regression module, the details of these two network will be introduced in Chapter 4.

3.2.3 Comprehensive Studies on Training Applications

To show the effect of different types of indirect features and their combination, in this paper, we design several training applications for different skill using the proposed indirect network.

The first advance skill is alpine slalom skiing, which is a fast speed, massive, and dangerous sports, where common motion capture cannot work. Skiing pose estimation system using feet pressure mentioned before is developed to make use of the spatial indirect feature extraction. To better evaluate the performance in skiing, we also employed a stupendous motor-based skiing simulator which realistically reproduce the alpine skiing and being used by some national Olympics team. The simulator is used for both data collection and training performance evaluation to provide quantitative results on the pose estimation and its training effect.

Next, we focus on a more dexterous motor skill – playing the piano. This time, instead of the full body posture, we focus on the hand motion. The hand pose estimation is theoretically very similar to body pose estimation, but the finger hand less degree of freedom (DOF), therefore a hand poses can be easier represented with joint angle instead of keypoints. Since it is difficult to place markers directly on the hand during a performance, we pay attention to the back of the hand to obtain indirect features. As shown in Figure 3.2, different part of the dorsal part of the hand is changing when using different fingers. Since piano requires timing-perfect motion analysis, we also recorded the keystroke of the piano to extract indirect temporal features to support the prediction.

Last but not the least application in this paper is table tennis. Different from the

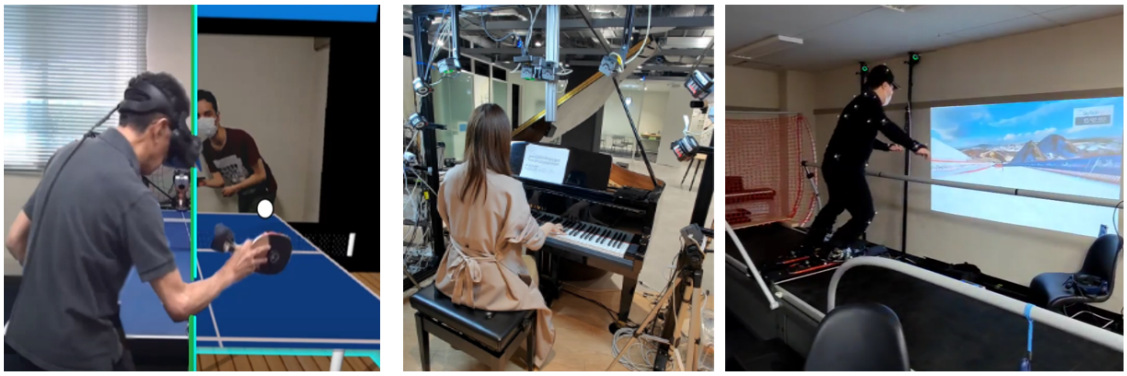


Figure 3.2: Skiing, Piano, and Table Tennis

previous two skill which is either massive or requires dexterous motion, table tennis is a skill which is relatively easy to begin, but difficult to master. Also, instead of pose prediction/estimation, in table tennis it is more important to predict the ball trajectory and its spin type. Therefore, in the last study we focus on how to return a strong spin serve by providing real-time prediction on the ball. The ball is predicted from both the temporal indirect features of previous frames, and the spatial indirect features of the opponent's motion. To realize a fair comparison, the users are asked to play against a pingpong robot before and after the training condition, to study their performances.

In terms of visualization methods, it is proved that the sense of immersive is helpful to skill acquisition [3], which is related to the concentration of the learner. Under that situation, XR technologies such as virtual reality (VR), augmented reality (AR) OR mixed reality (MR) might be a best way for 3D visualization for skill training. Different visual cues are used for the corresponding skill and are introduced in Chapter 7.

3.3 System Overview

Fig. 3.3 shows the overview of the proposed system. As mentioned before, the proposed indirect feature-based pose prediction system consists of two sub-network, the

FuturePoseNet and the InvisiblePoseNet. Different types of features of different skill are inputted to the two networks, respectively, to support the final body/hand/ball prediction. The trained model then are used for visualizing feedback for training beginners/learners in different types of applications.

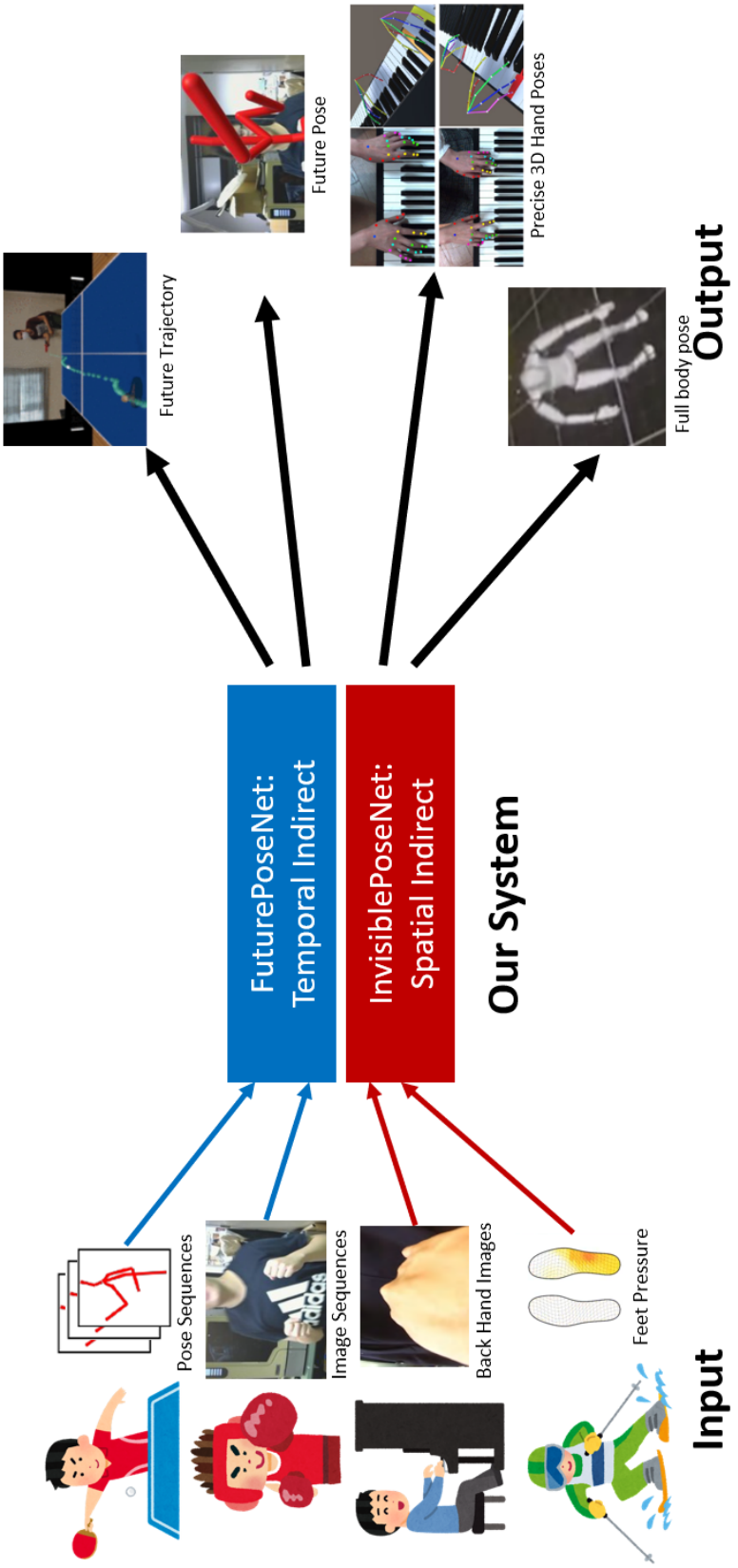


Figure 3.3: System Overview.

Chapter 4

Network Architecture

The network architecture and algorithms of deep learning are developing rapidly. In this chapter, we will first introduce the two common feature extraction networks and several common technologies which could be used for temporal and spatial feature extraction.

4.1 Basic Feature Extraction

4.1.1 Convolutional Neural Network

Convonlutional Neural Networks (CNNs) is a class of deep neural networks, which is commonly applied to analyzing visual imagery. CNNs are regularized versions of multilayer perceptrons and are on the lower extreme on the scale of connectedness and complexity. It is because CNNS used a different approach towards regularization: they take advantage of the hierarchical pattern in data and assemble more complex patterns using smaller and simpler patterns.

The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The final convolution, in turn, often involves backpropagation in order to more accurately weight the end product. [1]

A Convolutional layer is the core building block of a Convolutional Network that does most of the computational heavy lifting. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a simple architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a image has size of $W \times W$ will has W^2 weights for each neuron in the second layer of a fully connected network.

In comparison, the convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. Regardless of the image size is $W \times W$, if the tilling regions of size $H \times H$, each with the same shared weights, required only H^2 learnable parameters. An example of convolution an image with size 5×5 filtered by a 3×3 regions is shown as Figure 4.1: Next, we will introduce some representative convolutional neural networks.

VGG

The VGG network proposed by Simonyan et al. [59] also won the ILSVRC 2014 image classification department. The network structure, different from GoogLeNet which include some special layer, is very simple but deeper. There are two common types of VGG networks, the VGG-16 and VGG-19, where the number 16 and 19 stand for the number of layers. All convolutional layers are divided into 5 groups and each group is followed by a max-pooling layer. The only difference between them is that in the last 3 group of VGG-19 there are one more convolutional layer, as shown in 4.2. Since the network structure is quite simple, it is often used for fine-tuning to solve other problem than image classification.

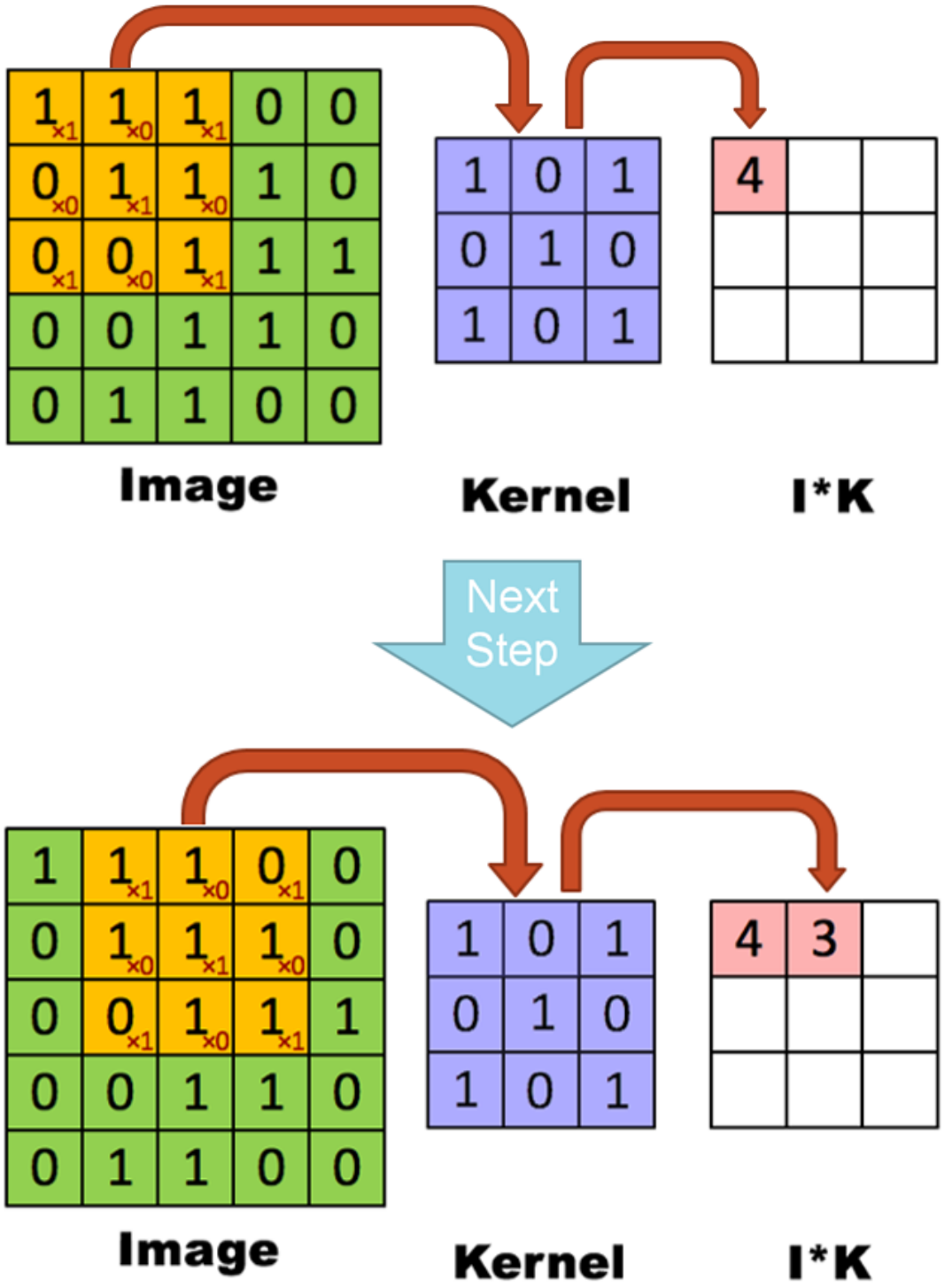


Figure 4.1: Convolution Computation

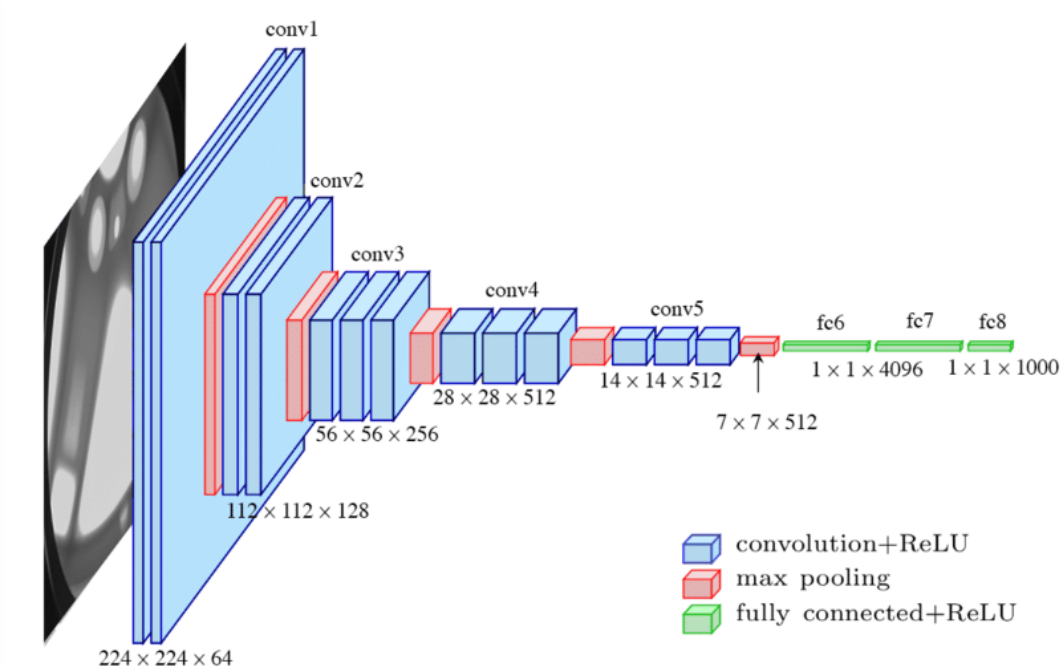


Figure 4.2: Fully Connected Layers

GoogLeNet (Inception)

As the winner of ILSVRC 2014 object detection department, the GooLeNet [62] increased the mean average precision of object detection to 0.439329, and reduced classification error to 0.06656, the best result to date. Their network has 22 layers, and approximately 12 times less parameters than AlexNet. Their inception model aims to bring deep learning also to some low-end processing unit such as smartphone. The idea of the inception layer is to cover a larger area, but also keep a fine resolution for small information on the images. As a result, their network is able to convolve in parallel different sizes from the most accurate detailing (1x1) to a bigger one (5x5).

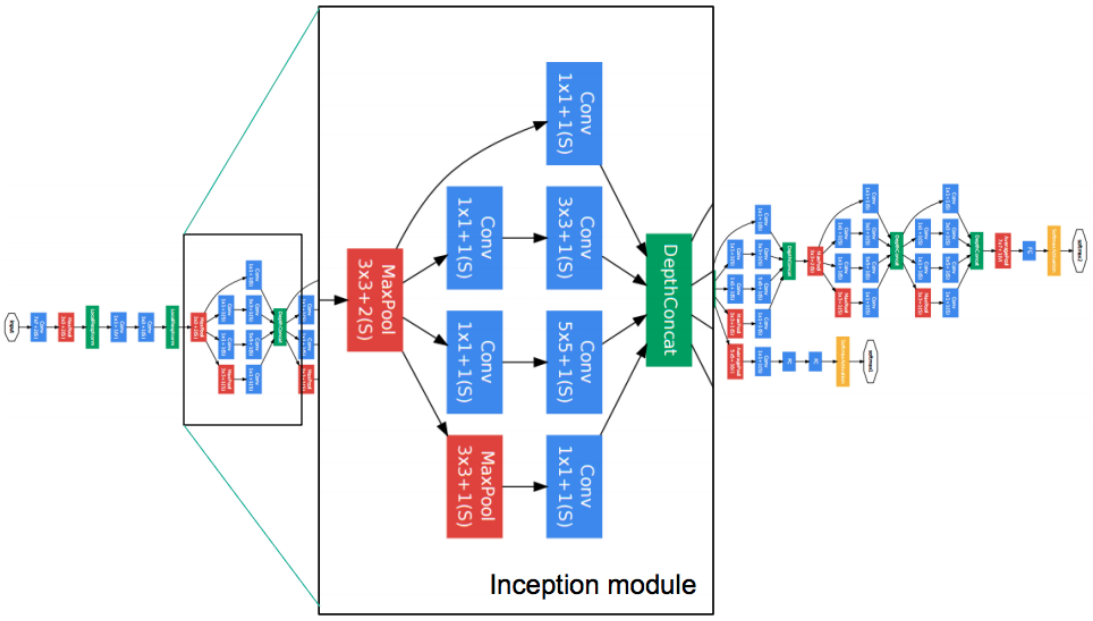


Figure 4.3: GoogLeNet (Inception)

ResNet50

ResNet50 [24] is introduced by He et al. from Microsoft Research as a residual convolutional neural network. It was the winner model of ILSVRC 2015. The biggest feature of its network is its very deep structure, with the 152 layers. A special techniques was used to make it possible to compute such a large network with great quantity of parameters, the residual block (Figure 4.4).

With the help of this special structure, the ResNet succeeded in having a lower complexity with a 8x deeper network than VGG.The identity mapping is multiplied by a linear projection W to expand the channels of shortcut to match the residual. This allows for the input x and F(x) to be combined as input to the next layer.

$$y = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}. \tag{4.1}$$

Equation used when F(x) and x have a different dimensionality such as 32x32 and 30x30. This Ws term can be implemented with 1x1 convolutions, this introduces additional parameters to the model.

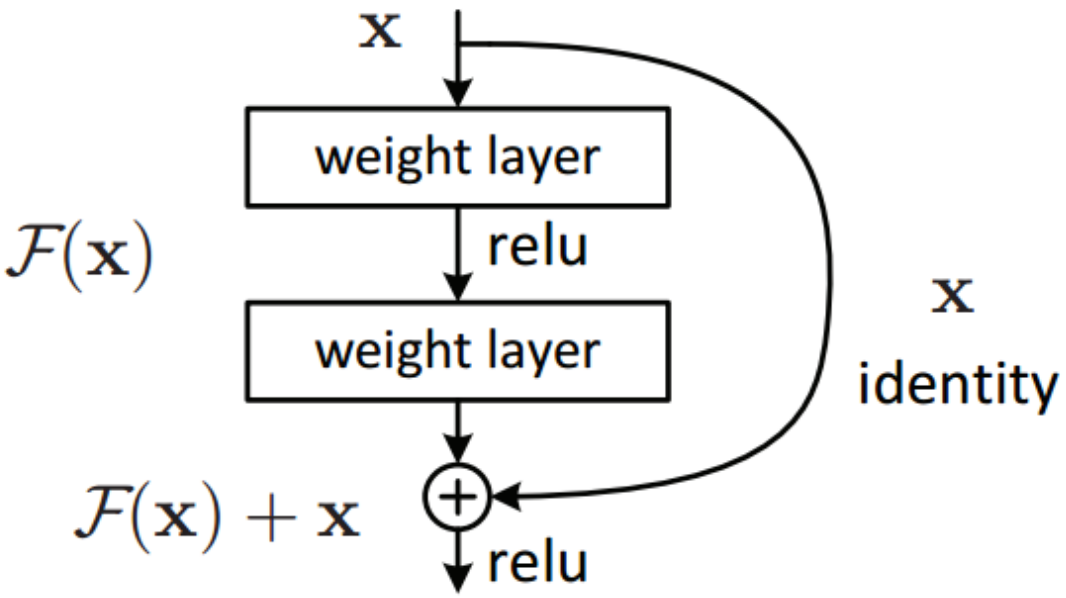


Figure 4.4: Residual block

4.1.2 Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition[1] or speech recognition.

The term "recurrent neural network" is used indiscriminately to refer to two broad classes of networks with a similar general structure, where one is finite impulse and the other is infinite impulse. Both classes of networks exhibit temporal dynamic behavior. A finite impulse recurrent network is a directed acyclic graph that can be unrolled and replaced with a strictly feedforward neural network, while an infinite impulse recurrent network is a directed cyclic graph that can not be unrolled.

Basic RNNs are a network of neuron-like nodes organized into successive "layers."

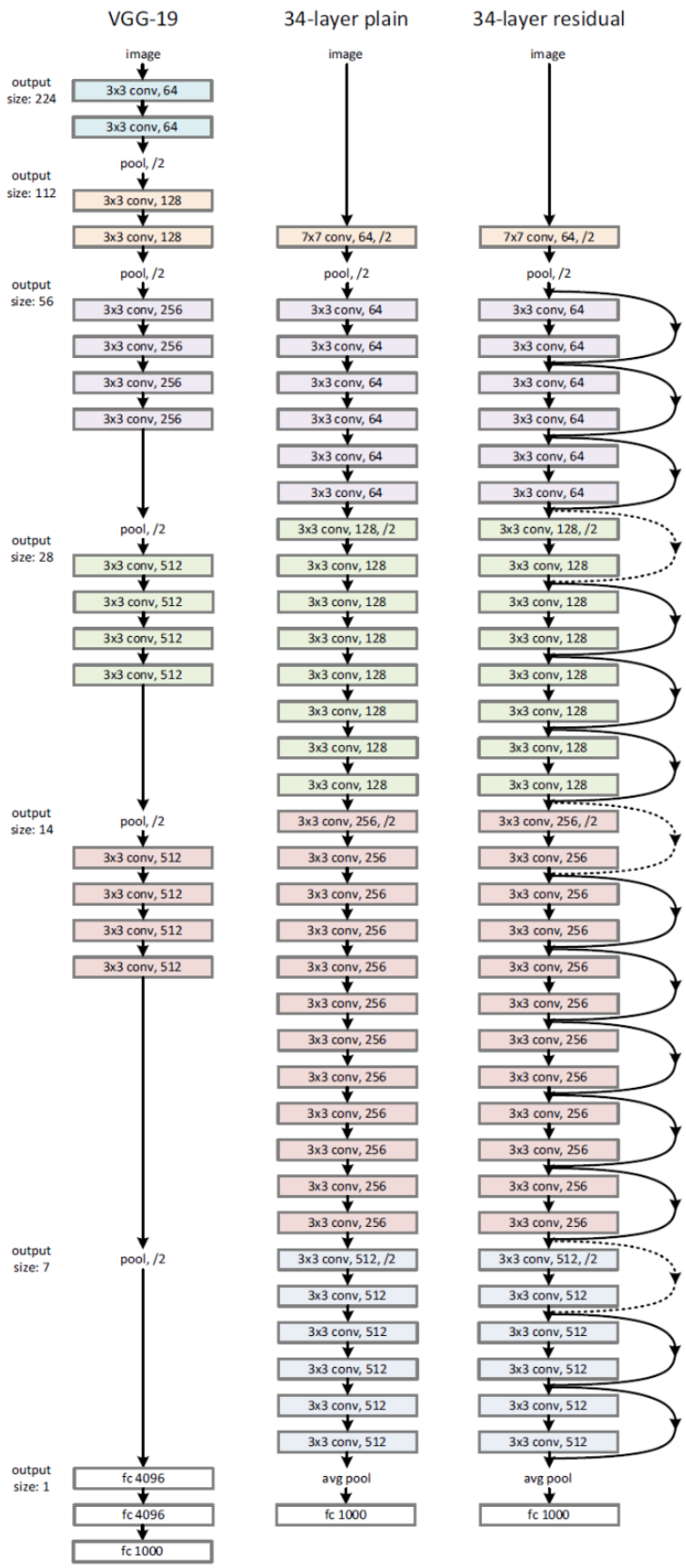


Figure 4.5: ResNet comparing with plane CNN and VGG.

Each node in a given layer is connected with a directed (one-way) connection to every other node in the next successive layer. Each node has a time-varying real-valued activation. Each connection has a modifiable real-valued weight. It can be thought of as multiple copies of the same network, each passing a message to a successor, Figure 4.6 shows how the loop is unfolded.

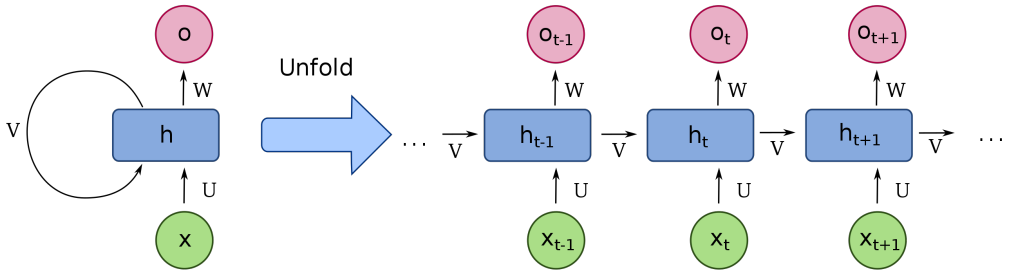


Figure 4.6: Recurrent architecture

As a result of these chain-like structure, output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, which is therefore weak at learning sequential or temporal features. Thus RNN came into existence, which solved this issue with the help of a hidden layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

Long short-term memory

One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task. However, it is already proved that basic RNNs don't have the ability of learning long-term dependencies. Long short-term memory networks (LSTMs) are a special RNN architectures which were introduced by Hochreiter and Schmidhuber [27]. LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing

machine can) [57]. Unlike standard RNNs, the repeating module of LSTMs is not a single neural network layer, but 4 layers interacting in a special way (as shown in Figure 4.7).

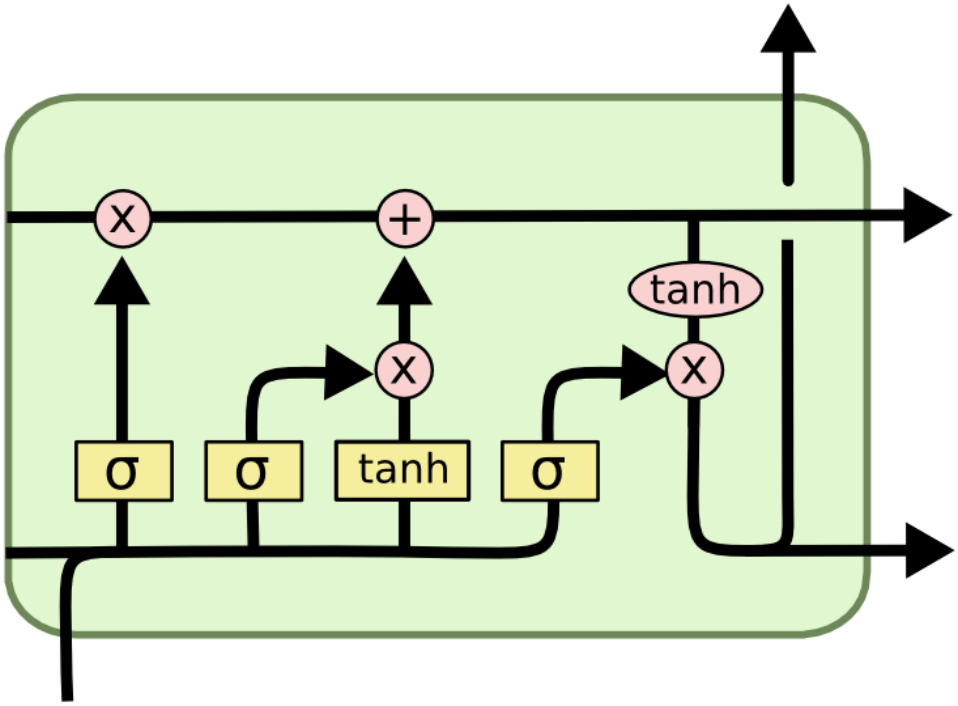


Figure 4.7: LSTM Architecture

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

4.2 Temporal Indirect Feature Extraction

4.2.1 Optical Flow

Lucas-Kanade optical flow method (LK-OF)[40] is the most famous algorithm for calculating the optical flow between 2 frames. The LK-OF assumes that the displacement of the image contents between two nearby frames is small and approximately constant within a neighborhood of the point p under consideration. Thus the optical flow equation can be assumed to hold for all pixels within a window centered at p. Namely, the local image flow vector (u, v) must satisfy the following equation:

where $q_1, q_2, \ldots, q_n$ are the pixels inside the window, and $I_x(q_i), I_y(q_i), I_t(q_i)$ are the partial derivatives of the image I with respect to position x, y and time t, evaluated at the point q_i and at the current time.

To solve the optical flow constraint equation for u and v, the Lucas-Kanade method divides the original image into smaller sections and assumes a constant velocity in each section.

Then, it performs a weighted least-square fit of the optical flow constraint equation to a constant model for $[u \quad v]^T$ in each section Ω. The method achieves this fit by minimizing the following equation:

$$\sum_{x \in \Omega} W^2 \big[I_x u + I_y v + I_t \big] \tag{4.2}$$

where W is an $n \times n$ diagonal matrix which is also a window function that emphasizes the constraints at the center of each section. The solution to the minimization problem is:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum W^2 I_x I_t \\ -\sum W^2 I_y I_t \end{bmatrix} \tag{4.3}$$

The result of LK-OF can be seen as bellow:

Figure 4.8: Result of Lucas-Kanade method, the upper figure refers to time T_n, while the lower image refers to time T_{n+1}.

$$\begin{aligned} I_x(q_1)u + I_y(q_1)v &= -I_t(q_1) \\ I_x(q_2)u + I_y(q_2)v &= -I_t(q_2) \\ &\dots \\ I_x(q_n)u + I_y(q_n)v &= -I_t(q_n) \end{aligned} \tag{4.4}$$

4.2.2 Motion History Image

Compared to a per-pixel dense motion estimation such as the above mentioned optical flow, motion history image (MHI) converts the 3D space-time information in a video sequence into a single 2D intensity image. The process needs first background subtraction to segment the foreground region in each individual image in the sequence. A foreground pixel is then assigned with a large fixed intensity value that represents the duration of an action. It is reduced over time by a small constant value when the pixel becomes a background point. The intensity value in the MHI thus records the history of temporal changes at each pixel location. The MHI is formally defined as [6]:

$$H_\tau(x,y,t) = \begin{cases} \tau, & \text{if } D(x,y,t) \in \text{foreground} \\ max\{0, H_\tau(x,y,t-1)-1\}, & \text{otherwise} \end{cases} \tag{4.5}$$

where $D(x,y,t)$ is a binary image that indicates the presence of moving objects at time frame t. The parameter τ critically defines the temporal duration of an action. If the preset τ value is smaller than the actual number of frames of an action, the prior movement of the action is lost in the MHI. Conversely, the changes of intensity values in the MHI become indistinct and residuals of previous unrelated motions are retained when the τ value is overly large. In the MHI representation, all detected foreground points (i.e., $D(x,y,t) = 1$) have the same intensity value τ , regardless of movement durations and moving speeds at individual pixels. It is thus very sensitive to background noise and cannot well describe local movements of a target object.

4.3 Spatial Indirect Feature Extraction

4.3.1 Graph Convolutional Network

Recently, generalizing the CNN to the graph convolutional network (GCN), which can handle arbitrary graph-structured data, has received widespread attention. The

GCN model has been successfully used in many applications, which also has the potential to extract indirect relationship in a image. (as shown in Figure 4.9

The GCN model constructs a filter in the Fourier domain, the filter acts on the nodes of graph and its first-order neighborhood to capture spatial features between the nodes, and then the GCN model can be built by stacking multiple convolutional layers. As shown in Figure 4, assuming that node 1 is the central road, the GCN model can obtain the topological relationship between the central road and its surrounding roads, encode the topological structure of the road network and the attributes on the roads, and then obtain spatial dependence. In summary, we use the GCN model [47] to learn spatial features from traffic data. A 2- layer GCN model can be expressed as:

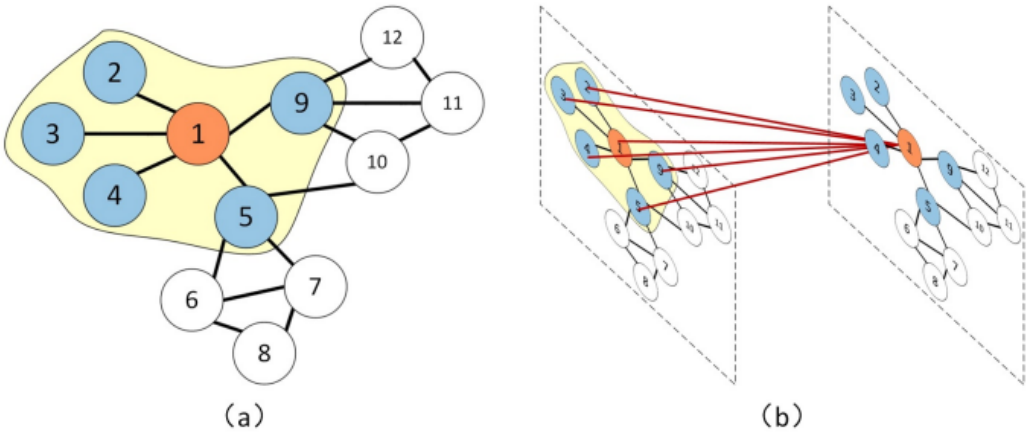


Figure 4.9: A Graph Layer

$$f(X, A) = \sigma(\hat{A}Relu(\hat{A}XW_0)W_1) \tag{4.6}$$

where X represents the feature matrix, A represents the adjacency matrix, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ denotes preprocessing step, $\tilde{A} = A + I_N$ is a matrix with self-connection structure, $\tilde{D}$ is a degree matrix, $\tilde{D} = \sum_j \tilde{A}_{ij}$. While, W_0 and W_1 represent the

weight matrix in the first and second layer, and $\sigma()$, Relu() represent the activation function.

4.3.2 Self-Attention

Self-attention network is designed to solve the problem that CNNs cannot process long-range relations and grasp high-level semantic information, which not only receive efficient features in a local region, but also perceive contextual information over a wide range. Therefore, it can be applied to indirect feature extraction.

As shown in Figure. 4.10, feature maps from the previous hidden layer are first transformed to three feature spaces (q, k, v). q indicates a query space vector while k is a key space vector, they are used to calculate weights which represent the similarity features between feature map. Reweighting the long-term information on the value space vector v enables the network to capture joint relationships easily.

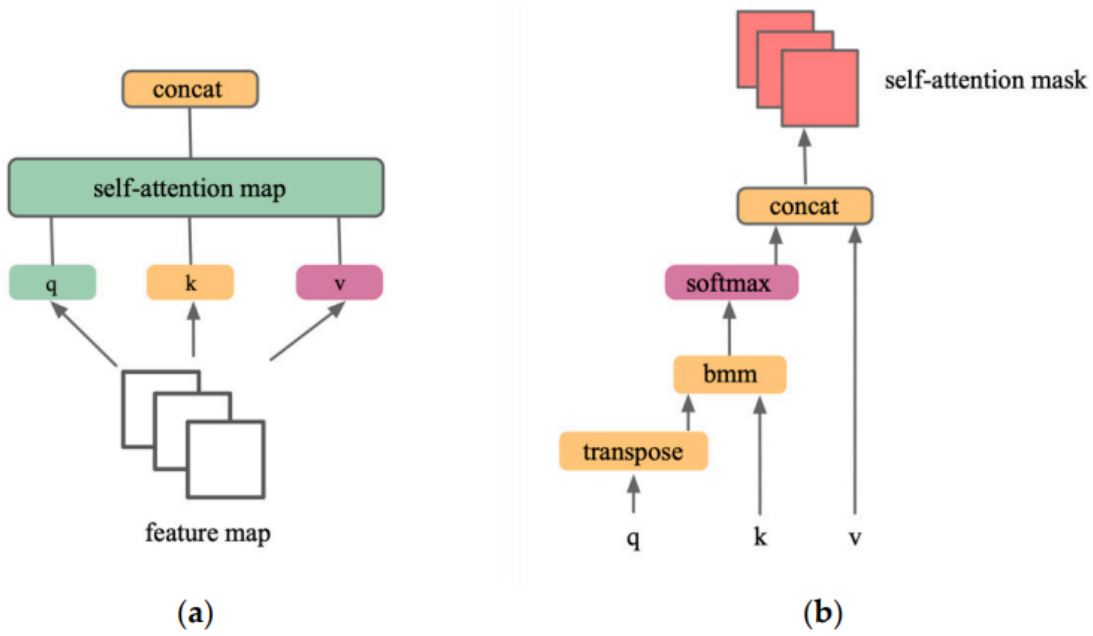


Figure 4.10: Self Attention Layer

4.4 Proposed Network Architecture

For the network architecture used in this system, to realize both direct and indirect regressions, we use a two-stream architecture [19, 41] to perform both extractions simultaneously. As shown in Fig. 4.11.

Both the spatial indirect feature extraction and the temporal one use the same network structure, the only difference is the algorithm used in the indirect feature extraction module and the regression layer in the indirect feature-network. In the next two chapter, a detailed explanation is given to the two types of indirect network, the FuturePoseNet (temporal indirect feature) and the InvisiblePoseNet (spatial indirect feature).

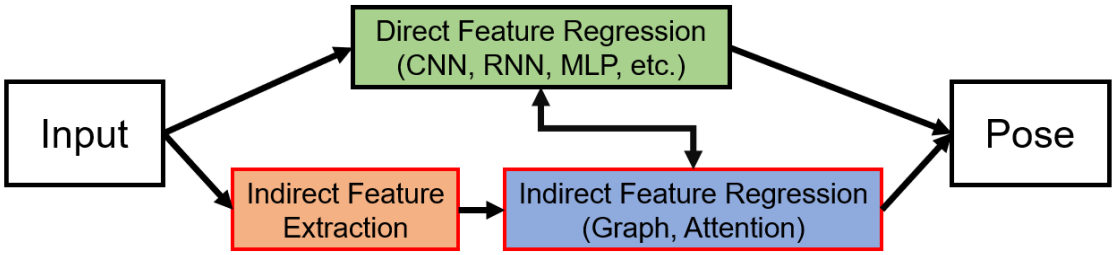


Figure 4.11: Proposed Two Stream Network Architecture

Chapter 5

FuturePoseNet (Temporal Prediction)

5.1 Overview

Based on the previous mentioned network design, we first developed FuturePoseNet, which focuses on predicting the future posture from the previous information, which is a very important ability in some sports. To realize this, we enhance the temporal indirect feature regression by proposing a new optical flow method

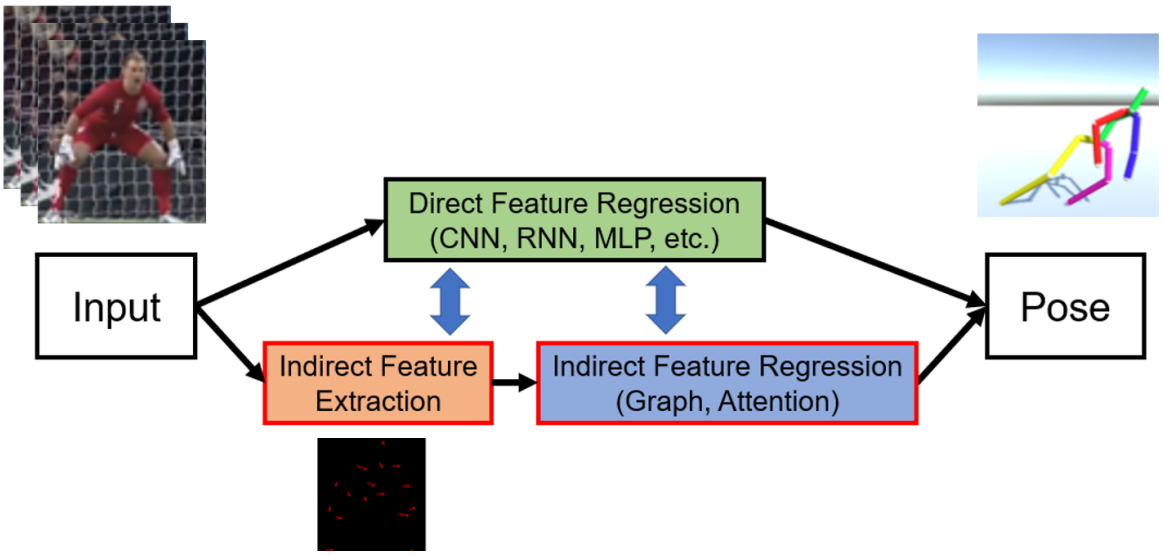


Figure 5.1: Overview of FuturePoseNet.

and using graph neural networks. The final layer is an LSTM layer to regress 3D poses from temporal features of the previous frames. The indirect module and the overview can be seen in Fig 5.1.

5.1.1 Keypoint Lattice-Optical Flow

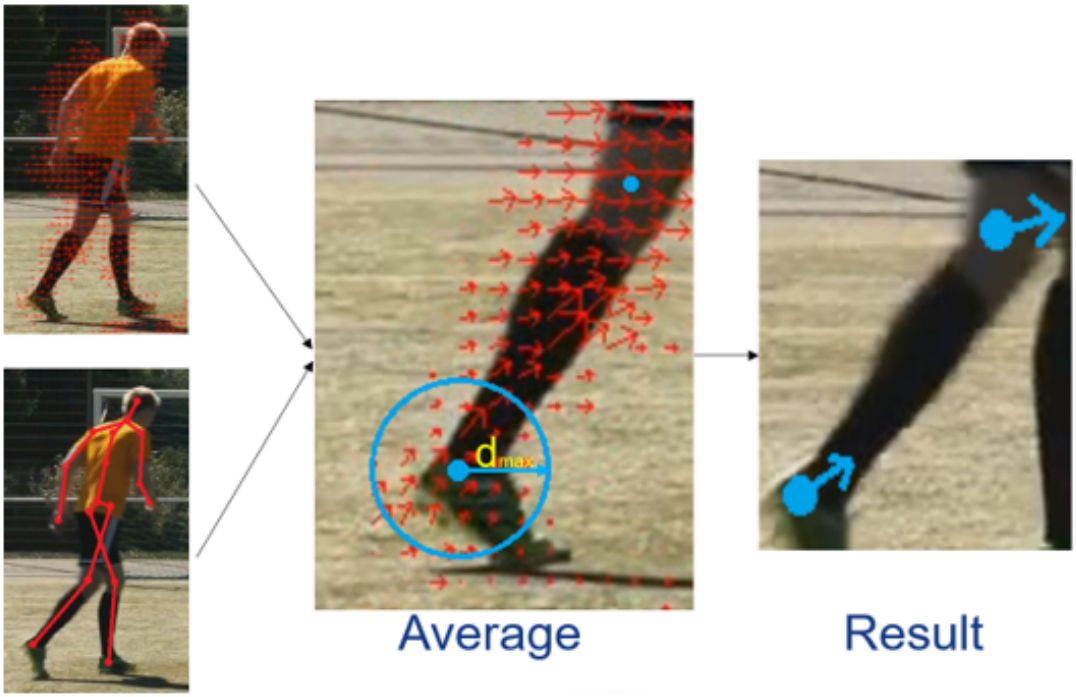


Figure 5.2: Our method of lattice point optical flow, sparse lattice points on human body are divided into several groups according to joint positions, while optical flow of each group of lattice points will be averaged to represent the optical flow of corresponding joint.

Since the LK-OF method requires a huge computation, we down-sample the image to 32x32 to make it possible for real-time calculation. From some pilot test, we noticed that the prediction result was not that good and we found that the motion feature wasn't extract totally by observing the optical flow result. It might be

cause due to the low resolution of the image, however, we found that in some image where the human body was cropped tightly which almost cover the whole image, the optical flow can detect the motion clearly.

Therefore, we developed a new type of sparse optical flow called Keypoint Lattice-Optical Flow (KL-OF), which creates several lattice points and only calculates the optical flows of those lattice points which are close to a keypoint (in our case, the human joint). First, we decide the space distance d of each lattice point to have a sum of $224/d \times 224/d$ lattice points, then we use the Lucas-Kanade algorithm [40] to calculate the optical flow vector of each point, defining the vector of the corresponding point (x, y) to be $LK(x, y)$. As a result, the computation of optical flow was reduced by at least d^2 times comparing with normal dense optical flow. Because the 2D joint estimation works in parallel and is faster (in most cases) than the optical flow calculation, we can obtain the 2D joint positions and directly determine the lattice points near the joints from their distance D_j.

In the following equation, (X_j, Y_j) stands for the joints position in 2D image. We can obtain the average optical flow Avg_j representing the movement of the joints as follows:

$$Avg_j = \{\frac{\Sigma_i^n LK(x_i, y_i)}{n} \quad | \ D_j < d_{max}\} \tag{5.1}$$

where $LK(x_i, y_i)$ is the specific optical vector and d_{max} is the maximum distance we use to average the optical flow near the joints. The parameters d and d_{max} need to be tuned for different applications to obtain higher accuracies; however, smaller d and larger d_{max} will lead to heavier computations. In our experiments, with cropped images of a size 224×224 as input, we used $d = 8$ and $d_{max} = 24$, which means that, at most, 28 lattice optical flow vectors are averaged for one joint, as shown in Figure 5.2.

The two graphs above show the comparison of applying LK-OF to a original image in 32x32 scale and applying our method to the same image. It is obvious that in our case the density of vectors are higher on human's body. On the other

Figure 5.3: Comparison of LK-OF with our method.

hand, since the amount of point is in most case even smaller than using normal optical flow, the computation is faster.

In conclusion, from the comparison we can know that our method provides more motion (temporal) information with less computation, which leads to a faster and precise prediction.

In the next chapter, an evaluation of inference time and accuracy will be performed to proof the effect of our method.

5.1.2 Graph Convolutional Layer

In addition to the LKOF methods, a graph layer is added to the indirect feature stream instead of conventional direct MLP. With the help of the final LSTM layer and the LKOF module, it is able to learn both the long-term and the short-term changes in the temporal sequences. However, the relationship between these indirect features and the target 3D posture is not well-learned by the network thus might be over-fitted to other features. Relationship between indirect features and the target output needs to be correctly learned. (For example, when estimating finger movement from the back of the hand, it is easy to know that the middle finger is somehow related with the middle part of the dorsal hand.)

Therefore, we refer to the Graph Convolutional Network [7] and include a graph convoltional layer in the indirect feature network.

5.1.3 Network Architecture

The whole network architecture can be seen in Figure 5.1.

The 224×224-size cropped RGB input image sequences, after a pooling layer, are divided into a temporal direct feature extraction stream and and an indirect feature extraction stream. Then the output of the feature extraction are further passed to an LSTM layer and MLP to enhance the temporal learning. Based on these feature extractions, the network finally predicts the temporal future posture of the person in the input images.

For the direct feature regression, we refer to the pose estimation network of

Dushyant et al. [46, 47]. They use a customized ResNet50 [24] to allow the convolutional layer to regress the 2D joint data and is trained on an annotated 3D human pose dataset such as the Human3.6M [9, 30] and MPI-INF-3DHP [44] dataset. We adjust the network to directly estimate the current 2D joints which will be further passed to the indirect instead of generating location maps for the depth estimation; therefore, the activation layer become a linear regression with dimensions of the number of joint positions. The residual network structure can be seen in Figure 4.5.

The indirect feature stream, as mentioned before, is first processed by our optical flow algorithm to obtain the motion vector for each joint and then further learned by the graph convolutional layer. The input for the optical flow are the current RGB image and the current 2D pose extracted by the direct feature stream, and will output a 17×2 2D joint vector which includes the movement information of each joint. This movement vector is finally passed into the graph neural network to learn the relationship between the previous joint and the future joint.

The output feature of the direct and indirect stream are stacked for five continuous frames and passed to the LSTM layer. The network learns both long-term and short-term relationship between frames and finally output the future 2D posture.

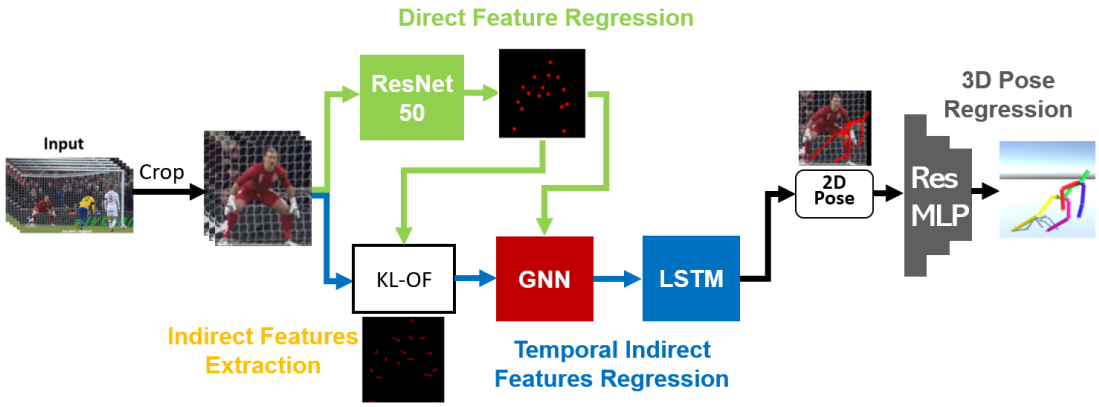


Figure 5.4: Network Architecture

The latter part of Figure 3 is a residual linear network which recover the 2D joints position to 3D. We refer the method of Martinez et al. [42], who developed an effective network for 3D pose recovery. After the forecasted 2D joint positions are output from the LSTM network, we apply a noise filter to the output coordinates. Even though we tried to use a Kalman Filter [34] or a Moving Average Filter [54], the noise was not clearly filtered and some correct joints were incorrectly filtered. After observing the data, we found out that most of the noise was radical errors which are completely wrong compared to the correct position. Therefore, we use a threshold filter, which only filters the joints that are away from the center of body for more than 70% of height.

The filtered data are then passed to the recovery network for the 3D construction. The network only consists of two linear layers and two residual blocks, which means that there are six linear layers in total.

5.2 Experiment on Pose Prediction

We performed our experiments from two different perspectives: quantitative evaluation and qualitative user study. In the quantitative evaluation, the real-time ability and the forecast accuracy was examined comparing with seven different methods including ours. While the qualitative user study asked some amateurs of martial arts to experience the system by receiving attack from a martial arts practitioner.

5.2.1 Dataset

For training the model, we used the sports motion from MPI-INF-3D and Human3.6M [9, 30] datasets for pre-training. Afterwards, for fine-tuning, we took data from 10 (8 male, 2 female) different subjects of 5 different motion: walking, side jumping, boxing, knee bending, and tennis swing. Each subjects did 10 sequences of each motion, while each motion is roughly 10 seconds. Which means, in total, for each type of motion there are approximately 3000 frames (100s) of video from each

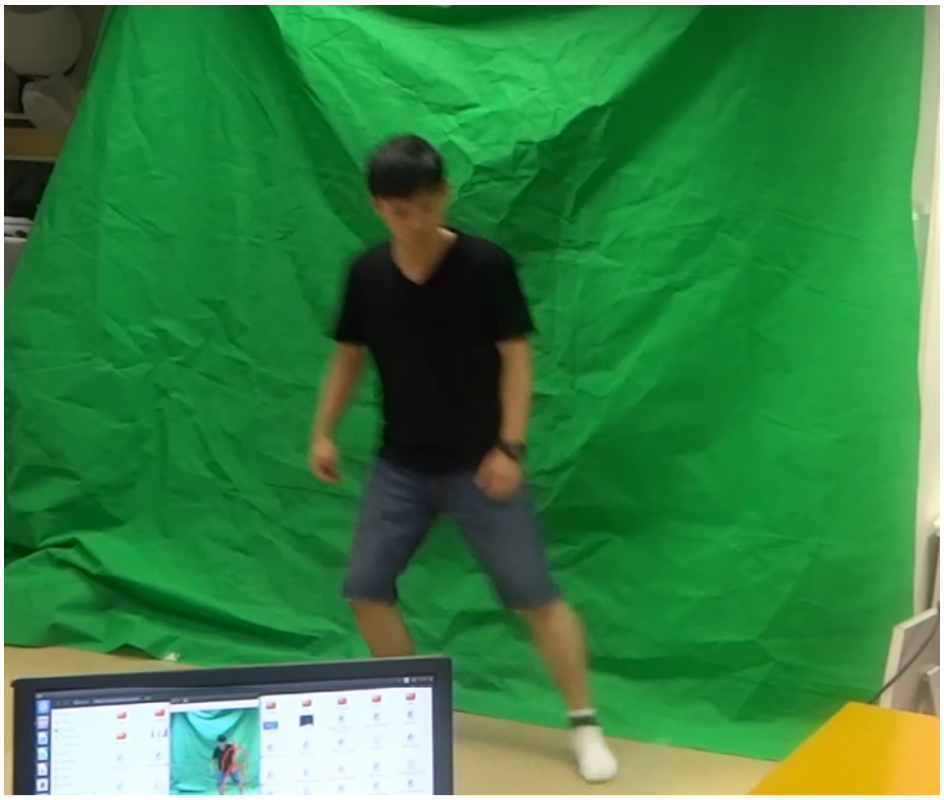


Figure 5.5: Quantitative Evaluation

subject. All the ground truth of 3D pose were taken by 2 KinectV2 depth camera in a green screen studio to make sure there is no occlusions or other noise. Also, to test the effect of our network working with online videos, we took 10 clips video of dance and penalty kick each of different people from youtube. In that case, the ground truth was given by the VNect [47].

All the data mentioned above was simply cropped before training, and are also split into a ratio of 8:1:1 for training, validation, and testing, respectively.

To test the real-time performance, since there is no necessity to differ from the training data. we simply used all the data from the MPI-INF-3D human pose [44] dataset. While estimating the accuracy is done by the test split of the data. To ensure the robust of the test, each estimation was done twice with the split to be shuffled. And the result will be averaged only when there is not a significant (less

than 5%) difference between the two times of testing.

5.2.2 Baseline

For a baseline, because there are few prior studies covering real-time pose forecasting on raw RGB frames, we first compared our method to other real-time prediction methods such as the five-layer neural network from Yuuki Horiuchi et al. [28] and normal convolutional LSTM [73]. We devised other baselines by changing the pose estimation module to VNect [24, 47] or Kinect [56, 80]. Further, we added the offline 3DPF-Net [10] model for comparison.

To evaluate the accuracy, we compared our method to both off-line approaches and real-time approaches. We also imported the baseline which is also used by Chao et al. [10] called Nearest Neighbor (NN), which uses the closest former frame to represent the prediction result of the predicted frame, as shown in Figure 5.10. We fine-tuned all the network using our practical dataset and used the test split data for the estimation. The Kinect method [28] was test with the same situation where the data are taken, a Kinect V2 camera was place to exact the same position of the RGB camera to perform the evaluation.

For the ground truth, the 2D ground truth are calculated from a 2D heat map regressed by ResNet100 [24] and the 3D ground truth are using the data from Kinect as a base architecture for a fair comparison.

5.2.3 Real-time Performance

We used multi-threaded programming for the image preparation (reading and cropping), pose estimation (including prediction), and visualization, which means the computing time of our system only depends on the most computational heavy part, the pose estimation.

To check the real-time ability, we examined the average inference/prediction time per image in milliseconds of our system compared to other methods which is shown

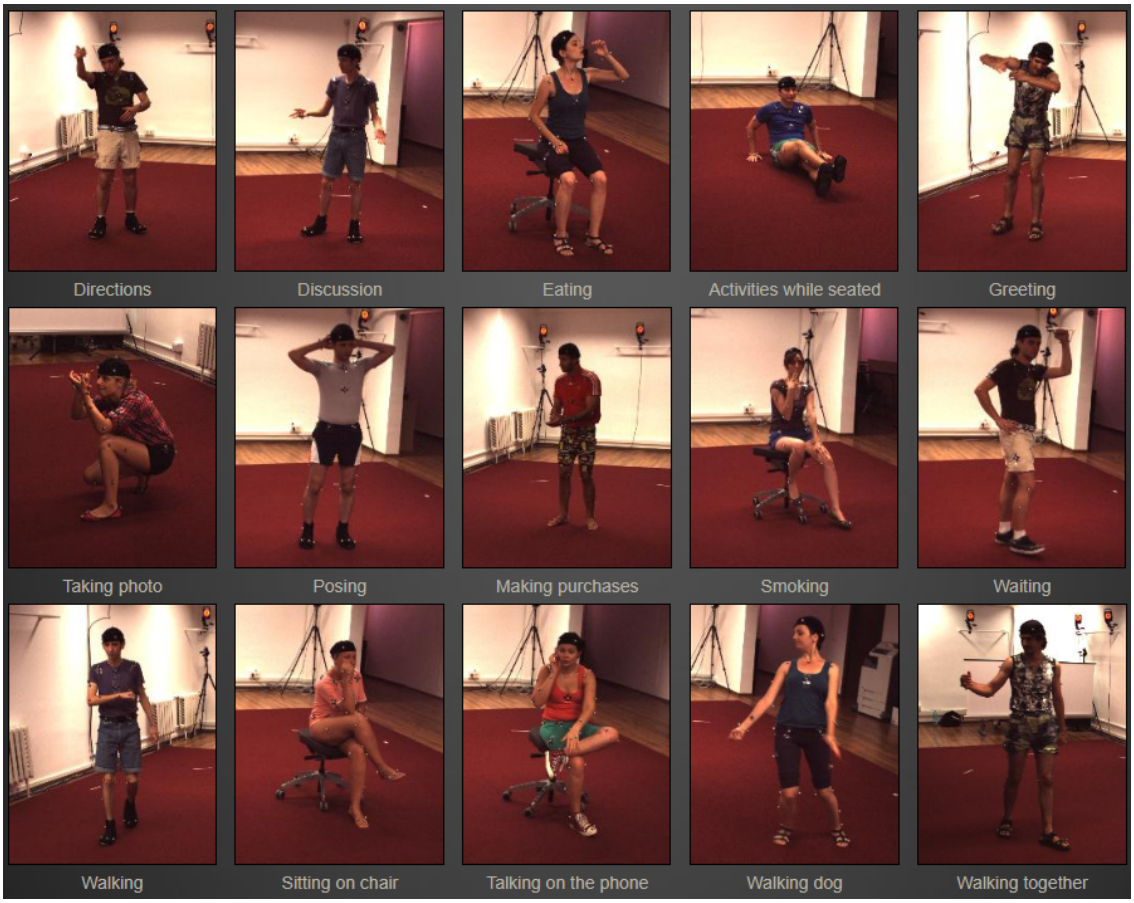


Figure 5.6: Human 3.6m

Figure 5.7: MPII dataset

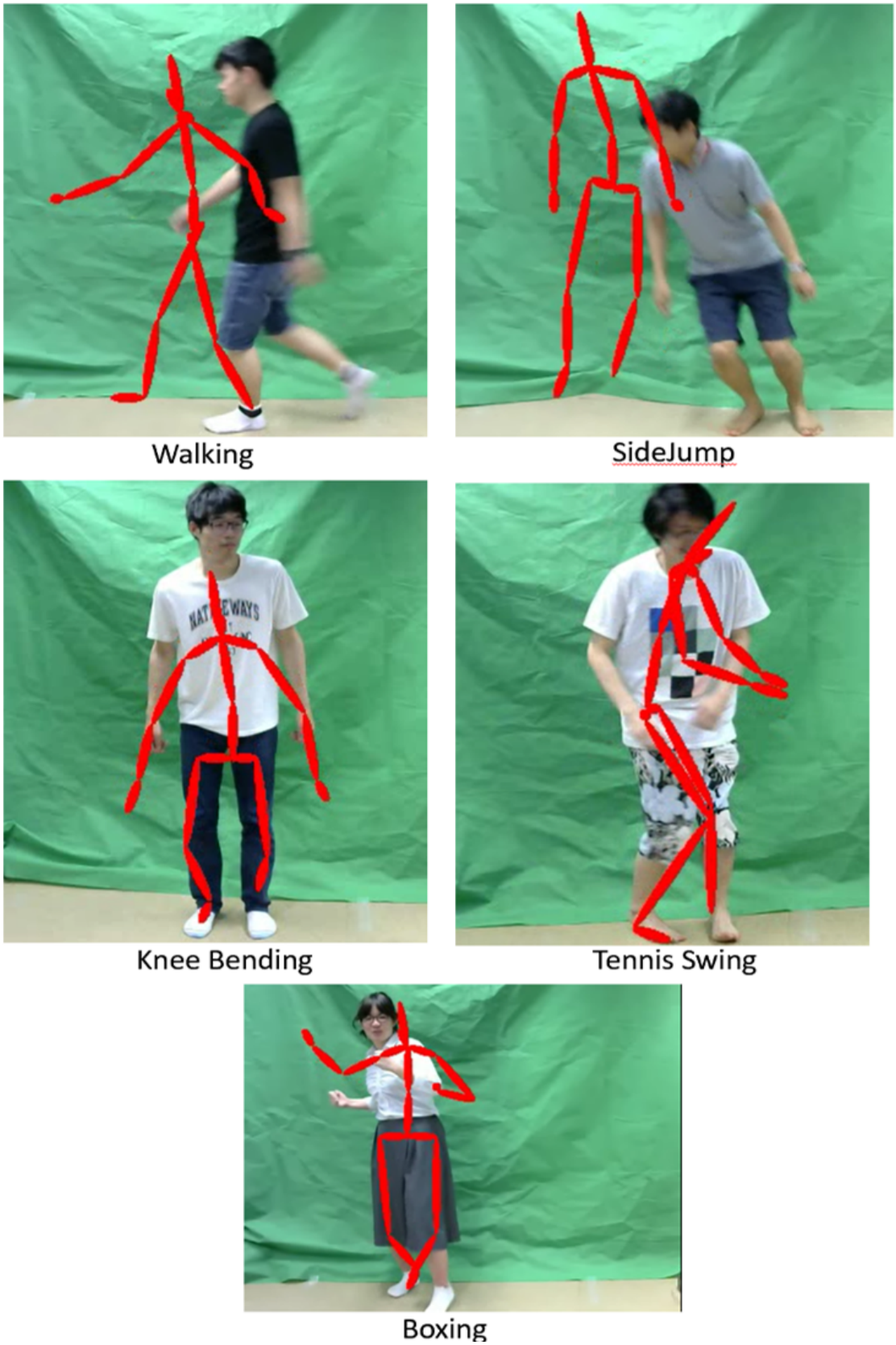


Figure 5.8: Our dataset

in Table 5.1. Note that, the input dimension of the neural network of Horiuchi et al. is 10 frames of data, which is twice that of our system and the LSTM method, and they also used the center of gravity (COG) as an input of the network. In addition, even though Kinect and VNect can generate 24 and 21 joints, respectively, we only used the same 17 joints for input as in our system.

Method	AIT(ms)
3DPF-Net (Offline) [10]	2500
Horiuchi et al. (Kinect) [28]	41.7
Horiuchi et al. (Direct)	55.3
Yagi et al. (LSTM)	39.6
Yuan et al. (LSTM+OF)	73.5
Ours (FC+LSTM)	42.1
Ours (GNN+LSTM)	40.0

Table 5.1: Average Inference Time (AIT) from one image being inputted till corresponding 2D forecasted pose being outputted of 30 test results (5 times for each type of motion).

The Table 5.1 shows the result of the evaluation, 4 baseline method and two types of our method (which is only different in stacking 10 images or 5 images as an input) was compared. Every method was test 5 times in 6 different type of motions, which means in total inference time of 30 trials was averaged. The definition of inference time in this study is the beginning of capturing image till end of getting the 3D output (shown in Figure 5.9).

From the result it is easy to know that the off-line network 3DPF-Net (which require more than 2.5 seconds for computing each image) is heavier in computation by 2 orders of magnitude than the other methods.

Comparing with other models with low inference time, the methods using Kinect appear to have the best performance because it uses RGB-D camera for the pose estimation. Convolutional LSTM is approximately 20ms slower than the neural network of Horiuchi et al., while our method has an approximately average level of performance. However, despite Kinect, which has hardware dependencies, our method did not fall far behind the neural network using VNect for the pose estimation with a frame rate of approximately 17 FPS, which is acceptable on a notebook without a high-end graphics process unit.

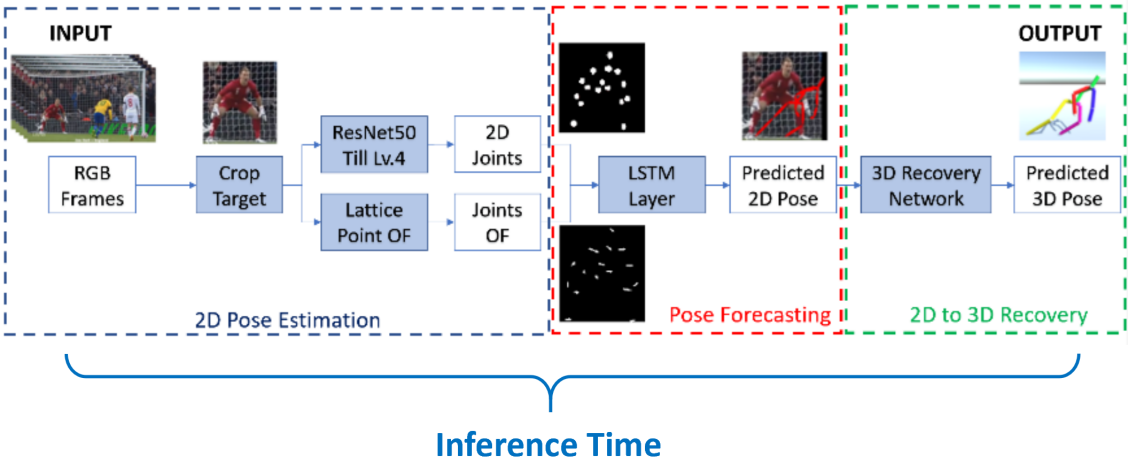


Figure 5.9: Definition of Inference time

5.2.4 Prediction Accuracy

For pose forecasting, we ran experiments predicting different time steps of future poses using different methods.

Real-time Forecasting Accuracy Result

The Table 5.2 in the next page shows the result of the accuracy test experiments in practical pose forecasting of 15 frames (0.5s in a 30-fps video) in advance. While Table 5.3 infers to the result of predicting 30 frames (1s in a 30-fps video) in advance.

For the evaluation, we used the PCKh@0.05 evaluation [4] measure which calculates the percentage of correct key point that uses a matching threshold of 50% of the head segment length. As mentioned before, the 2D ground truth of joint positions in these videos are calculated from a 2D heat map regressed by ResNet100 [24] as a base architecture for a fair comparison. The root-mean-squared error (RMSE) was also calculated to show the deviation of the predicted data.

In the 15-frame-forecasting test, the result of PCKh@0.5 (higher is better) and RMSE (lower is better) shows that our method performs better in most of the action (Unit of RMSE is pixel, 1 pixel is approximately 9.1mm in our experiments). Of which the result almost overcome or at least equal to the result of the 3DPFNet, which is the offline state-of-the-art.

In the 30-frame-forecasting test, all the result decrease except the nearest neighbor baseline. While the neural network method performs far worse than the LSTM and Our methods, which can proof the usefulness of the long-term network.

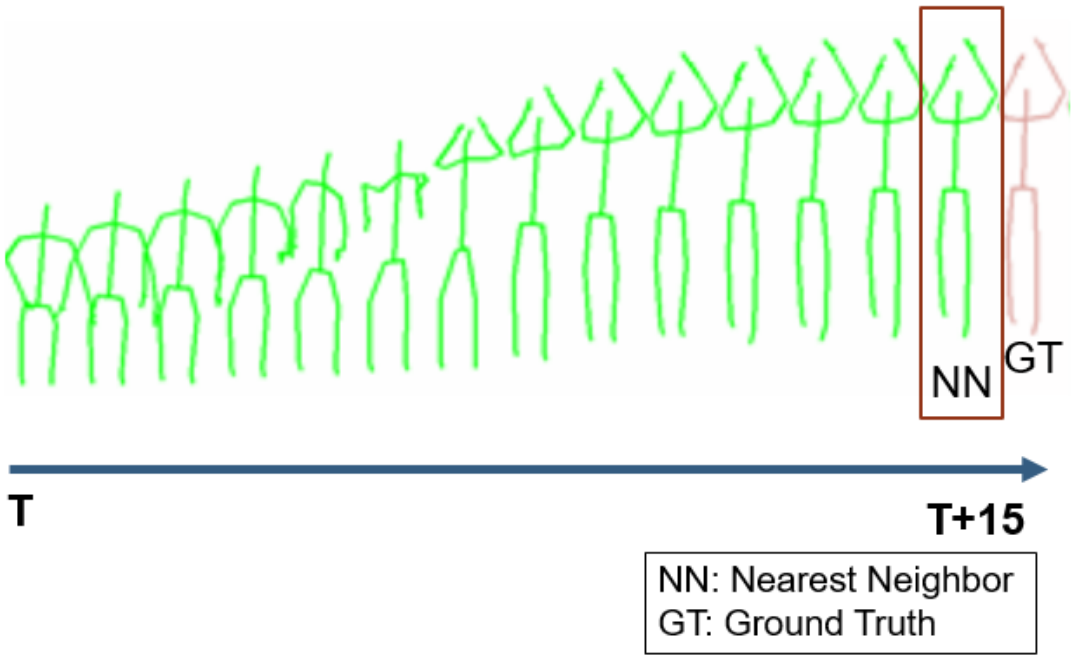


Figure 5.10: Nearest Neighbor

Method	SideJump		Walking		Boxing		Tennis	
	PCK	RMSE	PCK	RMSE	PCK	RMSE	PCK	RMSE
3DPFNet(off) [10]	42.5	8.1	60.7	7.9	68.2	7.8	61.4	8.0
NN-all [10]	39.1	12.2	58.1	8.9	60.3	8.4	52.9	10.0
Horiuchi et al. (Kinect) [28]	35.6	12.4	46.1	9.0	53.3	9.4	50.7	10.1
Horiuchi et al. (VNect)	33.4	12.3	44.4	9.8	51.9	9.4	49.7	10.3
Yagi et al. [73]	33.1	12.1	49.0	9.5	52.3	9.2	49.3	9.9
Yuan et al. [79]	42.1	8.8	59.4	7.8	66.7	7.9	55.5	8.7
Ours (w/o GNN)	42.2	7.9	60.9	7.0	70.6	6.8	61.0	**7.5**
Ours (w/ GNN)	**47.0**	**7.7**	**61.2**	**6.9**	**71.3**	**6.6**	**61.5**	**7.5**

Table 5.2: The PCK and RMSE result of predicting a 15-frame future from a 30-fps video.

Method	SideJump		Walking		Boxing		Tennis	
	PCK	RMSE	PCK	RMSE	PCK	RMSE	PCK	RMSE
3DPFNet(off) [10]	36.6	11.2	55.7	9.1	60.0	9.1	51.3	10.1
NN-all [10]	39.0	11.0	59.0	8.9	59.9	8.7	54.9	9.9
Horiuchi et al. (Kinect) [28]	29.0	13.4	38.9	11.0	42.6	10.4	39.6	11.5
Horiuchi et al. (VNect)	28.4	13.2	37.7	11.8	42.9	10.7	40.3	11.3
Yagi et al. [73]	31.5	12.6	45.7	10.2	49.9	9.4	46.3	10.0
Yuan et al. [79]	38.0	9.8	57.0	8.5	57.9	8.9	54.4	9.0
Ours (w/o GNN)	39.2	9.9	57.8	8.9	65.0	7.9	56.2	9.0
Ours (w/ GNN)	**45.9**	**9.4**	**60.0**	**7.9**	**67.0**	**6.9**	**60.2**	**7.7**

Table 5.3: The PCK and RMSE result of predicting a 30-frame future from a 30-fps video.

3D Joints Accuracy Result

The Table 5.4 shows the RMSE of specific joint (head, neck, chest, spine, shoulders, elbow, wrist, hips, knees, ankles, and torso were calculated.) of the 2D to 3D recovery method. Comparison was only done in the 15-frame-forecasting condition with the 3DPF-Net off-line method which performed the best in our accuracy experiments except our method.

For the evaluation, we calculated the difference between the forecasted 3D pose and the 3D ground truth which is taken by Kinect, and calculate the RMSE (the unit is pixel, 1 pixel is approximately 9.1mm in our experiments).

From the result, we can know that, even though the 3DPFNet had a smaller error in predicting 5 parts(Head, Nect, Chest, Shoulder and Hip) of 3D joints, our method have a better average score and is better in predicting the 3D position of the limbs such as Wrist or Ankle (of which the error is more than 10% lower).

Boxing

Method	Head	Neck	Chest	Spine	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Avg
3DPFNet(off) [10]	**72.9**	**64.7**	**62.8**	**65.3**	**64.1**	91.7	134.4	**57.9**	89.6	112.7	85.4
Ours(Stack 10)	73.0	75.1	68.7	**65.3**	65.4	**87.1**	**99.7**	63.6	**85.7**	**98.4**	**80.1**

Table 5.4: Root-square-mean per specific joint position errors (mm) of timesteps 15. Our system achieves a lower average error than the off-line 3DPF-Net.

5.3 Experiment on Ball Trajectory Prediction

Next, to show more potential of temporal indirect estimation, we apply the FuturePoseNet to table tennis. Instead of human posture, we believe there is also a temporal relationship between the served ball and the previous posture of the server. Therefore, with some adjustment in the input and output, we tuned the network to predict the pingpong ball trajectory served by an opponent.

The main difference of the network is that we crop the input video to the upper body and change the output of the 2D joint position to 10×2 (10 keypoints). Since the camera is placed in the front of the player (as shown in Fig. 5.11), it cannot see the lower body, which is covered by the table tennis table. Thus, only the 2D upper body joints positions (10 joints) are estimated.

The rest of the network are similar, the 10 joint positions are further passed to the indirect feature extraction and the LSTM to obtain temporal information. Ten previous poses are stacked as an input, which results in an input size of $10 \times 10 \times 2$. The output of LSTM is then passed to another 2 fully-connected layers, of which the final output is the 2D landing position (a 2D vector).

An overview of the real-time ball trajectory prediction system is shown in Fig.5.11.

5.3.1 Dataset

To collect the data for training, we used another 240-fps camera to track the precise trajectory of the ball for ground truth. However, the data are down-sampled to 30-fps for training to meet the real-time condition. The skeleton data are generated by the same residual CNN network shown in Fig.5.12.

For data annotation, in order to label the landing point of a serve, we also used the audio data. Since we performed the data collection in a practically silent environment, the rebound sound can be simply filtered by a amplitude threshold to acquire the bouncing frame which result in the landing point.

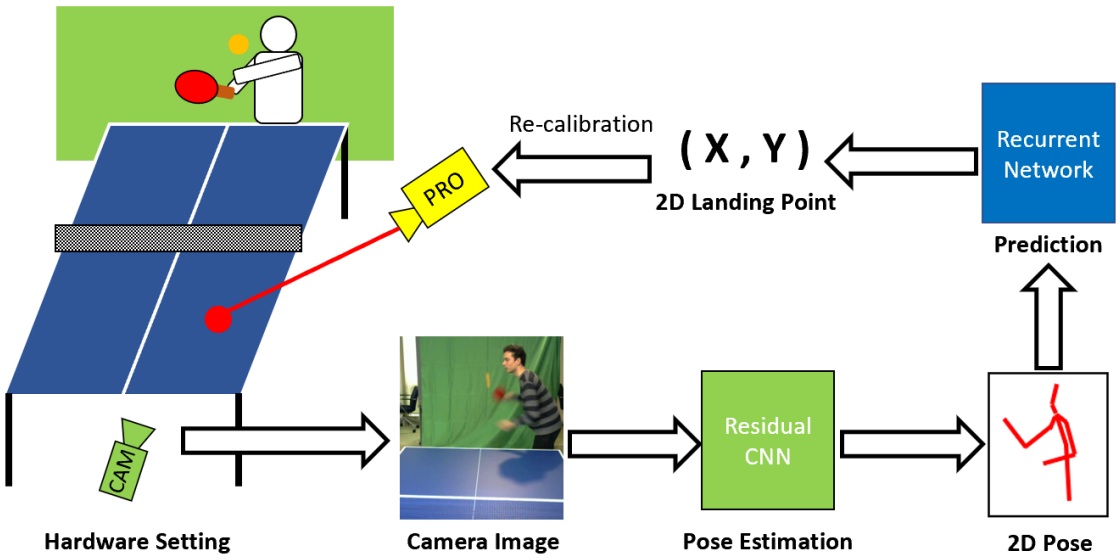


Figure 5.11: System Structure Overview

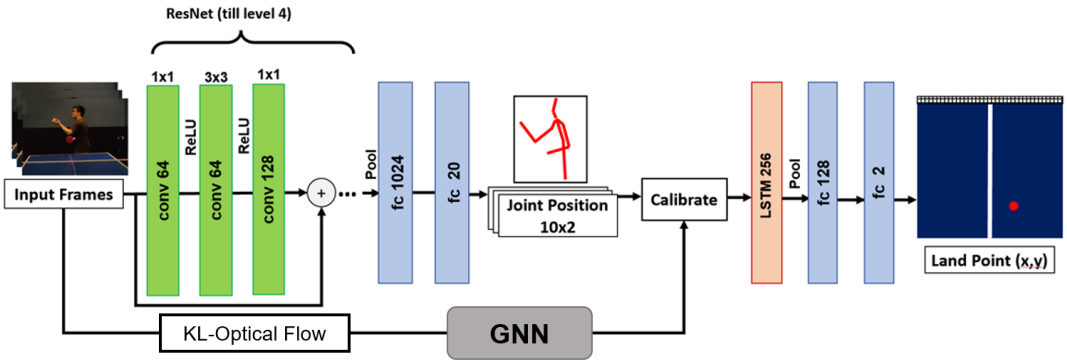


Figure 5.12: Network Architecture

We calibrated the 2D human posture according to the table position, where the center of the baseline was set as origin. For the ball position, the 2D position related to the table of the second rebound (obtained from audio data) considered as the landing point of a serve. Nevertheless, no rotation data are considered in this study.

Prediction Result of Curved Serve

Truth Result of Curved Serve

Figure 5.13: Result of Curved Serve

5.3.2 Prediction Accuracy

We performed two types of experiments: (1) a quantitative evaluation on forecasting accuracy, and (2) a qualitative study which visualize the result for beginners.

In the accuracy evaluation, the pose estimation network was pre-trained with MPI-3D [45] and Human3.6M [31] dataset while the LSTM network was pre-trained with 300 clips of table tennis serve gathered online. These online clips are manually trimmed from table tennis instruction video where the coach is making different serves with the camera in the front. After that, we collected data of 8 subjects (4 amateurs, 4 practitioners, all right-handed) doing 20 successful straight serves, which results in total of 160 video clips. for each subjects, we use 16 clips (80%) of the data for fine-tuning and the remaining 4 clips (20%) for testing. The entire data was shuffled randomly across person and tested in 3 conditions (Only amateurs; Only Beginners; Mixed) to study the robustness of our system. The data starts from when the subjects release the ball and ends right before they hit the ball (we call this part serve motion). Besides the straight serve data used for evaluation, curved serve from an expert player was also taken for attempting (Fig.5.13).

The results of accuracy tests are shown in Table 5.5. Condition A stands for amateurs only condition while condition P stands for practitioners only, Mix means the condition with data of all 8 subjects. The Percentage of Correct Point (PCP) shows the percentage where the predicted point is within the diameter of a pingpong ball (40 mm) of the ground truth. The results show the expected difference (12.5%) between the average PCP accuracy of the amateurs (81.25%) and the practitioners (68.75%), while the mixed condition is 75.0%. Among all the condition, the max error is only 8.9 cm.

5.3.3 Qualitative Results

In the qualitative study (Fig.5.14), we invited 6 table tennis amateurs only basic experience in table tennis to return the serve of an experienced player (with 10-

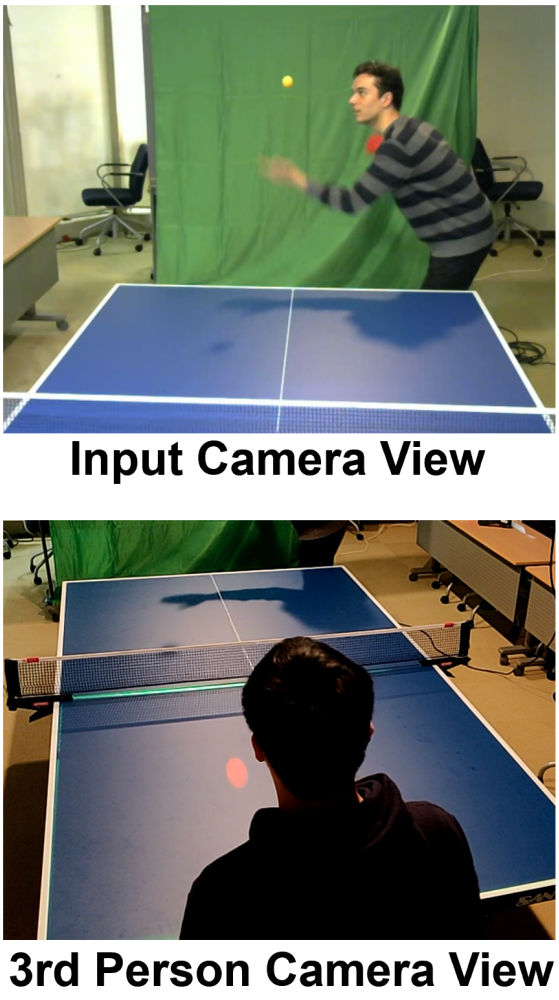


Figure 5.14: User Study Camera View

Cond	PCP	RMSE	Max
A	81.25	2.34	6.5
P	68.75	4.24	8.9
Mix	75.0	3.29	8.9

Table 5.5: Result of Forecasting Accuracy (Error unit: cm), PCP: Percentage of Correct Point, Max: Max difference.

year experience). The experienced player made 20 serves in each condition with or without the future visualization. The participants were asked to compare the experience W/WO the future visualization, and an interview were given to ask the overall impression of the prediction result.

From the interview afterwards, 5 participants stated that the system predicted the trajectory precisely and increased their interest in learning table tennis. 3 participants also claimed that the forecast was helpful to train the form of return, since it allowed more time to think about how to return the ball. However, two participants claimed that the visualizing was sometime disturbing and attracted the attention from the ball, a more intuitive feedback is demanded.

Chapter 6

InvisiblePoseNet (Spatial Prediction)

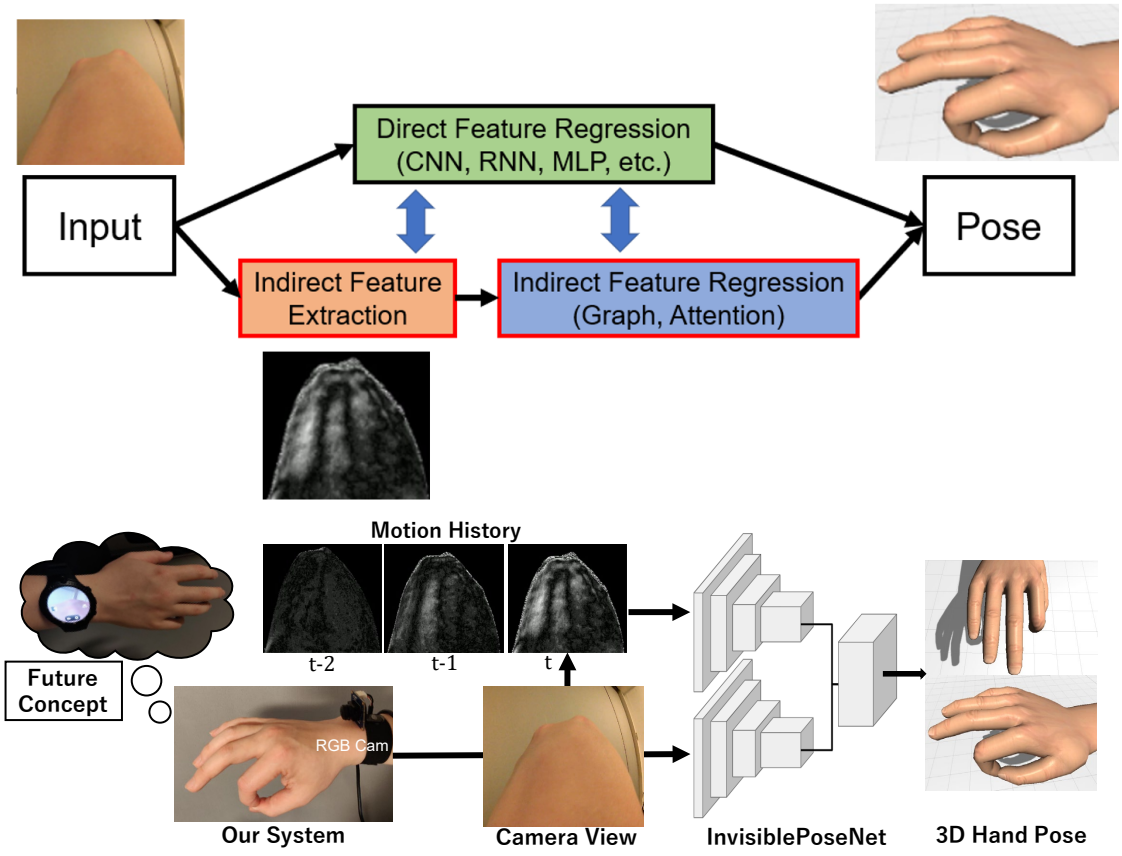


Figure 6.1: Overview of InvisiblePoseNet.

6.1 Overview

Spatial prediction is another advanced ability which enables human to understand the whole characteristics only from a part of an object. We propose the InvisiblePoseNet, which is trying to realize this spatial feature extraction using indirect features. Different from the FuturePoseNet, the InvisiblePoseNet focuses on predicting the whole body/hand posture from another part of the body which is not directly related with the target posture. To realize this, we enhance the spatial indirect feature regression by employing an optical flow-based motion history image and another self-attention neural networks for spatial feature extraction. The final layer is an graph layer to extract the relationship between specific regions of the input images and the target posture. The indirect module and the overview of the InvisiblePoseNet can be seen in Fig 6.1.

6.1.1 Optical Flow-Motion History Image

In the conventional MHI, every detected foreground pixel is assigned with a fixed intensity value τ .A slow movement and a fast movement of different body parts will have the same motion strength. Tsai et al. [66] introduced a spatio-temporal representation, where they combine optical flow and MHI. Similar to their idea, we also use the optical flow length $s(x, y)$ to represent each individual pixel (x, y) over time. The resulting intensity value then indicates the historical motion speeds at that location. It can better describe local movements of a target object. The optical flow itself is also used for foreground segmentation to extract moving objects. The motion duration in the conventional MHI is critically determined by the fixed parameter value of τ. The proposed OF-MHI (optical flow-motion history image) representation can be defined as:

$$E(x, y, t) = s(x, y, t) + E(x, y, t - 1) \cdot \alpha \tag{6.1}$$

where $s(x, y, t)$ represents the optical flow length of pixel (x, y) at time frame t.

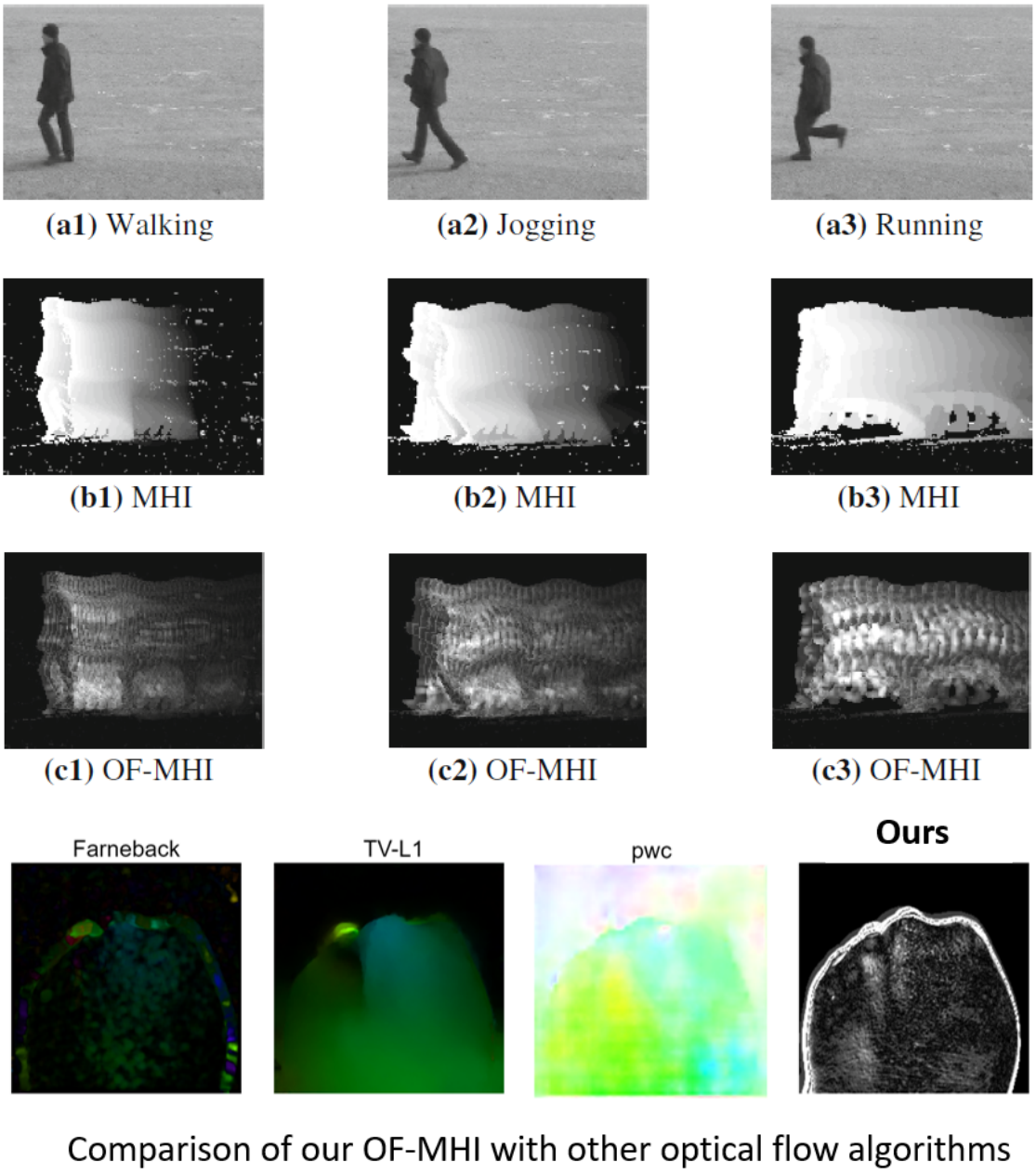


Figure 6.2: Spatial indirect feature extraction of optical flow-based motion history image.

The parameter α is the update rate, with $0 < \alpha < 1$. Note that the motion strength is adaptively given by the flow length $s(x, y, t)$ for each individual pixel (x, y). If the optical flow length $s(x, y, t)$ is very small, it indicates pixel (x,y) is a background

point.

6.1.2 Residual Kalman Filter Layer

In addition to the LKOF methods, a residual Kalman Filter layer is added to the final LSTM to enhance the feature extractions. This is because human's motion are considered to be linear movements in sequential video clips, which could be regularized by a Kalman filter (KF) [34]. However, KF require a motion model and measurement model to be specified a priory, which are often only crude approximations of reality. In the work of Coskun et al. [14], they introduced a LSTM-based KF to use LSTM to learn the motion and noise model, which shows promising effect on learning human dynamics. Therefore, this architecture is imported to obtain a more stable temporal feature sequence $\psi_{1:T}$. We also add a residual connection to bypass the Kalman filter for more direct feature learning, so the network will choose whether to use Kalman filter based on the target motion.

6.1.3 Network Architecture

Here, we use hand poses as an example to explain the network architecture and the purpose of each component. Figure 6.3 shows the overview of a network extracting indirect features on the back of the hand to predict full hand 3D finger posture.

For each training sequence of length T (in this paper, we use T=5), the preprocessed data consists of the masked hand images $I_{1:T}$, the opitical flow-based motion history images (OF-MHI) $X_{1:T}$, and the hand pose labels $y_{1:T}$. Each hand pose y_t includes the joint angles $\alpha_t^1, \alpha_t^2, \alpha_t^3, \alpha_t^4$ of the index, middle, ring and little fingers and the 3D position e_t of the thumb top. As shown in Fig. 6.4, the joint angle α_t^i of each finger has four elements (M_v^i, M_h^i, P^i, D^i) where M_v^i, M_h^i correspond to the vertical and horizontal rotation of the first joint and P^i, D^i correspond to the rotation angles of the second and third joint respectively. Our goal is to learn a neural network based regressor $\tilde{y}_{1:T} = f(I_{1:T}, X_{1:T})$ that maps the indirect features

in the input masked images $I_{1:T}$ and OF-MHI $X_{1:T}$ to a sequence of estimated hand poses $\tilde{y}_{1:T} = f(I_{1:T}, X_{1:T})$.

To this end, the InvisiblePoseNet, a two-stream graph convolution-based network whose architecture is outlined in Figure 6.3 is proposed. For each timestep t, two ResNet18 [23] are used to extract visual features from the masked hand image I_t and the OF-MHI X_t respectively. The two visual features are then concatenated together and passed through a fully-connected layer to form a unified visual feature ϕ_t.

Previous research [77] already showed that simple two stream CNN is not sufficient for extracting temporal features of the back of hand. Thus, we use an graph convolutional layer to process the visual feature sequence $\phi_{1:T}$ into a graph-based feature sequence, which is proved to be useful in indirect pose estimation [70, 79].

On the other hand, we noticed that most of our finger motions are simple linear movements, which could be regularized by a Kalman filter (KF). However, KF require a motion model and measurement model to be specified a priory, which are often only crude approximations of reality. In the work of Coskun et al. [14], they introduced a LSTM-based KF to use LSTM to learn the motion and noise model, which shows promising effect on learning human dynamics. Therefore, this architecture is imported to obtain a more stable temporal feature sequence $\psi_{1:T}$. We also add a residual connection to bypass the Kalman filter for more direct feature learning, so the network will choose whether to use Kalman filter based on the hand motion. For each frame t, the temporal feature ψ_t now includes information from past frames to help make hand pose predictions. Finally, another fully-connected layer is added to map the temporal feature ψ_t to the estimated hand pose $\tilde{y}_t$. We use a single LSTM instead of the three from the previous work [14], because the two stream CNN architecture is heavy in computation, we focus on light-weighting the whole networks to achieve a real-time inference time. That is also the reason why ResNet18 is used but not deeper CNN architecture such as ResNet50 or ResNet101. As a result, the inference time of the whole network using the mid-range notebook

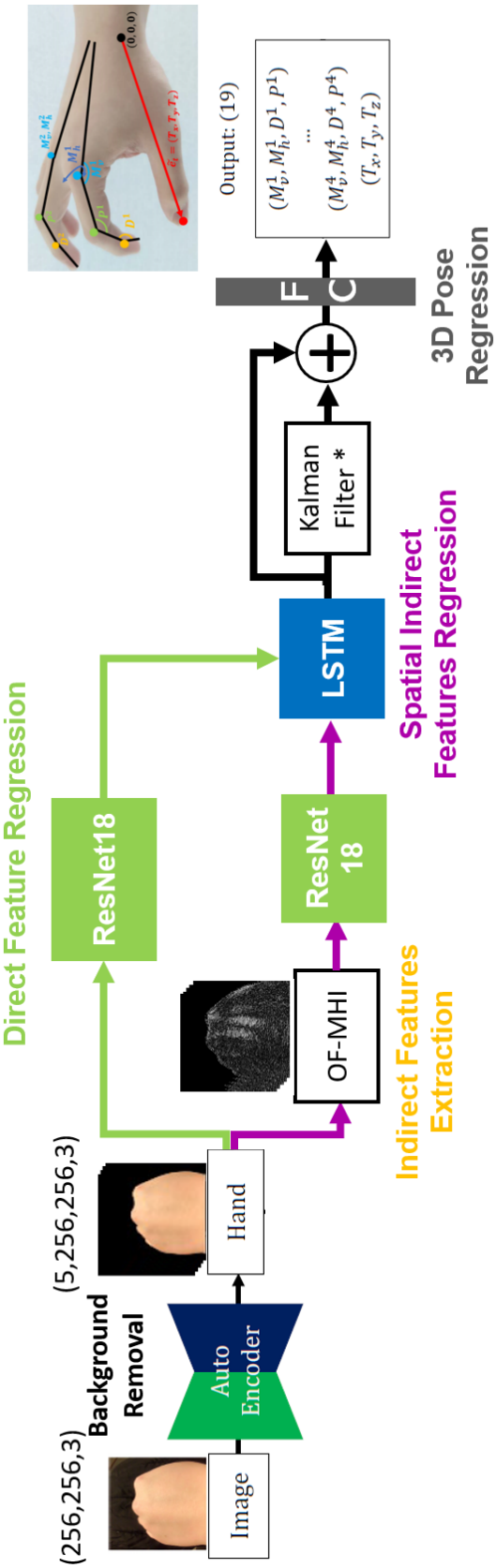


Figure 6.3: Network Architecture of InvisiblePoseNet

PC mentioned in the hardware section is approximately 38ms. To provide supervision for training the DorsalNet, we define the following loss function:

$$L(y_t, \hat{y}_t) = L_{\text{fingers}} + L_{\text{thumb}}\,, \tag{6.2}$$

$$L_{\text{fingers}} = \frac{1}{16}\sum_{i=1}^{4}\left\|\alpha_t^i - \tilde{\alpha}_t^i\right\|^2\,, \tag{6.3}$$

$$L_{\text{thumb}} = \frac{1}{\pi^2}\arccos^2\left(\frac{e_t \cdot \tilde{e}_t}{|e_t||\tilde{e}_t|}\right)\,, \tag{6.4}$$

where we use symbols with tilde to indicate it is the estimated output of the network and symbols without tilde to indicate ground truth. We also use different losses for the fingers and thumb because their pose representations are different. For the fingers, we use mean squared error (MSE) as the loss for the joint angles as shown in equation (6.3); for the thumb, we compute the angle between the estimated thumb top vector and the ground truth one as the loss function (6.4).

6.2 Experiment on Back Hand Pose

First, we examined the InvisiblePoseNet by applying it to predict full 3D hand poses from images of the back of the hands. The ultimate goal is to extract spatial indirect features on the dorsum of a hand, such as the deformations of skin, veins, or tendons, and learn the indirect relationship between these features and the motion of each finger.

For 3D hand representation, instead of location-based 3D coordinates, we use the relative joint angle-based representation [32] for the 4 fingers except the thumb, which is independently estimated by end point position (as shown in Figure 6.4, M, P, D stand for the MCP, DIP, PIP joints of the specific finger, while the v and h stand for the vertical and horizontal bending of MCP). For the thumb, it is more difficult to detect the relevant deformations since they mainly take place on the side of the arm. After a number of trials, we decided to treat the thumb separately and to let the network learn to estimate a 3D vector of the thumb top from the

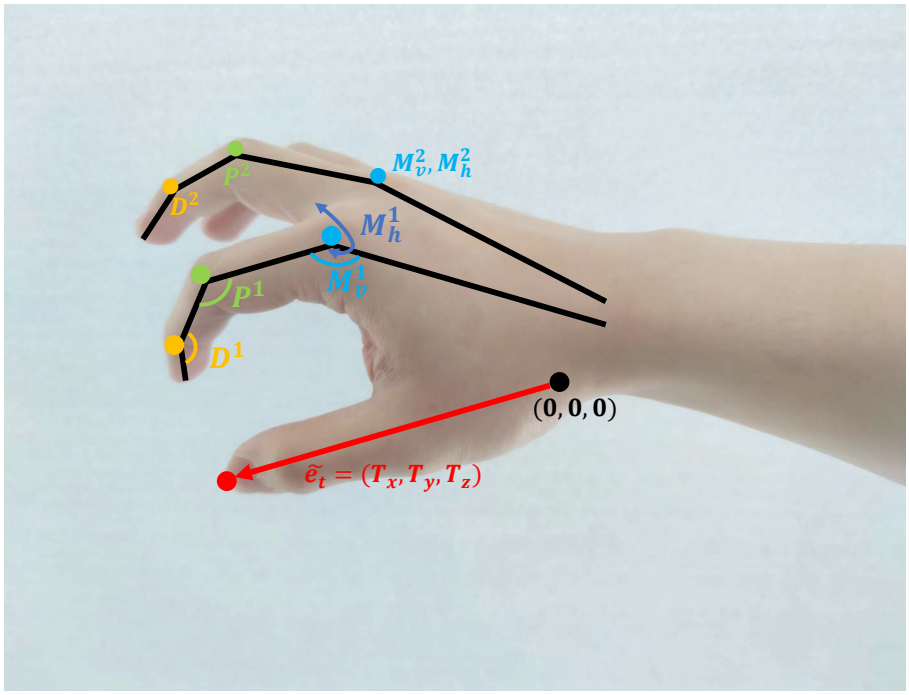


Figure 6.4: Our 3D hand model representations, the thumb is represented by a single 3D vector and the other 4 fingers are using joint angle.

edge information of the dorsal hand, and we then recover the thumb joint angles using inverse kinematics [39]. The joint angle error we use is another commonly used metric for 3D hand pose estimation which is widely employed [67, 81].

6.2.1 Data Collection

Data was collect from 5 out of the 6 participants who also participated in the previous segmentation study. For hardware, we use a wide angle RGB camera (as shown in Figure 6.5), that has less environmental restrictions and is more likely to be found in smartwatches than IR cameras (used by previous work [77]), which suffers from stray infrared light from the sun. However, RGB cameras, different from IR cameras, cannot benefit from the easy segmentation of removing the background. Therefore, we perform a hand segmentation to reduce noise from the background.

Similar to prior work [77], we collected data of static gestures of American sign

Figure 6.5: Comparison with commercial smartwatch, (top) the hardware comparison and (bottom) the cropped images from our camera and the raw images from both Zeblaze.

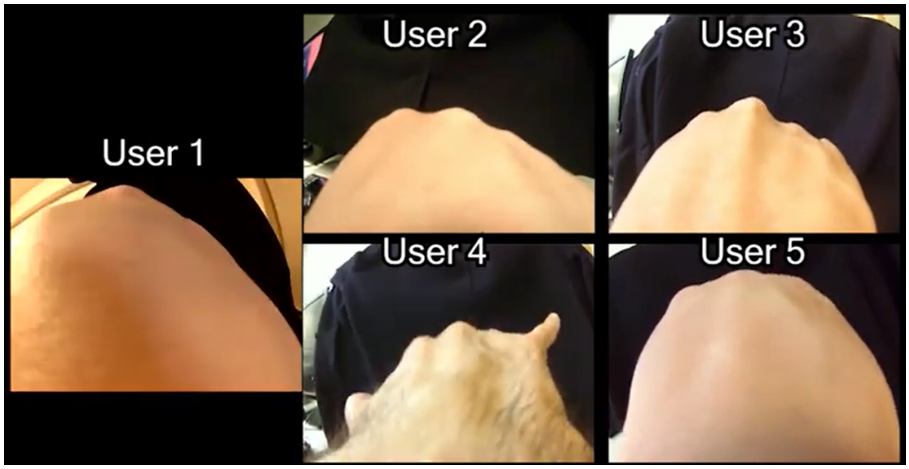


Figure 6.6: Examples of the data from the 5 subjects.

language (ASL) digits (0-9), and dynamic gestures of finger tapping. During the study, the participants were asked to put the right arm on an armrest and to wear our camera with Velcro tape to perform the action.

The entire collection procedure includes 5 sessions for all 15 gestures (both static and dynamic). In each session, the user was told to re-wear the camera and start from a relaxed hand posture to do the specific gesture repeatedly (for static gestures, the user have to return to relaxed posture every time). We asked the users to perform the gesture in a normal speed but the frequency is controlled by themselves, approximately 1 ASL gesture per 3 seconds and 1 tap per second were collected.

An auto-labeling program is written for multi-threading the Leap Motion API and camera image acquiring, where it is calibrated so that the root of the thumb becomes the origin, as depicted in Figure 6.4. We also fix the camera frame rate to 20 FPS to simplify the synchronization and to align with the inference frame rate. For each session of each gesture, 30 seconds of video at 20 FPS was collected. As a result, video of 600 frames was collected 5 times for all 15 gestures for each participant, which resulted in a total of 225,000 frames. These data were used in the training and evaluation for the hand pose estimation and gesture recognition. For the grasp recognition, we only use the mentioned data for pretraining, but use another dataset for fine-tuning and evaluation (which will be described in later sections). Also, we collect a single-user dataset of different lighting condition which will be described in the *Lighting Condition Study*. In all sections, the ratio of the train and test data split is set to 8:2.

To obtain robustness and usability, we performed several preprocessing steps including data augmentation, hand segmentation and motion image processing. Because a wrist-worn device is not always tightly fixed to the arm, the camera could have some slight rotations around the arm. Therefore, for each input image, we augment the data by rotating the image clockwise with varying angles from -10° to 10° with step size 5°, with the same ground truth. This resulted in 5 times amount of data to enhance robustness across device locations.

Hand segmentation is undertaken by fine-tuning an encoder-decoder network [5] to generate hand masks. In our setup, as the camera is fixed on the arm and looks directly at the dorsum of hand, the bottom half of the image is mostly occupied by the hand. Thus, it is relatively easy to mask the hand from background. As a first step, data for segmentation was collected from 6 participants (one female, aged between 25-30) across races of East Asian, Mediterranean, and European. All participants are students from the computer science department of two universities from different countries. They were told to wear our device and walk naturally inside a laboratory for about 2 minutes which results in 9,600 images being collected.

All images are then masked by color range and contour using OpenCV. Afterwards, the brightness and hand color of these images are changed for data augmentation. For each image, in the HSV color model, the H value is increased/decreased by a random value which generated 10 different color image including the original, and the brightness (v) value is also changed to -20%, -10%, 10%, and 20% for each image. As a result, we train the auto-encoder to create the hand mask of 480,000 images (50 times the amount of the original data). We randomly split the dataset into training (80%) and testing (20%), and the mean precision of the test result of generated hand mask is 98.9% in pixel scale.

As mentioned before, temporal motion images are required for training the two-stream network, which should be generated using pairs of adjacent frames. We explored the common Dense Optical flow (OF) (KV-L1), Lattice OF [70], PWC-Net [60] used in Ego-Pose [79], and motion history images (MHI) [6] used by Opisthenar [77]. Since the deformations need to be captured in a pixel-perfect way in real-time, it turns out that a refined version of the MHI shown in Figure 6.2 is the best solution, which provides great accuracy with fast computation speed. Different from the Opisthenar [77], our tweaked version use the parameter $\alpha = 0.2$ which means it is observing the weighted sum of 5 past frames. Another problem that might occur is that the network might focus on the hand contour movement instead of the skin deformations, which will harm the network's generalizability. Therefore, an erosion

operation is added to the hand masks to filter the outer-edge, and the intensity inside the hand is increased to let the network focus on inner motions on the back of the hand.

6.2.2 3D Hand Pose Accuracy

Procedure

The evaluation of hand pose estimation consists of three separate studies. We first trained our network on an individual user's data to evaluate the personalized model. This aims to study the precision of each specific joint and finger, which could be helpful for future improvement. For comparison, since our work is the first real-time hand pose estimation system using egocentric wrist-worn camera, some similar state-of-the-art networks dealing with direct/indirect pose estimation were used as baselines. Nevertheless, we also carried out a lighting condition study to study the robustness of our network and an ablation study of different network architectures and different inputs on the basis of the proposed method.

Baselines

As mentioned in the *Procedure* section, since there is no identical work for comparison, we used some typical standards or similar networks as baselines. The Direct (ResNet18) is the condition that directly regresses raw camera images to the 3D representations frame-by-frame, which can be considered as a base condition. Also, we included the Nearest Neighbour Search [15], also known as k-nearest neighbour (k=1), because it is a typical standard for pose estimation. Since the CNN-LSTM architecture we used is similar to the work of Yuan et al. [79], we also include them as baselines, even though they used bi-directional LSTM which means their networks are offline. Another baseline is the work by Yeo et al. [77] which we followed-up. Although their system is not designed for full hand pose regression, we changed the output of their network and fine-tuned with our dataset. Instead

Joint\Finger	Index (1)		Middle (2)		Ring (3)		Pinky(4)		Joint Avg.	Thumb (0)	
	MAE	SD	MAE	SD	MAE	SD	MAE	SD	MAE	MAE	SD
MCPv	7.05°	±0.40	6.32°	±0.54	6.3°	±0.39	6.92°	±1.21	**6.65°**		
MCPh	7.94°	±0.75	7.87°	±0.62	7.17°	±0.64	9.78°	±1.99	**8.14°**		
DIP	6.92°	±0.59	6.78°	±0.76	6.70°	±0.70	9.80°	±1.73	**7.55°**	12.69°	±2.26
PIP	8.47°	±0.92	7.85°	±0.98	7.66°	±0.87	11.11°	±1.33	**8.77°**		
Finger Avg.	**7.60°**		**7.20°**		**6.96°**		**9.40°**				

Table 6.1: Average result of the individual model of each joint/finger (metrics: MAE(SD) unit: degree).

of using a Leap Motion camera, we used a monochrome masked hand as the input. Zhou et al. [82] is the state-of-the-art real-time 3D hand capture methods using a single monocular camera. Their network used a location map to extract positional features and regress the 3D joint location of the hand. They also used an IK-Net for learning inverse kinematics to recover the joint location to joint angle and match the output with the MANO hand model. To compare with this work, we fine-tuned their network by changing the input to our raw egocentrics dorsal hand images.

Together with the 5 baselines above, our method with/without Kalman filter are analyzed. All results are using joint angle-based representations while the baseline of Zhou et al. [82] also outputs the full thumb joint rotation since they use the MANO hand model. Therefore, we re-calculate the thumb vector from their output which might cause inaccuracy. However, we believe the overall performances are still comparable.

Finger and Joint Error Study

We first trained our network on individual subjects to study the precision of each finger and joint. Five personalized models were trained and evaluated on each specific user's data, 20% (9000 frames) of the user's data was randomly kept for this test. Table 6.1 shows the average result of 5 individual models, all results are recovered to angle unit for a better visibility, where the unit is degree. The first 4 rows show the mean absolute error (MAE) and its standard deviation (SD) of each joint of the 4 fingers, respectively, together with an average result of each joint. The columns stand for each finger joint rotation and the last row is vector angle error of the thumb.

Comparison Study

Next, to show the effect of our network compared with baseline conditions. In this study, we trained both the 5 individual models and a general model for each network. Here, we used a session-split of leaving one specific session (9000 frames for

Method	Individual	General	Leave-1-user
Nearest N.[15]	18.44°	21.78°	20.89°
Direct(ResNet18)	18.39°	22.09°	29.11°
Yuan et al. [79]	12.48°	14.40 °	14.53°
Yeo et al. [77]	16.67°	18.52°	20.24°
Zhou et al. [82]	15.95°	20.06°	21.06°
Ours (w/o KF)	9.28°	10.33°	10.71°
Ours (w/ KF)	**8.81°**	**9.77°**	**9.72°**

Table 6.2: Comparison with baseline methods, Our methods are divided into with/without Kalman filter (KF).

one subject, 45000 frames for general model) for the Individual and General model to study the cross-session generalization of our system. As well as a user-split of leaving one user out, to perform a cross-user validation.

6.2.3 Ablation Study

Starting from the very basic two CNN networks (VGG16 and ResNet18), we analyze the effect of the network by gradually adding other model parts. This ablation study is mainly comparing how different input and different temporal feature extraction will affect the precision of the hand tracking, and the same data were used as the comparison study. Three different types of input together with a with/without data augmentation condition were compared, while the network is changed by with/without LSTM or Kalman filters. In total, seven conditions are compared as shown in Table 6.3, the inference time (ms) using the laptop is also recorded for comparison. To notice, the *ResNet18 (RGB)* method here is different from the *Direct (ResNet)* in the former study for it uses the masked hand images preprocessed by our system instead of raw images.

6.2.4 Lighting Condition Study

Our study is mostly done in an indoor with fluorescent lamp lighting condition. To show the performance of our network in different lighting, we also conduct a comparison of angle MAE in different conditions shown in Figure 6.7. We asked one of the participants to take data under 4 other lighting conditions besides the base condition (In-Light), which are:

- Outdoor Day: Natural day light on a cloudy day outside.
- Outdoor Sun: Strong sunlight on a fine-weather sunny day.
- Indoor Dark: The lamp is turned off with only stray light from a PC monitor.
- Outdoor Night: Only light from street lamp at night.

In all condition, we take the same quantity of data as the former studies from the participant, which results in 45000 frames for each lighting. However, in the Out-Sun condition, we cannot use Leap Motion to capture the ground truth due to high

Architecture (Input)	Angle Error		Inference
	Individ.	General	Time (ms)
VGG16 (RGB)	16.07	18.19	54
ResN18 (RGB)	16.11	18.70	17
ResN18+LSTM (RGB)	11.95	14.01	35
ResN18+LSTM (Motion)	9.29	10.69	33
ResN18+LSTM (TS)	9.28	10.13	40
Ours (w/o Data Aug.)	9.35	11.11	40
Ours (TS)	**8.81**	**9.77**	41

Table 6.3: Results of ablation study of different network architecture and input data. The metrics of Angle Error is MAE (degree), TS stands for two-stream input, 'Ours' stands for ResN18+LSTM+KF (TS).

Figure 6.7: Images of camera under different lighting conditions.

intensity infared light so only the other 4 conditions are evaluated. Qualitative performance of Out-Sun is shown in our video.

6.2.5 Results

Finger and Joint Error Study: The result (Table 6.1) shows that the index, middle, and ring fingers achieve higher precision (MAE around 7) since the deformations occur in the middle of image, while the pinky finger performs worst (MAE=9.40). For the joints, it is a bit surprising that the MCPhs also do not perform well (MAE=8.14), worse than the DIPs (MAE=7.55), while the PIPs are the worst (MAE=8.77).

Comparison Study: When compared with other baseline methods (Table 6.2),

	Base(In-Light)	Out-Day	In-Dark	Out-Night
MAE	7.93°	7.77°	8.46°	8.21°

Table 6.4: Comparing the accuracy of our method in different lighting condition (Out-Sun removed due to lack of ground truth).

the proposed method with KF outperforms the baseline with a large advantage. (MAE: Individual =8.81), General=9.77, Leave-1-user=9.72). Even the proposed method w/o KF leads the baseline with an average of approximately 4-degree error. In the baseline methods, the work from Yuan et al. performs the best (MAE: Individual=12.48, General=14.40, Leave-1-user=14.53). Also, different from other methods, the proposed method does not show a great difference between the general, leavel-1-user, and individual model, which could be a proof of the generality of our network.

Ablation Study: Observing the results (Table 6.3), in the first and second row, the VGG16 and the ResNet18 show similar results, yet the ResNet18 is much faster in inference time. Comparing different inputs of row 3-5, the motion input (MAE: Individual=9.29, General=10.69) obtains a much higher accuracy with less inference time than the RGB input (MAE: Individual=11.95, General=14.01), while two-stream input obtains a higher accuracy in the general model (Motion: General=10.69; TS: General MAE=10.33) with a slightly greater inference time. For the network architecture, comparing row 2 with row 3, we can notice there is a 4-degree difference with/without LSTM. Also, comparing row 5 and 7, it is evident that the LSTM-based KF outperforms normal LSTM with the highest accuracy. Overall, it is clear that with two streams of input and more complex networks, the accuracy becomes higher. Lastly, the difference from row 6 and row 7 indicates that, using data augmentation will greatly increase the general accuracy (from 11.11 to 9.77, 12% increase).

Lighting Condition Study: From Table 6.4, we can tell that the performance becomes worse when the lighting gets darker. However, the difference is relatively small between the best (Out-Day, MAE=7.77) and the worst (In-Dark, MAE=8.46).

6.3 Experiment on Feet Pose in Skiing

Similar to the BackHandPose, we also apply the InvisiblePoseNet to full body posture by using pressure map of the feet. In some motor skills such as skiing, it is difficult to place sensors on the body where the only place that can be captured might be the feet. In addition, in most specific motor skills, our posture is closely related with our feet motion, so feet pressure is a good indirect clue for regressing whole body posture.

The only difference from the back hand pose is the indirect feature extraction module. Since the feet pressure already represents the intensity of strength (change in speed), the conventional MHI is used instead of the OF-MHI for a lighter computation. The network architecture can be seen in Figure 6.8.

6.3.1 Data Collection

The data is collected by motion capture system and feet pressure sensor, as shown in Figure 6.9, performed on a motor-based ski simulator. This SkyTech Pro simulator is sufficiently realistic while the national ski team of the US and Canada also employ it in the training. The user's feet has a 5 degree of freedom (DOF) while the distance between the skis are fixed.

For the motion capture, we employ a reflective marker-based Optical motion capture OptiTrack, with 8 IR camera and an RGB camera for reference. The feet pressure is taken by two Moticon OpenGO insole pressure sensors which can be synchronized to the mocap system, under a 1000Hz refresh rate.

In total, data from four subjects are taken (4 males, 1 females, two is professional in skiing and three is intermediate) to serve as the training and testing data. The third-person-view RGB reference images are also collected for the vision-based baseline network for comparison. As a result, about 20K frames of data in a 240-fps setup is recorded.

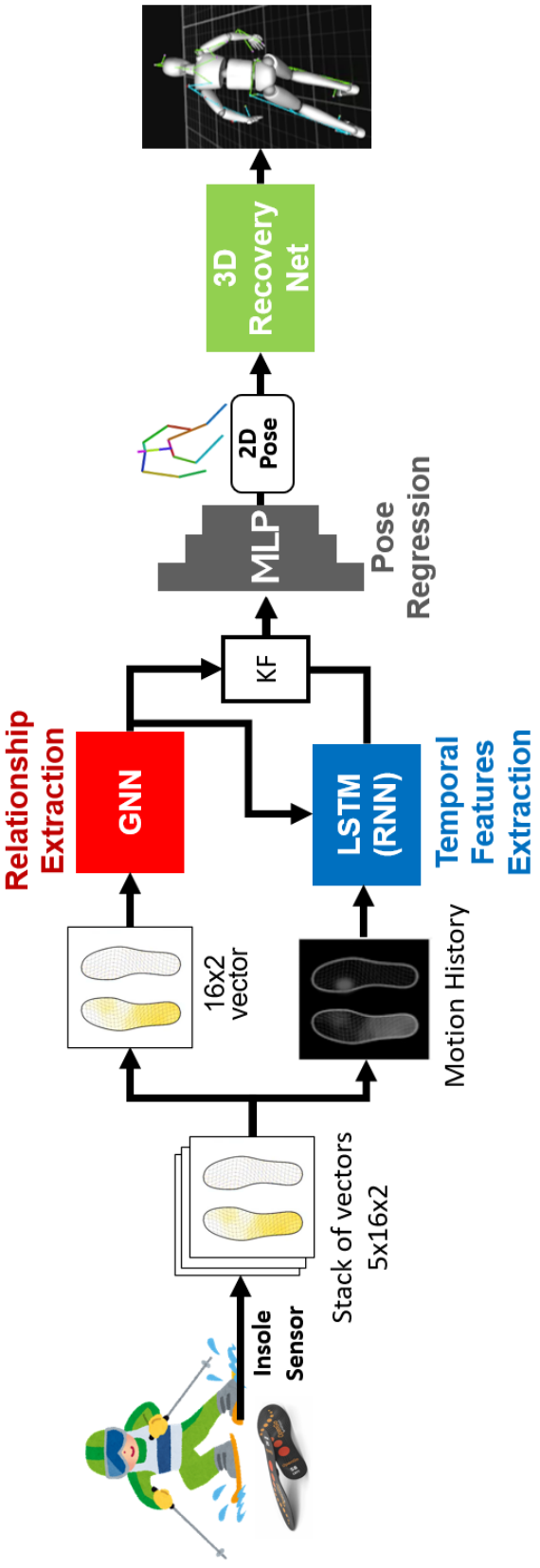


Figure 6.8: Network Architecture for SkiFeetPose.

Figure 6.9: Skiing FeetPose data collection.

6.3.2 3D Body Pose Accuracy

Baseline

Since few previous work focus on the same topic, the baseline we used here is a direct CNN regression, which simply regress the 3D posture from the feet presure map. The CNN used here is the ResNet18, which is the same one used for feature extraction in the proposed InvisiblePoseNet.

Also, to study whether the proposed model is practically useful, we also trained the state-of-the-art vision-based method by Mehta et al. [46] as another baseline for comparison. To note that, the model here is finetuned from the pre-trained model using our reference RGB data, while

Procedure

A cross-validation is performed to study the precision and the robustness of the feetpose network. For the two baselines and the proposed method, we divide the training and testing data in three different ways:

- Individual Model: All the training data and the testing data are from the

Methods	Skiing Pose MPJPE ↓ (mm)		
	Individ.	General	Leave-1-user
Direct CNN Regression	91.8	90.7	127.3
Mehta et al. (Vision Based) [46]	55.4	57.4	63.4
Ours (FeetPose)	55.4	61.8	74.5

Table 6.5: Precision of feet pose estimation in skiing (MPJPE).

Methods	Skiing Pose 3D PCK ↑ (%)		
	Individ.	General	Leave-1-user
Direct CNN Regression	51.0	50.4	39.8
Mehta et al. (Vision Based) [46]	85.2	84.0	78.4
Ours (FeetPose)	85.4	80.1	67.6

Table 6.6: Precision of feet pose estimation in skiing (3D PCK).

same person. 80% on the data are used for training and 20% are used for testing. For each subject, an individual model is trained and tested, while the final result of all models are averaged.

- General Model: All the training data are mixed together, while 80% of all data are randomly picked out for training and the remaining are used for testing. Only one general model is trained using all data.

- Leave-1-user Model: From the 5 subjects, the data of 1 subject is left for testing while the others are used for training. This is the cross-validation test for checking the robustness of the model.

6.3.3 Results

The results (Table 6.5 and Table 6.6) shows the Mean Per Joint Position Error (MPJPE) and the 3D Percentage of Correct Keypoints (PCK) of the testing result. From both table, we can observe that the proposed indirect feature-based network greatly outperforms the direct CNN regression. Even compared with the state-of-the-art vision-based real-time 3D pose estimation XNect by Mehta et al. [46], the proposed method does not fall much behind in the general model (MPJPE: Ours 61.8mm v.s. XNect 57.4mm, PCK: Ours 80.1% v.s. XNect 84.0%) and even outperform the vision-based method in the individual model (PCK: 55.4mm v.s. XNect 55.6mm, PCK: Ours 85.4% v.s. XNect 85.2%.

However, when it comes to the leave-1-user out result, the proposed method performs worse with an approx. 11mm mean error (MPJPE: Ours 74.5mm v.s. XNect 63.4mm) compared to the vision-based method. This result suggests that the proposed indirect-feature based method is weak in generalization, which might be the reason of the not normalized feet pressure data.

Chapter 7

Application on Skill Acquisition

In this chapter, we will introduce three training application for three different types of skills using the proposed indirect feature-based network.

The first is the ski training system using the spatial indirect feet pose motion tracking for real-time feedback.

Next is a VR table tennis training system which is based on the previous temporal ball trajectory prediction system. The system uses VR to visualize 3D future trajectory to provide intuitive visualizations.

Lastly, is the Piano training, which make use of the spatial BackHandPose and the temporal indirect keystroke features to estimate precise hand poses for piano playing analysis.

7.1 Alpine Skiing

7.1.1 Implementation

Our training system consists of an indoor ski simulator, a VR system (HTC Vive Pro[1]), which includes a head mounted display, two base stations, a pair of trackers for tracking the skis, and the proposed motion tracking system. Since real skiing also requires a helmet and goggles, which narrow the field of view, the use of a head mounted display does not greatly disturb the skiing experience.

For the training in VR, we created a virtual ski slope environments in Unity 3D.

[1]https://www.vive.com/eu/product/vive-pro/

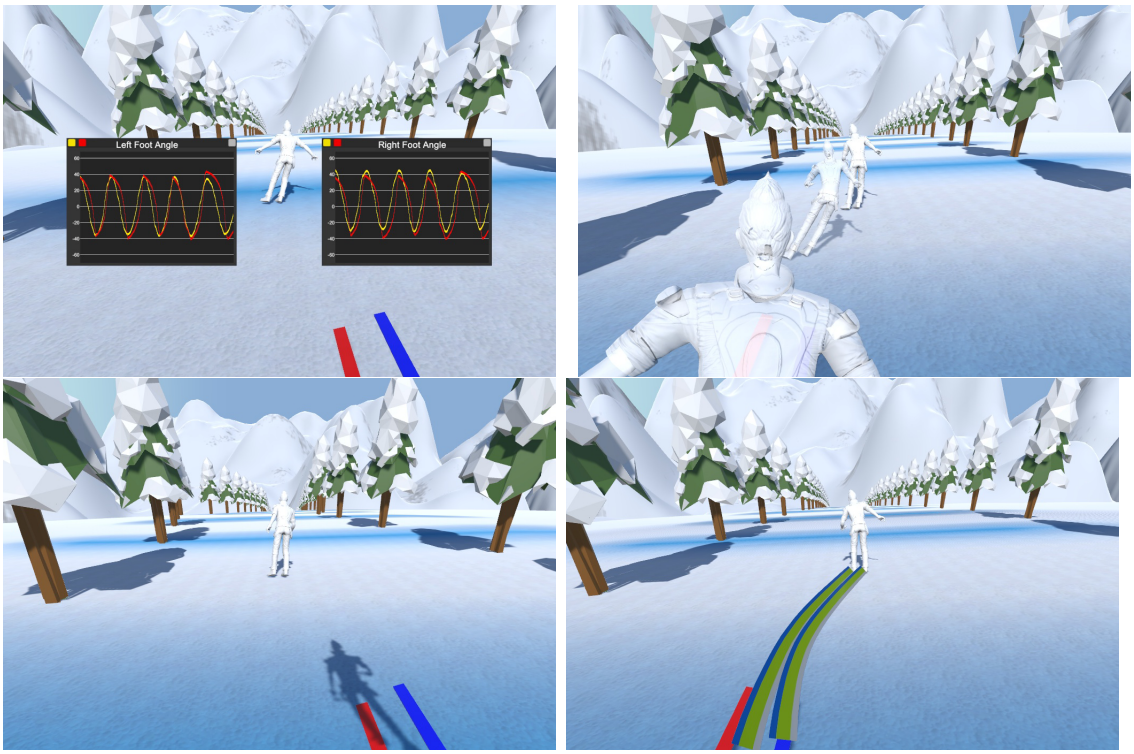


Figure 7.1: Four Visualizations: *Graph Feedback* (top left), *Pose Breakdown* (top right), *Ground Shadow* (bottom left), and *Color Trail* (bottom right)

As its main purpose is to serve as a test environment for different visualization, we designed a plain, smooth, down hill ski slope with a steadily increasing grade, and the course is designed as the slalom skiing.

Next, for the training, to study the use of feet pose motion tracking, we develop different feedback based on the difference between a coach's motion and user's motion.

Graph Feedback

To provide the users with direct feedback on the difference between their own and the expert's movement, we visualize two graphs on the HMD to show the average body angle of the user and the expert. The data is captured from the VR tracker's rotation, and plotted as 2D coordinates in a graph of which the horizontal axis

represents the time sequence. The two graphs in Figure 7.1 (top left) are showing the right and left foot leg angle of the user (the red line) and the expert (the yellow line), respectively. Since the angles are position-related, the user can know how the expert move his/her feet in the same position and notice the difference. In our pilot tests we noticed that professional skiers continuously output periodic curves while a beginner's graph is less periodic.

Pose Breakdown

To better visualize both the temporal and spatial information of the expert's motion, we implemented a visualization that shows the sequential poses of the professional along the trajectory. This is done by rendering static copies of the expert avatar in even intervals so that the users can match the motion and position. This function is designed to support users with following the expert's trajectory correctly, which can be difficult without any visual cues. When a successful "mimic" is performed by the user, the coach model shall perfectly collide with user's which is very intuitive to observe.

Ground Shadow

Another idea is to place the shadows of the coach's and user's avatars rendered in the *Pose Breakdown* on the ground. From this initial idea we finally use a single shadow that continuously shows the posture of the user (see Figure 7.1, bottom right). Using shadows for learning movements from experts has already been explored successfully in other sports, such as golf [29] and might also be beneficial in skiing. Also, observing the posture of oneself from the shadow is the most natural behavior of human being, since it is not possible to bring a mirror to the ski slope.

Color Trail

The graph feedback might be quite overwhelming from our pilot study. Hence, we searched for simpler ways to provide feedback. Our observation was that it is hard to adapt to a single value that is constantly changing and that the feedback should rather help to quickly judge the current performance. Thus, we looked at ways to summarize the user's performance so that it can be perceived in one glimpse. This led to the use of color as a performance indicator (green = good, red = bad). After experimenting with various individual UI elements we decided to paint the feedback into the texture of a trail. The trails, which consist in pairs, one for each foot, do not only show lateral movement but also rotation by being rendered as a ribbon to indicate the ankle rotation of the expert (see Figure 7.1, bottom right) .

7.1.2 User Study

Participants

We recruited 12 participants (4 females) with an average age of 23.5 (SD = 3.2) from a computer science department students at a local university. Five of them had hardly any skiing experience, five have skied before but do not do it regularly, while two can be considered more experienced. Participants are divided into two groups to experience vive tracker-based motion capture and the proposed insole-based motion capture.

Procedure

After an initial briefing in which we introduced the different conditions and the goal of the study, the participants were asked to put on ski boots and step on the simulator. They could familiarize themselves with the movement on the simulator before they put on the HMD. We then put on the corresponding motion capture and started the simulation which presented the different conditions to them. Besides the four visualization introduced before, a baseline condition is added for comparison,

where no feedback but only the coach is running in front of the user. The order was randomized using Latin square. Each condition consisted of 1 training trial to get familiar with the visualization and 2 test trials. Each trial started with an 8 s countdown, to pick up the movement pattern and was then followed by a 30 s trial period.

After performing all 4 conditions, the participants were asked to qualitatively rate their experience on the accuracy of the motion capture and the comfortability of using the system, also a semi-structured interview was conducted. The entire process took approximately 45 min. per participant (10 min. briefing, 20 min.

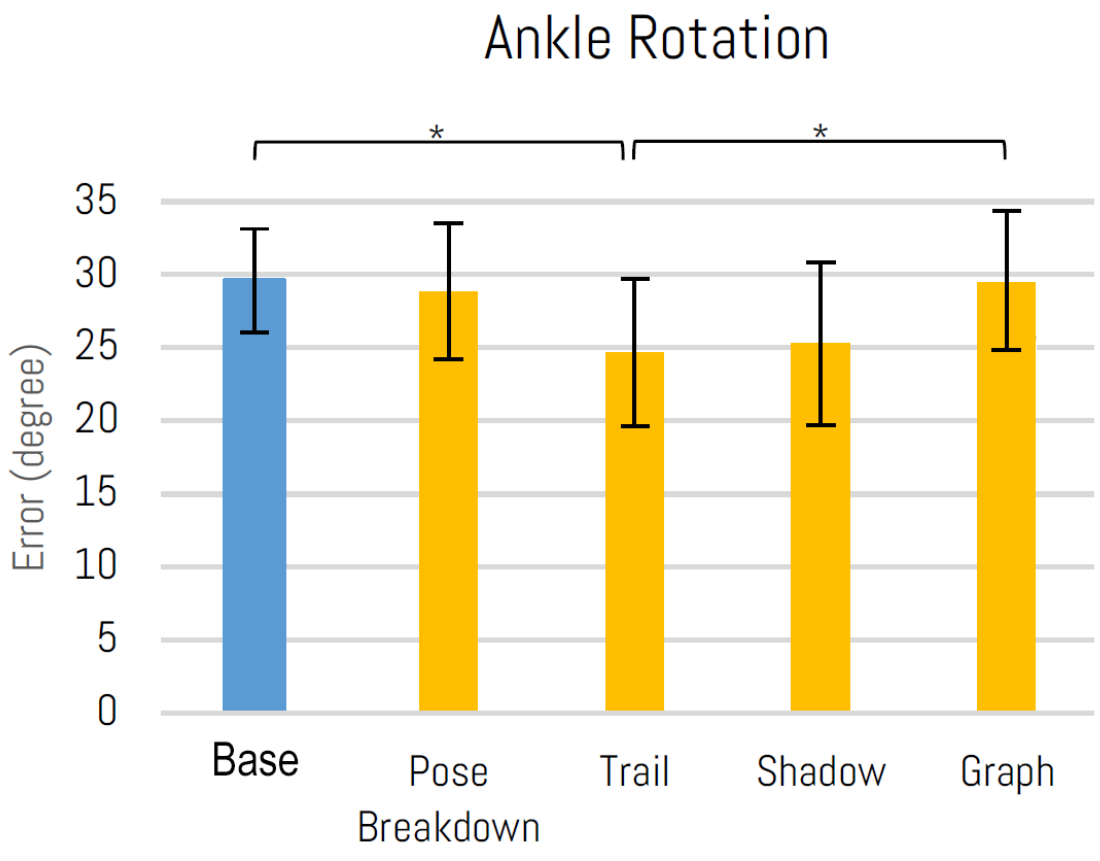


Figure 7.2: Quantitative results for the Ankle Rotation. The colors categorize the conditions into baseline (blue) and the proposed visualizations (yellow). The brackets on the top indicate the significance between the conditions: * ($p < 0.05$)

study, 15 min. interview).

7.1.3 Results

For the quantitative experiment result (as shown in Figure 7.2), we obtained the ankle angle difference between the user and the coach and conducted a repeated-measures ANOVA ($\alpha = .05$) on the ankle rotation. A significant difference between the conditions could be detected ($F_{6,162} = 15.837$, $p < 0.001$).Tukey's range tests as post-hoc unveiled several significant differences between conditions.

The performances in the *Ground Shadow* ($M = 25.25$, $SD = 5.58$) and *Color Trail* condition ($M = 24.68$, $SD = 5.05$) were considerably better, with the *Trail* leading to a significantly better result than the *Baseline* and the *Graph* condition ($t_{21} = -3.355$, $p < 0.05$) regarding ankle rotation.

For the preference of the motion capture, the results was more controversial. In terms of the accuracy of motion capture, no significant difference can be detected between the vive tracker-based and insole-based condition, which indicates a close performance on the proposed system. For the comfortability of the motion capture, the results show a clear difference between the tracker-based method and the proposed method ($Z = -2.153$, $p < .05$). Overall, the insole condition was perceived quite positively as it was *"natural to wear on"* (P4), and *"not disturbing at all"* (P13). This results suggest that when applied to alpine skiing, the users cares more about usability and comfortability of the system instead of accuracy, especially for these beginner participants.

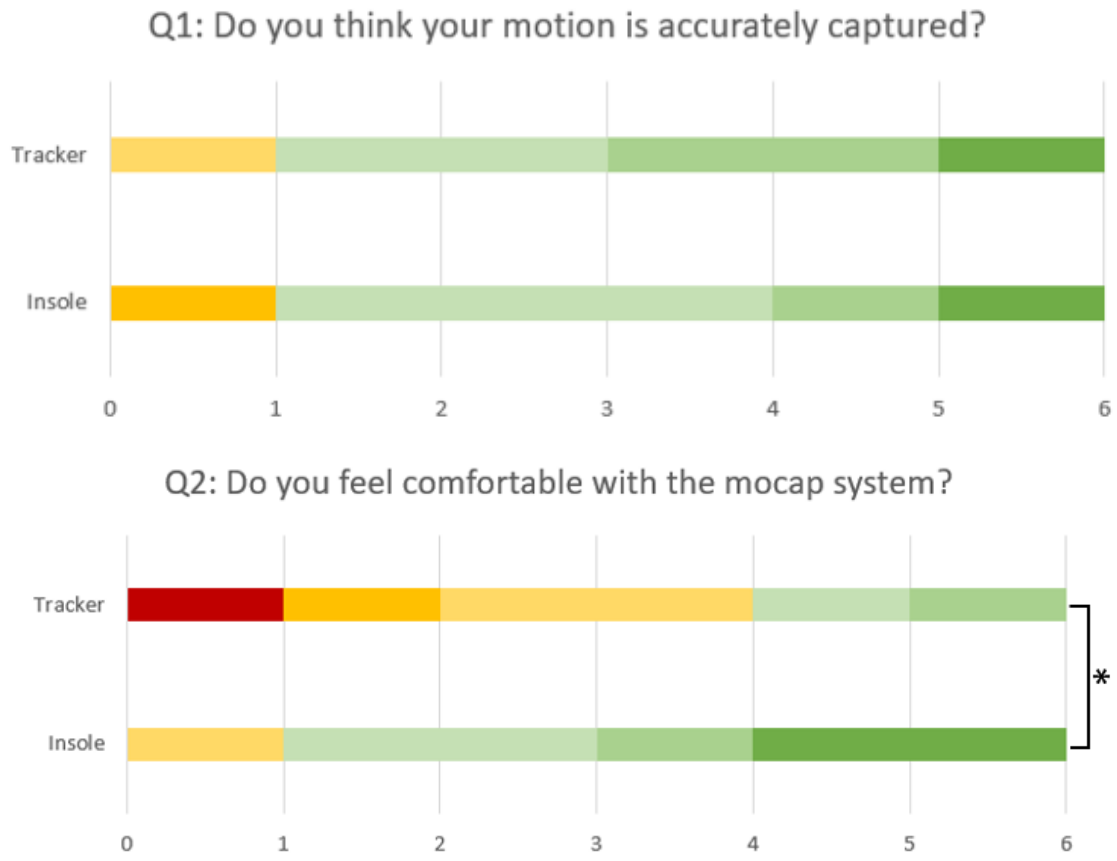


Figure 7.3: Qualitative results of users' preference in 6-point Likert scale.

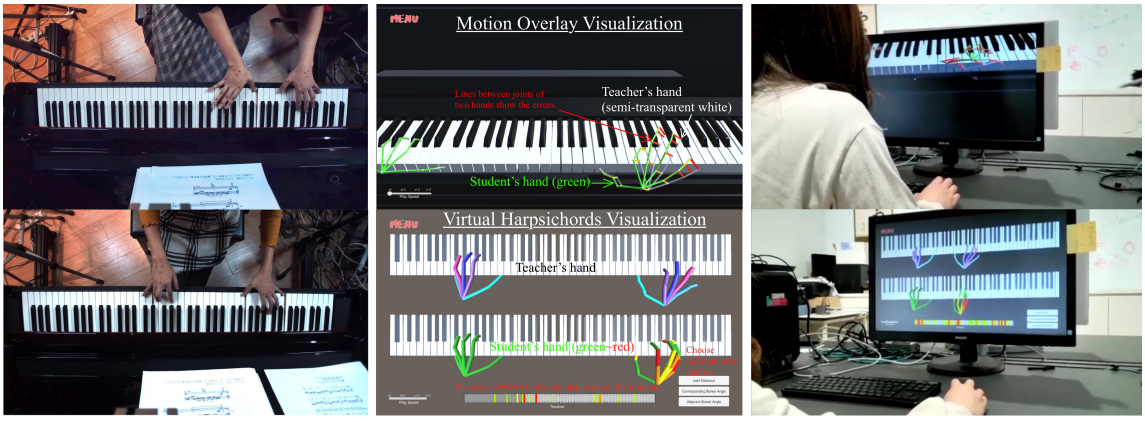
7.2 Piano

7.2.1 Implementation

In piano, it is difficult to collect precise hand pose data because it is too bulky to have gloves or markers being worn by pianist's hands. Our spatial indirect feature-based wrist-worn back hand pose system is suitable for a natural motion capture. However, piano estimation requires degree-perfect high precision analysing and accurate timing, which is difficult to realize using the current system. Therefore, we combined the spatial and temporal indirect features and developed a PiaSim to realize precise and natural hand pose estimation.

PiaSim Spatio-Temporal Network

In a piano performance, the timing of sound (when a key is pressed) is considered to be the most essential factor. Accordingly, the fingertip position (also the PIP rotation) needs to be accurate. To enhance the training to be more specific towards piano hand motions, a PiaSim network is developed to output keystroke based on an input hand sequence. The network consists of a Long-short Term Memory (LSTM) [27] layer to extract time series motions and a fully-connected layer to



(a) Data Collection (b) Our Visualization Methods (c) Pilot Study

Figure 7.4: PiaSim Spatio-Temporal Network

reproduce the keystroke. The input stacks the last 5 frames of 3D hand postures, which are the 3D positions relative to the piano (same as the taken data), resulting in an input size of $5 \times 21 \times 3$. The output is an 1D array with a size $n = 12$ (for the 12 keys in one scale) showing the key-depth $\mathbf{k}_i$ of the i-th key (ranged from 0-10 mm, normalized to 0-1). Due to the fact that not all the keys are always pressed, using an L2 loss is not suitable for such a sparse vector. Assume that the ground truth is $\mathbf{k}^*$, and the number of value greater than zero in $\mathbf{k}^*$ is $\mathbf{N}_{(k^*>0)}$, the loss function for the keystroke results are as follows:

$$\mathcal{L}_{\mathbf{key}} = \sum_{i=1}^{n} \|\mathbf{k}_i - \mathbf{k}_i^*\|_2 / \mathbf{N}_{(k^*>0)} \tag{7.1}$$

The output size is set to 12, so an octave (a note and the same higher note is played, for example C4 and C5) is considered to be pressing the same key in the prediction. For training, given that the keystroke information is not obtained for the EG group, we developed a keystroke simulator to simulate keystroke information from either MIDI or ground truth hand poses.

Finally, the overall loss function for the training procedure is as follows, where λ_1, $\lambda_2$and λ_3 are the weights for the joint position loss, heat map loss, and keystroke loss, respectively:

$$\mathcal{L} = \lambda_1 \|\mathbf{P} - \mathbf{P}^*\|_2 + \lambda_2 \|\mathbf{H} - \mathbf{H}^*\|_2 + \lambda_3 \mathcal{L}_{\mathbf{key}} \tag{7.2}$$

Training Conditions

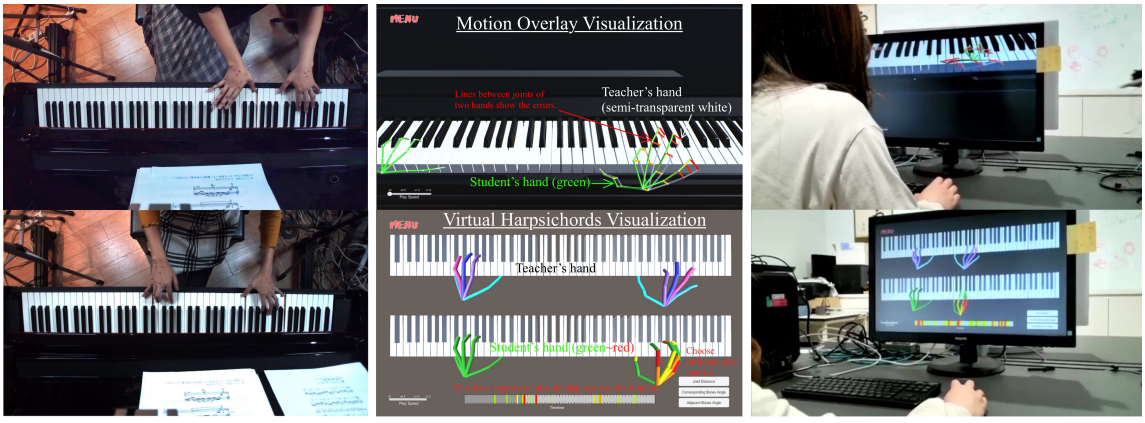
As a prototype, we collect test data from experienced pianists using marker-based motion capture. To solve the problem of unmatched playing speeds, we use the previously mentioned TCC network [17] and a dynamic time warping algorithm to synchronize the data. The main idea is to visualize the hand differences between two plays (student and teacher, they can also be different plays from the same person), after some interviews with pianists, we build two different approaches. Both methods have some common features such as a scroll-bar to adjust play speeds, and a controllable camera to observe the hand movements from multiple perspectives.

Motion Overlay: The first visualization is the motion overlay, which is a straightforward approach to simply displaying both the teacher's and the aligned student's hands on the same piano keyboard.

In this condition (middle left in Fig.1), the student's and teacher's hand skeletons are in different colors for a better visualization, where the teacher's hands are semi-transparent. To provide intuitive feedback on the error between a student and a teacher, the distance between the corresponding joints is connected by a line. The colors of the distance lines, as well as each bone, are changed from green to red based on the magnitude of the error, as shown in Fig.1.

Virtual Harpsichord: On the other hand, some pianists suggest that instead of a "noisy" overlaid visualization, they demand a side-by-side option to compare each play individually. Therefore, we implement a second visualization which displays the four hands on two separate keyboards, inspired by a harpsichord (piano with up and down keyboards, as shown in the middle right in Fig.1). It allows the user to better see where the differences between the hands occur.

Besides the location of the error, we also hope the user can realize the timing of the error more intuitively. Therefore, an interactive timeline is added to visualize which segment of the entire clip the differences are happening. Based on error thresholds, the fault part will gradually turn from green to red. Also, to avoid



(a) Data Collection (b) Our Visualization Methods (c) Pilot Study

Figure 7.5: Overview of Piano Training System.

too informative visualization, the errors are divided into three types: joint distance error, corresponding bone angle, and adjacent bone angle, which a user can choose from the bottom right checkbox (shown in Fig.1).

7.2.2 User Study

The motion overlay is a straightforward and intuitive visualization, while the virtual harpsichord provides side-by-side comparison and clear error feedback. We run a pilot study to compare them with two baselines. The first baseline (B1) is the most conventional way which simply plays the two original videos of the student and teacher, and since the videos are not synchronized, users cannot compare them simultaneously. Another baseline (B2) is the synchronized videos where the student's play speed is aligned with the teacher's, and users can have side-by-side comparison. Our proposed methods are labeled as V1 (Motion Overlay) and V2 (Virtual Harpsichord).

7.2.3 Results

Seven experienced pianists (6 female, 1 male, with experience ranging from 15 to 39 years.) are invited as participants. They are told to try each condition for 5 min and provide their overall preference for that condition in a 7-point Likert scale. Figure 7.6 shows the result of mean scores for each condition. An ANOVA statistical test suggests significant differences in the results so we conduct a Tukey's HSD post-hoc test. The result suggests that both the motion overlay and the virtual harpsichord are significantly better than the baseline of the two original videos. (V1-B1: $p=0.004$, V2-B1: $p=0.001$). When compared to the condition of two aligned videos, the virtual harpsichord are significantly better. (V2-B2: $p=0.033$).

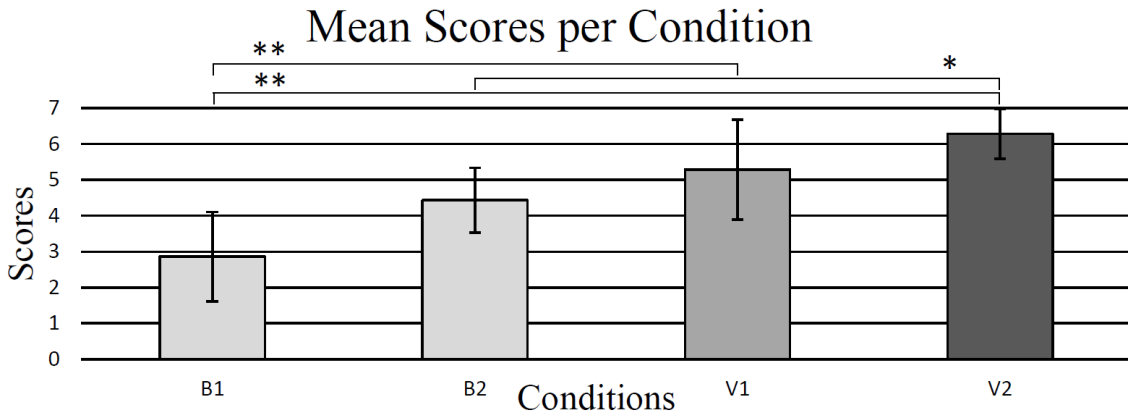


Figure 7.6: Quantitative results of the questionnaire. *($p < .05$), **($p < .01$).

7.3 Table Tennis

7.3.1 Implementation

A spin serve in table tennis is very difficult to return. In this application, we focus more on how visualizing future visual cues could affect the understanding of the spin for beginners.

A baseline condition is firstly developed with a real person serving a strong spin ball in virtual reality, as shown in Figure 7.7, a guidance condition using the predicted ball trajectory is build on top of the baseline condition. We also include another two common visualizations for comparison.

Future Guidance (Cond. G)

This is the condition which make use of our prediction system. A translucent ball trajectory (obtained from the prediction system) is visualized in this condition as shown in the 4th figure in Fig.7.7. A racket is shown in this condition to guide the user to a correct way to return the ball in advance, while the virtual instruction is started from the front of the user's body and completes a return based on different serves. This correct form is taken from an experienced player by recording their

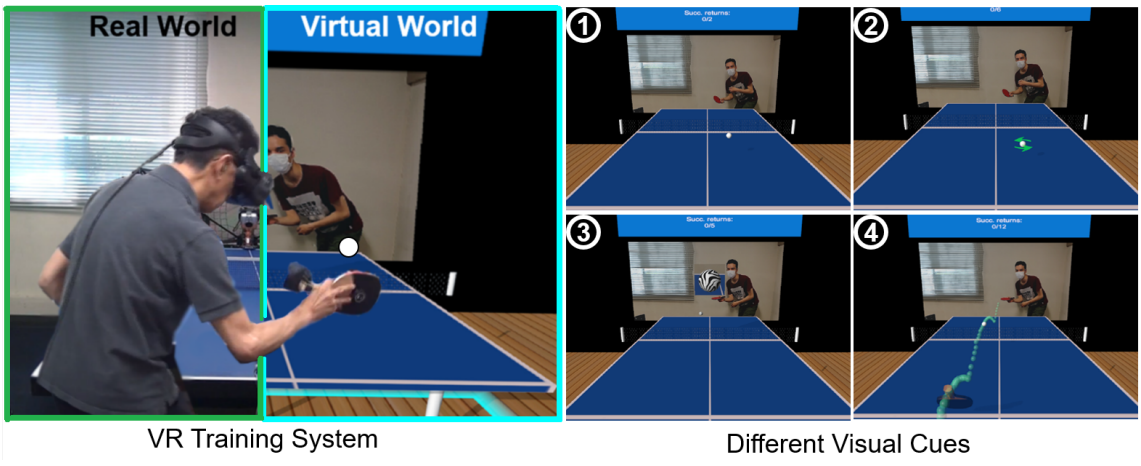


Figure 7.7: The 3 VR conditions along with the base condition.

racket position. This condition is designed to show a direct way of teaching with less freedom for the user.

Spin Arrow (Cond. S)

Next, the Spin Arrow condition is a simple way of visualizing the spinning information for the user. As shown in the second figure of Fig. 7.7 two green arrows are constantly rotating around the served ball to show the spin direction of the shot. This condition is developed to satisfy the most basic requirement of viewing the spin direction, but is also less informative which does not not disturb the user.

Bullet Time (Cond. B)

Next, inspired by some action movies and games, where there is super slow motion for the character to avoid bullets (called "Bullet Time"), we build a function with the same name that extremely slows down the time to 0.05x after the first bounce on the opponent's table. It lasts for 0.25s to match the total time of the other conditions and during this time the user can see a zoomed window above the ball showing a zebra texture on the ball (as shown in the 3rd figure of Fig. 7.7) for a better visualization of spin. This condition is designed to show another possibility of temporal distortion, where users have more time to observe the spin and trajectory of the ball.

7.3.2 User Study

We want to study the detailed effect of prediction functions compared with other visualization and prove the usability of the training system. Therefore, we introduce new performance metrics to evaluate the performance of the user. Also, a questionnaire in 6-point Likert scale is performed instead of an oral interview to quantify the user's feeling. Through the two types of detailed experiments, we want to answer the following research questions:

- RQ1: Which is the best virtual condition that improves the skill most?
- RQ2: Which condition can motivate the user the most?
- RQ3: Which condition can help the user to understand spin?
- RQ4: Is the new haptic device noticeable by the user?
- RQ5: Is the VR training effective in the long-term?

Participants

For this experiment, we invite 12 healthy participants (2 female) with an average age of 27.5 (SD = 10.3) to perform a within-subjects study. All the participants are still right-handed with less table tennis experience. The participants in this study were gathered from a computer science department of a university, and all the participants were paid for the one-hour study. This study as well as the previous initial pilot study are approved by the local ethics committee.

Performance Metrics

Another issue we found in the initial study were the performance metrics. A simple success rate is too discrete to evaluate the user's skill, because a close return that flies barely pass the table's edge should be more valuable than a return flying 1 meter away from the table. In this experiment, along with the previous success rate (S.R.), we also introduce another 3 metrics to enhance the evaluation:

- Table Distance (T.D): Normally, a much more precise distance of a specified target should be used in high-level table tennis training. Since the participants are all beginners, here we used a simplified table distance metric for evaluation. Because the ball trajectory both in the real world and virtual world can be precisely tracked, it is possible to calculate the closest distance when a returned ball reaches the table level (as shown in Fig 7.8). The distance of a successful return is 0 while the maximum distance here is set to 1

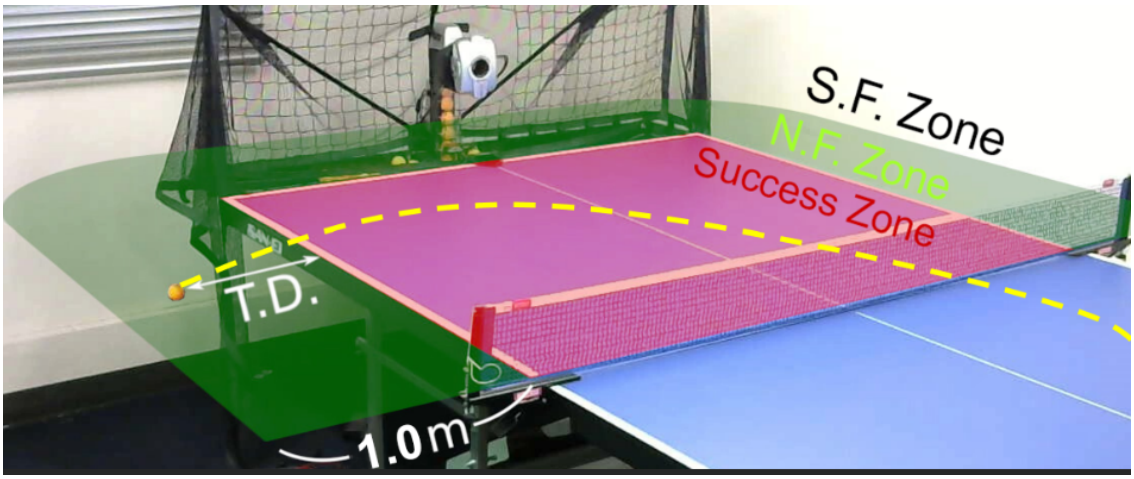


Figure 7.8: The 4 Performance Metrics for the study.

meter because of our room size. A return where the distance is greater than 1m or never reaches the table level (i.e. a ball is returned below the table and never reaches the table height) will be counted as "super failure" (which is mentioned later) and the distance is counted as 1. Finally, the values of T.D. are averaged in each condition, which result in the Mean T.D. (0∼1 m).

- Super Failure Rate (S.F.R): As mentioned above, most of the returns by the user fly far away from the table, we count these returns where the distance is greater than 1 m as "super failure" and this allows them to be differentiated from some close failures. Here, a return below the table as well as a missed return is also counted as a super failure.

- Normal Failure Rate (N.F.R): The rest of the failed returns are counted as a normal failure, which includes those returns within 1-meter off the table when reaching the table height, and those directly hit on the net. This is treated as a quite "close" return which can reflect the user's performance to some extent. Of course, the sum of S.R., S.F.R, and N.F.R. should be 100%.

Procedure

An overview of the experiment procedure is summarized as follows:

1. Five-minutes free training versus the robot with no-spin serve for understanding table tennis.

2. Cond. R: Train with the real robot system for 60 shots of the 3 types of spin ball (20x each, random order).

3. Cond. V: Train with the base VR system for 60 shots of the 3 types of spin ball (20x each, random order).

4. Shuffled order of the following 3 conditions, same shots as above:
   Cond. S: The Spin Arrow
   Cond. B: The Bullet Time
   Cond. G: The Future Guidance

5. A questionnaire after each condition and an interview at the end.

6. Cond. RL (3 weeks later): A follow-up real world training which was performed 3 weeks after the above training.

This time, before we started the study, a simple introduction was given to the participants to let them acquire basic knowledge of table tennis such as the rules and the types of spin. Also, an additional free training time was provided to the participants before starting the experiment. Since our main target is to study the effect of different visual cues and time distortion, we let all the participants experience all the conditions to obtain more data. However, this will lead to a learning-rate problem which means the user will perform better in the latter conditions. To fix this problem, we shuffled the latter three conditions (The cond. R and cond. V are treated as baseline in this experiment) to counterbalance the learning rate. Three conditions resulting in 6 permutations were done twice by the 12 participants, while each participants have to return 300 spin balls in total.

The only difference is that we asked the participants to use the same haptic racket in Cond. R which would be used in VR later, which could provide a smoother transfer to VR. In addition, a later real world training (Cond. RL) is done to show the training effect versus robot. However, since the participants might be tired after returning 300 shots, this condition is performed on another day 3 weeks later. Here, we decide to wait for 3 weeks to also study the long-term training effect.

The 3 questions that were asked for each condition were analyzed similarly to the first evaluation. To study the statistical significance here, we use the Wilcoxon Signed Rank tests to study the difference between each condition in each question.

The qualitative evaluation is obtained by a 6-point Likert scale questionnaire asking 3 questions to study the previous RQ1∼4:

- Q1: "Did you have fun in this condition?" This is designed to evaluate the motivation of the user, which is related to RQ2.
- Q2: "Did this condition improve your understanding of spinning shots?" This question is related to RQ3 of understanding of spin.
- Q3: "Do you think your skill improved in this condition?" This question qualitatively answers the RQ1, together with the result from the performance evaluation.

The questionnaire is given to the users right after each condition to obtain timely feedback from them. After the whole study, similar to the initial study, an oral interview was given to the users to ask about their overall impression of all the conditions. There was also an independent yes-no question in order to find out whether the user noticed the difference of the haptic feedback after entering the VR world without being told in advance, which is for RQ4.

Condition	S. R.	N. F. R.	S. F. R.	M. T. D. (m)
Cond. R	11.25%	9.58%	79.17%	0.81
Cond. V	8.89%	**9.03%**	82.08%	0.83
Cond. S	22.78%	15.56%	61.67%	0.62
Cond. B	14.86%	19.03%	66.11%	0.68
Cond. G	**25.83%**	15.14%	**59.03%**	**0.59**

Table 7.1: The mean performance using the 4 metrics of each condition, M.T.D.: Mean Table Distance.

7.3.3 Results

In this evaluation, we want to answer the RQ1, RQ3 and RQ5 using several quantitative numbers, which means we compare the 3 new conditions with the 2 baselines and also compare within the 3 conditions. The 2 baseline conditions are not compared with each other. First, by logging the ball trajectory in VR as well as tracking the ball using cameras in the real world, we calculated the distance from the raw data as described in the performance metrics section. The result is then averaged by each condition which is shown in Table 7.1.

On the other hand, the absolute percentage cannot fully represent the improvement in the specific condition since the order might infect the result. Therefore, to also obtain the relative improvement in each condition, we calculate the difference of each metric between before and after each condition, which is stated as $\Delta S.R.$, $\Delta N.F.R.$, $\Delta S.F.R.$, and $\Delta M.T.D.$ in Table 7.2. Hereby only the latter 3 conditions, which are shuffled for counterbalance, are compared.

We also conducted a one-way repeated-measures ANOVA ($\alpha = .05$) on all the metrics and a post-hoc Tukey's range test was performed on the Success Rate and the M.T.D. metrics, which are the two most representative metrics for the user's performances.

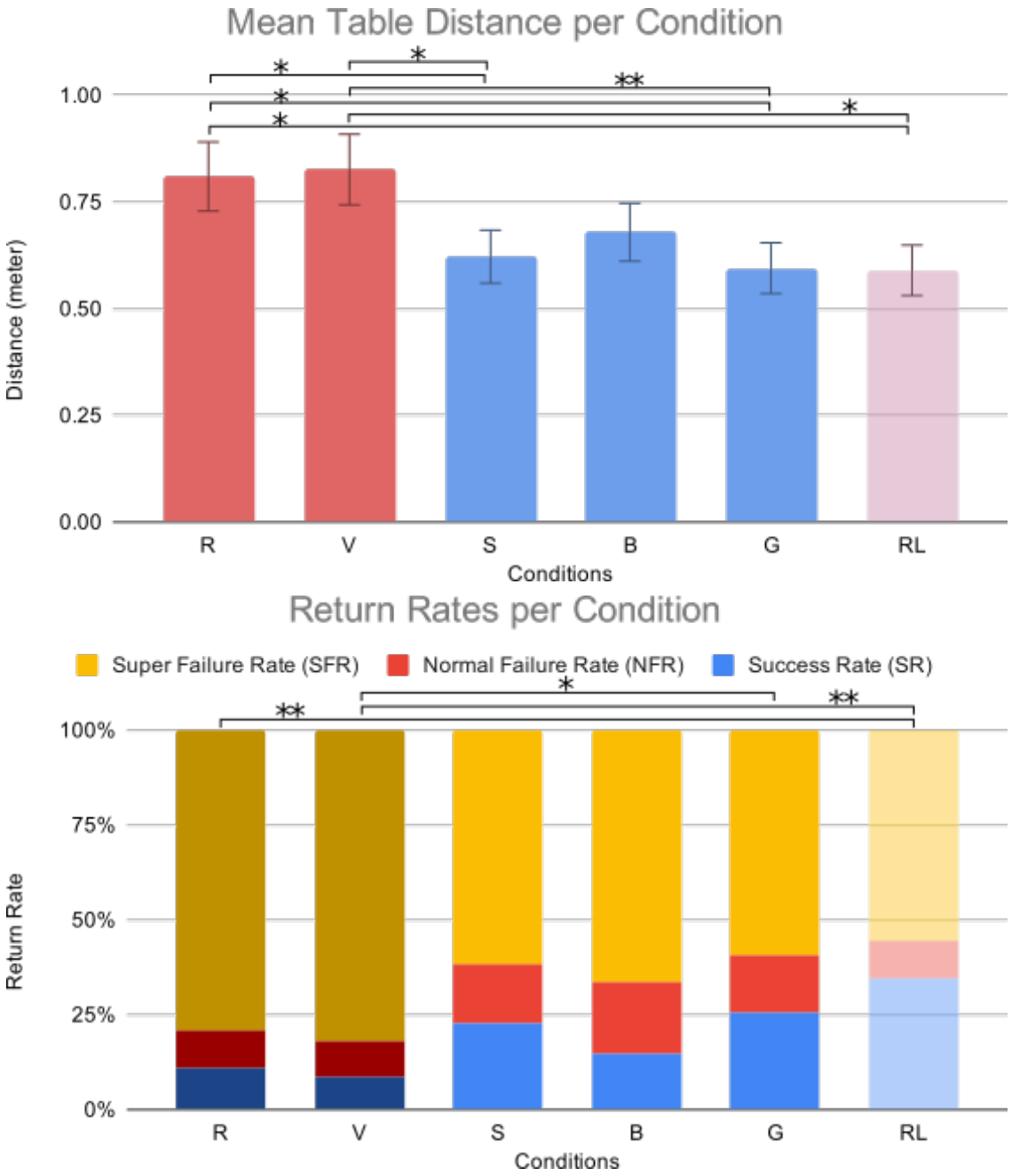


Figure 7.9: Quantitative results for the M.T.D (top) and the S.R (bottom, blue bar), The brackets on the top indicate the significance between the conditions: *($p < .05$), **($p < .01$).

Quantitative Results

From Table 1, it is very obvious that the success rate greatly improved in the 3 new conditions compared to the 2 baseline condition, among them the Future

Guidance Condition (Cond. G) almost performs the best for all metrics except for N.F.R., while Table 2 also shows a similar result that the performance improved most in Cond. G. Also, when taking the Later Real-world condition (Cond. RL) into account, we surprisingly found that the users performs even better after 3 weeks in the real-world, with a 22.5% increase in success rate comparing with the initial Cond. R.

By analyzing the data of Table 1, the result of ANOVA indicates that a significant difference between the conditions could be detected in all metrics (S.R.: $F_{4,55} = 9.232$, $p < 0.00001$; S.F.R.: $F_{4,55} = 11.949$, $p < 0.00001$; N.F.R.: $F_{4,55} = 5.731$, $p < 0.001$); M.T.D.: $F_{4,55} = 15.94$, $p < 0.00001$). However, for Table 2, ANOVA suggests that the data is not significantly different for $p < 0.05$.

Next, Fig. 7.9 shows a detailed chart of the M.T.D. and the S.R. metrics for each condition. In the M.T.D., the Tukey HSD Test suggests that both Spin Arrow and the Future Guidance are significantly better than the two base conditions. (S-R:$t = 0.035$, $p < 0.05$, S-V:$t = 0.021$, $p < 0.05$, G-R:$t = 0.011$, $p < 0.05$, G-V:$t = 0.007$, $p < 0.01$). The performance in the S.R. metric is slightly different, even though the 3 new conditions all achieve twice as high values as the 2 base conditions, the Tukey test suggests there is only one significant difference between the Guidance condition and the Base condition. When we include the later real world training (Cond. RL), the new differences are significant as well, according to the ANOVA (S.R.: $F_{5,66} = 4.4268$, $p < 0.01$; M.T.D.: $F_{5,66} = 4.6854$, $p \leq 0.001$) as well as the Tukey HSD Test (S.R: RL-R:$t = 0.007$, $p < 0.01$, V-R:$t = 0.002$, $p < 0.01$, M.T.D.: R-RL:t$t = 0.033$, $p < 0.05$, V-RL:$t = 0.02$, $p < 0.05$). In the discussion section, we will discuss how these results answer our research questions in detail.

Qualitative Results

The result of the 6 point Likert-scale questionnaire is shown in Fig. 7.10. The side brackets indicate the significance between each condition. By observing the

Condition	Δ S.R.	Δ N.F.R.	Δ S.F.R.	Δ M.T.D. (m)
Cond. S	9.03%	**-0.56%**	-8.47%	-0.08
Cond. B	5.97%	2.78%	-3.19%	-0.02
Cond. G	**10.42%**	0.28%	**-10.69%**	**-0.11**

Table 7.2: The differences between before and after each VR condition.

overall result, the 3 new conditions all achieved over 80% positive answers, which is much better than the two base conditions. Especially in Q2, which is about the understanding of spin, the two base conditions have half of the negative responses while the 3 new visualizations did not get a score lower than Slightly Disagree.

The Wilcoxon test also shows a lot of significant differences between the new conditions and the base ones. In particular, in Q2, all the new conditions are highly significantly better than the two base conditions. The Bullet Time condition is the only condition which doesn't show significant results in Q1 and Q3.

The results of the oral interview are also very interesting. The participants were asked about their overall favorite condition and the least liked condition. The result of the first question was controversial, the 12 participants perfectly divided into 3 groups for the 3 new conditions (4 for each), respectively. On the other hand, the trend of the most dislike one was quite clear, 6 out of 12 participants chose the Real Robot condition while another 3 participants chose the basic VR condition, the two base conditions shared 75% of the negative votes. However, it was surprising to see that the remaining 3 participants all voted for the Bullet Time condition as their least preferred condition. Lastly, all participants claimed that they didn't notice the haptic feedback on the racket was fake before we mentioned it in the interview.

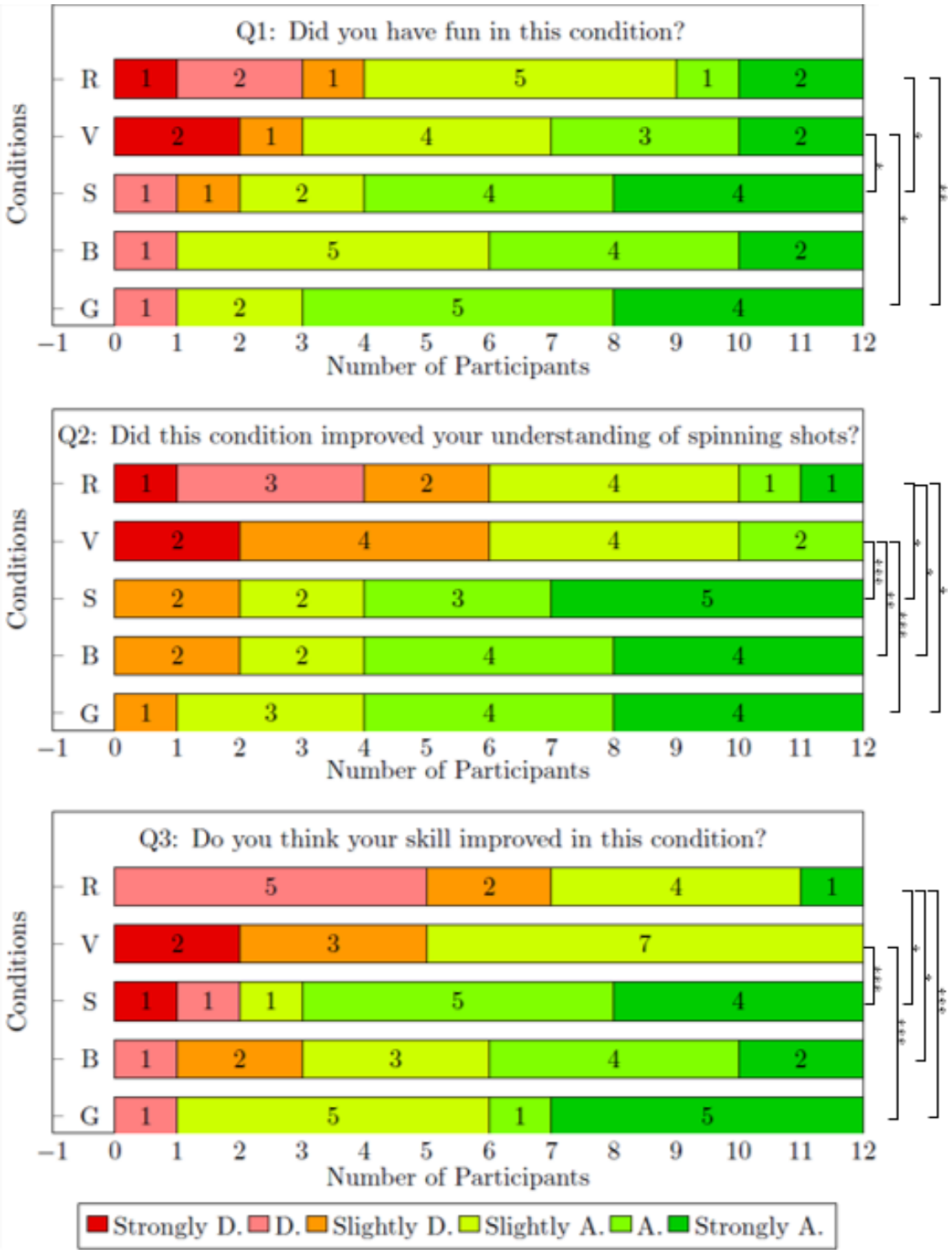


Figure 7.10: Questionnaire results for each study condition in a 6-Likert Chart from Strongly Disagree to Strongly Agree. The brackets indicate the significance between conditions: * ($p < .05$), ** ($p < .01$), *** ($p < .005$).

Chapter 8

Discussion

In this chapter, we will conclude the result of both the quantitative evaluation and the qualitative evaluation and discuss the findings from the result.

8.1 Discussion on FuturePoseNet

To conclude, in both the real-time test and the accuracy test, our method is proved to have the most balanced performance. In the accuracy of 3D recovery, the specific joint position of our system also has a lower average error than the 3DPF-Net. Especially, the positions of wrists and ankles are far more accurate than 3DPF-Net, which are more important in some specific sports such as martial arts. In the real-time test, our method(stack 5) has an equal performance to the Kinect Depth camera + neural network method, which is a hardware-based method and has severe environment dependencies.

From the quantitative evaluation experiments, it can be seen that our system did not fall behind other state-of-the-art real-time methods in inference time and had an equal or even better accuracy comparing to offline methods such as 3DPF-Net. The user study also shows that the user did not notice the difference of pose estimation accuracy between our RGB-based method and Kinect depth camera based method, which means our RGB-based method is possible to replace Kinect-based pose forecasting method for its wide usability.

In the condition of forecasting 0.5s future pose (15 frames in 30-fps video), it's obvious that our method greatly outperformed other real-time method in PCK eval-

uation as well as the 3DPF-Net off-line method, with a average of 60% PCK and especially in boxing with a accuracy of 70.6%. In the other condition of forecasting 1s future pose (30frames in 30-fps video), while the accuracy of other method decrease apparently, our method still out-perform the Nearest Neighbor baseline in most type of action, in which the boxing still have a 65% accuracy which is the state-of-the-art of 30-frame prediction.

8.2 Discussion on InvisiblePoseNet

From the results of the four studies in hand pose estimation, we could imagine a clear picture about the performance, with an almost half the angle-error than the other baseline. To notice, here we only compared with vision-based techniques because the main focus of this study is system that could be naturally embedded in wearable devices. The result of each joint shows that the index, middle, and ring finger gain a relatively high accuracy estimated by back of the hand features. And, the result of an average angle error of 8.81° for individual and 9.77° in general even outperforms some methods using TPV camera [81] where the fingers can be clearly seen. Also, the result of lighting condition study and the ablation study could provide information which might be helpful in developing robust networks for extracting temporal deformations.

For the hand segmentation, we augmented the data by changing the color or brightness of the dorsal hand and achieved a high accuracy, but it is not sufficient to claim our network is robust to different types of hand, without testing on diverse users. There are multiple factors that might affect the result, such as skin color, skin thickness, hair volume, hand shape, etc. However, to note that, from the ablation study, we can observe that the motion-only input performs very close to the two-stream input, and surpasses the RGB-only input. From which we can tell the network is more looking at the overall deformations than the color information of the hand. Also, one of our participants had hairy skin and the features

are still successfully extracted (which is not enough to claim this generalizability). Nevertheless, in the use case of wearing a personal smartwatch, the system is not necessary to be generalized but can be initially calibrated to the user by collecting a small amount of data and fine-tuning the model, this will result in a personalized model with higher accuracy and might also work as a security identification using the dorsal hand.

Lastly, in the feet pose estimation, even though the system and the study is relatively prototype, the results suggests that the current feet pressure-based system has the potential to achieve similar precision as vision-based methods. Especially in the case of skiing, it is more acceptable to equip an insole sensor instead of a third-person-view camera. However, the results also indicates that the current system is less robust across different users. The feet pressure needs to be normalized to people with different feet size, shape, and different boots. Also, more data across different types of skier need to be collected for a more comprehensive study.

8.3 Discussion on Three Applications

8.3.1 Skiing

Performing studies on visualizing expert ski motion provided us with a number of interesting insights and surprising results.

One surprise was that the performance measures of the *Pose Breakdown* condition showed a much clearer picture than the controversial discussions about it in the first evaluation. Its performance is considerably worse than comparable conditions, such as *Color Trail* and *Ground Shadow*, which shows that our developments went into the right direction. We assumed that the *Color Trail* would do well regarding lateral movement, but we were positively surprised that users still had a good performance regarding ankle rotation.

Another surprise was for the *Graph Feedback* condition, which address both as-

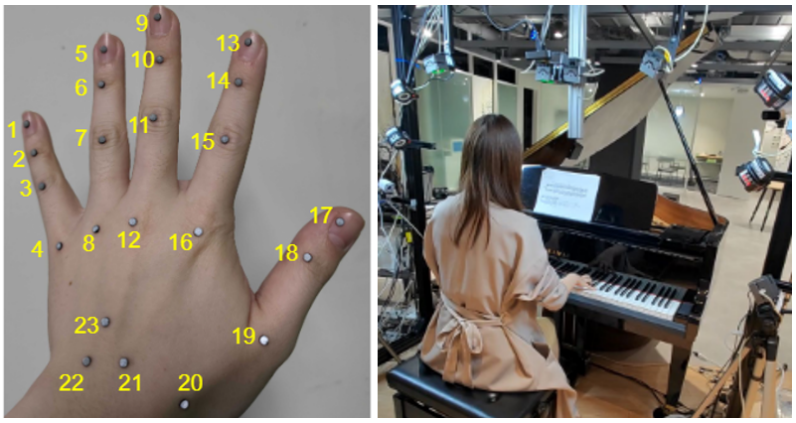


Figure 8.1: Conventional way of piano hand motion capture.

pects of providing feedback and providing motion information. However, its complexity makes it hard for users to effectively use the feedback. We assumed that participants might be more positive when they had some time to get familiar with the visualization in a training session as we ourselves got used to it during development, however, this was not the case.

In summary, we can conclude that using a *Color Trail* to provide motion feedback is the best option in our designed conditions. Based on the current results implementing feedback does not necessarily provide benefits and therefore needs to be carefully considered. Even though not tested directly, a combination with the *Shadow* condition might be interesting and could be considered.

8.3.2 Piano

From the pilot study, we can observe a trend that using temporally aligned videos is better than just viewing the original videos. Also, both proposed methods outperform the two baselines which suggests the effectiveness of the proposed 3D visualization. Between the two proposed methods, the majority of participants prefer the virtual harpsichord. From the later interview, we assume that this is because the side-by-side visualization and the error timeline provide better feedback on "when" and "where" the differences happen. On the other hand, one participant prefers

the motion overlay and comments that it is a novel way to overlay two hands and is not possible in real training, which is more intuitive.

However, this system is still a prototype and has its shortcomings. The current system relies on marker-based motion capture and cannot be used for online videos. Recent deep learning-based methods can perform 3D pose estimation from 2D videos which can be applied to our system in the future. Also, motion capture has some noises which cause independent errors on the timeline. This may be overcome by adding a filter to preprocess the data.

8.3.3 Table Tennis

Overall, our improved version of a VR training system does show a great improvement in both quantitative performance and qualitative evaluation. The three new functions all outperform the two baseline conditions with a significant statistical difference. Among them, the Future Guidance condition which employs our proposed prediction system performs the best. The qualitative user study also shows a similar result, the popularity of Future Guidance is among the highest on average, with a significant difference compared to the base V and R conditions.

In order to better understand the reason why the future trajectory improve the training, we interviewed most participants about their feelings in detail. According to our assumption and interview results, showing the future trajectory together with

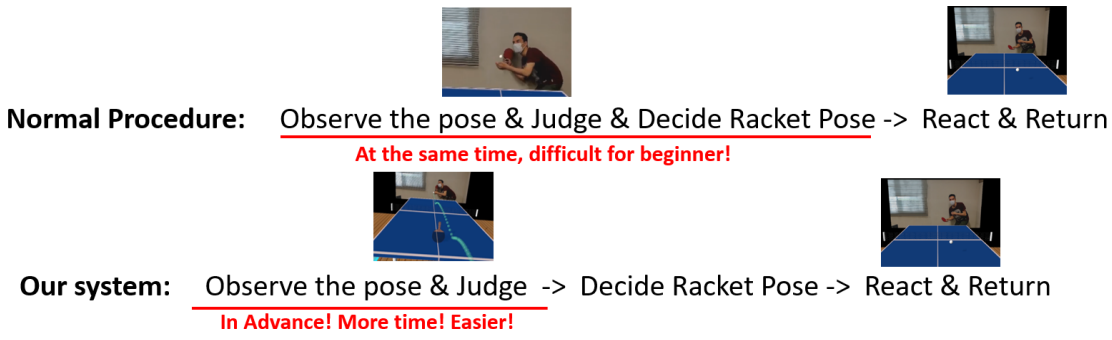


Figure 8.2: Comparison of the Procedure of returning a serve.

the serve motion makes it easier for learner to understand how opponent's posture is related with specific spin types. As shown in Figure 8.2, A normal procedure (for a beginner) of returning a serve is that the receivers need to observe the pose, judge the spin, and decide their way of return at the moment when the ball is served, which is technically difficult. When using the prediction system, as shown in the procedure below, the learner can observe the pose and judge before the ball is served, thus they have enough time to decide the racket pose and return the ball. Therefore, we assume that providing a future and current side-by-side training is effective for skill acquisition.

Chapter 9

Future Vision

9.1 Future Improvements

• In this paper, we studied different indirect feature-based human pose prediction, and apply the proposed method to different skill acquisition including sports and musical instruments to study its training effect. However, currently we decide the way of using pose prediction for each application arbitrarily. A detailed study on how pose prediction shall be applied to different types of skill need to be further studied.

• For the participants of training, except the piano training which is a collaboration with industry, other skill acquisition applications are only studied on beginner which are mainly students within the University. Although it is difficult to collect dataset and invite professional athletes from all types of sports, we will try to ex-

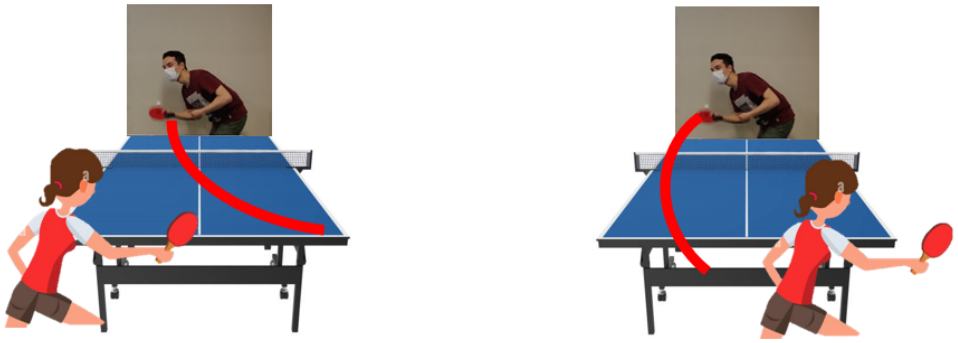


Figure 9.1: Interactive skill also depends on opponent's pose and position.

pand the dataset to more area of sports and perform study as much as possible in future work.

• In the quantitative evaluation, we mainly focus on inference time and accuracy on different algorithms. It would be better if we can analyze some hyper-parameter like d for the lattice point optical flow and threshold for the motion history images.

• The proposed system uses dual-modal networks for temporal and spatial indirect features, which enhance some indirect pose regression which benefits from both type of features. However, in case of some activity which mainly requires a single type of feature, the network might be bulky for it. We aim to design a non-supervised classification network to decide which kind of feature is needed for a specific input and output in the future.

• Currently, our system is only single person oriented. Designing the network to be multi-person oriented is possible but will lead to heavier computation. We can also place two cameras between them and estimate their pose separately, however, in that case, the system cannot learn interactions between people, which is quite important in some interactive or competitive skill. Therefore, we are trying to use

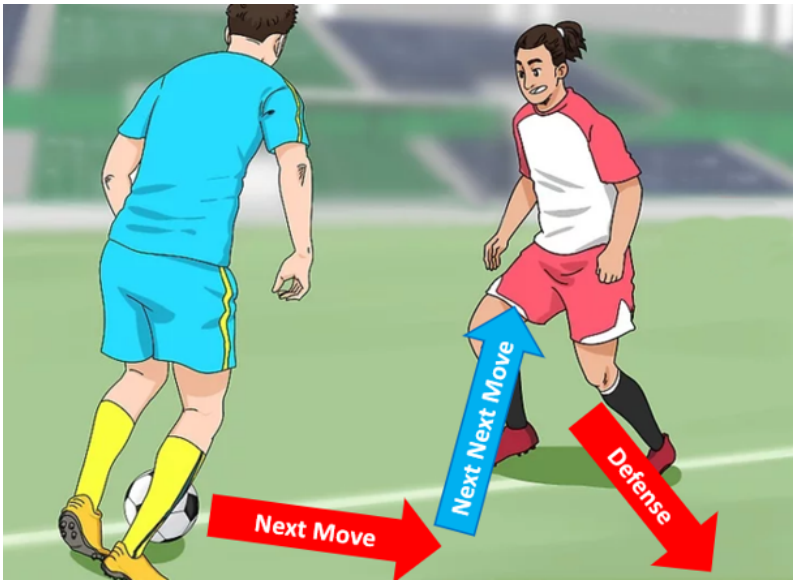


Figure 9.2: Predict the "Next Next" movement from the prediction of both players.

an omni-directional camera to capture human pose from all directions and change the network to parallel computing to make it possible to estimate multiple-people with multiple GPUs, and we also believe that with the development of graphic units, faster hardware could also solve this problem.

9.2 Future Applications

During the developing and study, we found that this technology cannot only be used in skill acquisition, but also in many artificial intelligence-related area. The behavior of prediction is not only used in motor skills such as sports, but also in our daily life.

For example, when we are stumbled and falling down, our hand will involuntarily support us from being hurt. This is because our brain predict the danger from those indirect features such as the shock to your shoes. If this mechanism can be imitated by neural network, there will be a wide field of application in many industries, such as robotics, medical care, human augmentation, etc.

Currently, we have also already started some new projects in distracted walking/driving where using indirect feature-based prediction can be helpful in estimating a risk degree of the user's surroundings. Different from simple direct detection, an indirect feature-based method can be more context-aware because it also extracts less related features such as user's attention or the characteristics of a potential obstacle.

Furthermore, the idea of applying the system to support distracted walking has already being awarded the 7th AIP Network Director Award, while the prediction system itself has being awarded an honorable mention in the international conference of VRST. It is obvious that the research is attracting attention by researchers and we believe it has the potential to change the AI industry after being improved more robust and accurate.

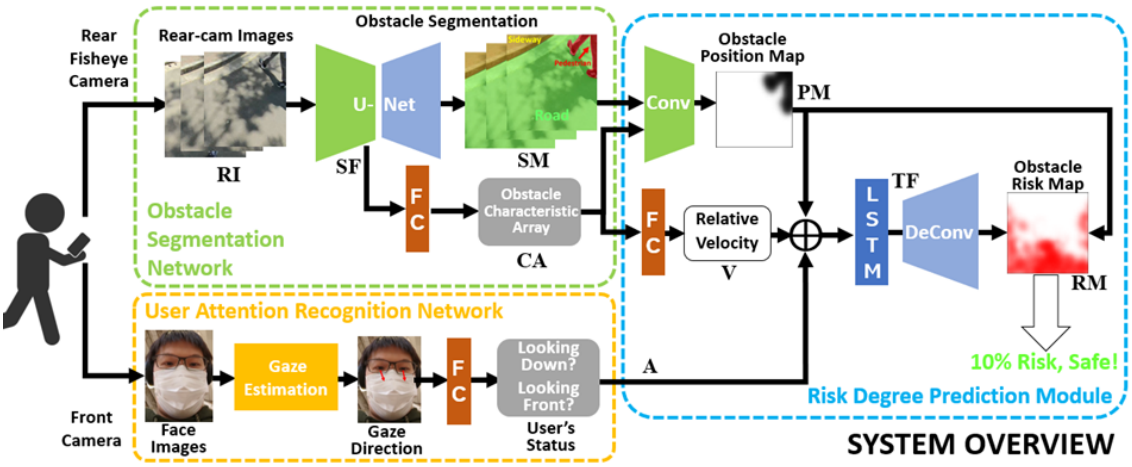


Figure 9.3: A future composition for risk prediction application.

Chapter 10

Conclusion

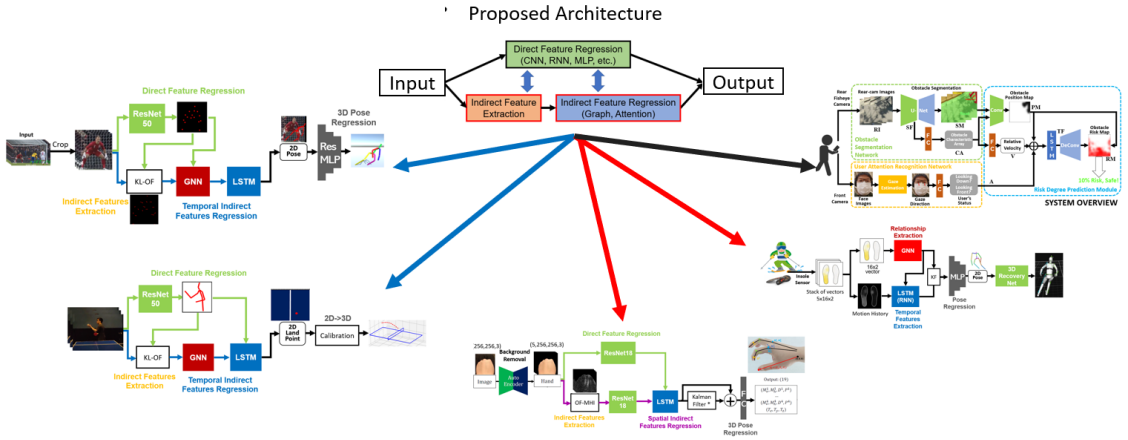


Figure 10.1: Summary of our approach.

10.1 Contribution of this Work

Hereby, from all the work mentioned in this paper, we can summarize the contribution of this work as follows:

1. We proposed a novel architecture for indirect feature-based prediction.
2. Our work is the first to realize real-time pose prediction based on two types of indirect features.
3. The proposed architecture has a good generality and modification ability.

4. Numbers of different networks using indirect spatial-/temporal features are introduced in this dissertation.

5. Quantitative and qualitative experiments are conducted to study the effect of the proposed network.

6. Three different types of training application for skill acquisition are developed using our system.

7. Comprehensive studies are also performed for the training application to prove the concept.

10.2 Summary

In this paper, we presented an indirect feature-based pose estimation system using a dual-module two stream deep neural network. The proposed system is also applied to skill acquisition and enables some new possibility for AI-based motor skill training.

Different from conventional direct feature-based pose estimation, the proposed system try to make use of those features which are indirect related with human posture (e.g. estimate full body pose from feet pressure).

The proposed network consists of twp parts: a FuturePoseNet which aims to extract temporal indirect features from the input video sequences and a Invisible-PoseNet which finds out the spatial indirect relationship within each images. For each network, a special indirect feature extraction module is developed to enhance the learning of an indirect feature. The performance of both networks are quantitatively and qualitatively evaluated in the experiment, and the results suggest that the proposed indirect feature-based prediction can achieve similar accuracy as the conventional methods, without observing the direct features.

For applications, three types of different skill are introduced: Skiing, Piano, and Table Tennis, which aims to study the results from three perspectives. The Skiing

is mainly focus on spatial indirect features while the piano requires temporal one. Table Tennis is the most well-studied application which includes both temporal and spatial indirect features.

To the best of our knowledge, this work is the first real-time 3D pose prediction using a dual-module indirect feature-based network, which is proved to be useful in different types of skill training.

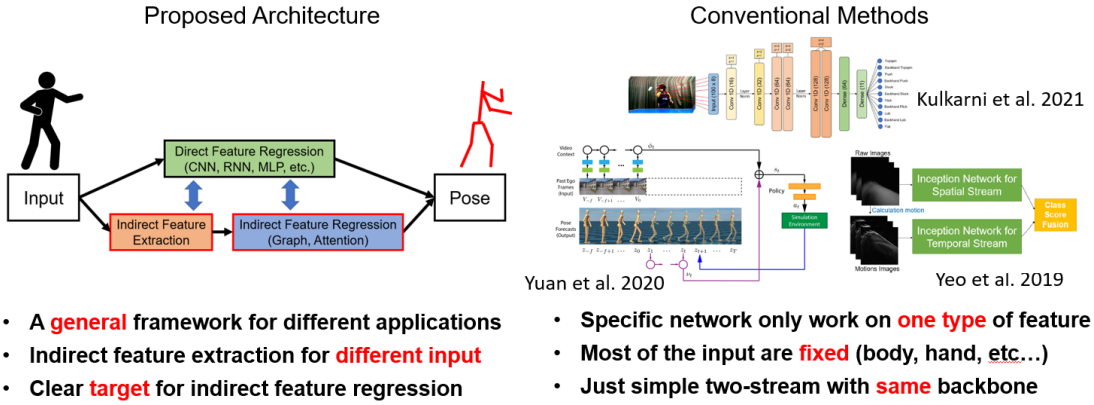


Figure 10.2: Comparison between the proposed framework and existing methods.

Acknowledgement

First and most importantly, the author gives thanks to Professor Hideki Koike, for the great guidance, constant feedback and suggestions that were made to allow this project to proceed as it did. Many thanks also goes to Associate Professor Shio Miyafuji for her advice.

Next, the author would also like to appreciate all students from the Koike Lab and the secretary Setsuko Mizoguchi for their support during the five years student life.

In terms of the user study, acknowledgments to all participants and other guests that gave important suggestions. Thanks are also given to those who took part in experiencing the system during the conferences. The author also want to give a really special thanks to the JST CREST, JSPS Gakushin for funding both the author and the project.

The same acknowledgments are also given to the Tokyo Institute of Technology, for waiving the tuition fees so that the author can concentrate on study, also appreciate the great resources and wonderful environments for doing research.

Finally, the author would like to thank his wife, Shiyu Wu, for the constant support during his whole student life. It is appreciated for her thoughtful kindness which gave the author endless vigor to face any difficulties.

Bibliography

[1] Convolutional neural networks for visual recognition. `https://cs231n.github.io/convolutional-networks/`. [Online; accessed 13-July-2019].

[2] V Aleshin, S Klimenko, M Manuilov, and L Melnikov. Alpine skiing and snowboarding training system using in-duced virtual environment. *Science and Skiing IV*, 2009.

[3] Vladimir Aleshin, Valery Afanasiev, Alexander Bobkov, Stanislav Klimenko, Vitaly Kuliev, and Dmitry Novgorodtsev. Visual 3d perception of the ski course and visibility factors at virtual space. In *Cyberworlds (CW), 2011 International Conference on*, pages 222–226. IEEE, 2011.

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, Dec 2017.

[6] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[7] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2272–2281, 2019.

[8] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

[9] Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011.

[10] Yu-Wei Chao, Jimei Yang, Brian Price, Scott Cohen, and Jia Deng. Forecasting human dynamics from static images. In *IEEE CVPR*, 2017.

[11] Henry Chen, Austin S Lee, Mark Swift, and John C Tang. 3d collaboration method over hololens™ and skype™ end points. In *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, pages 27–30. ACM, 2015.

[12] Po-Jung Chen, I-Wen Penn, Shun-Hwa Wei, Long-Ren Chuang, and Wen-Hsu Sung. Augmented reality-assisted training with selected tai-chi movements improves balance control and increases lower limb muscle strength in older adults: A prospective randomized trial. *Journal of Exercise Science & Fitness*, 18(3):142–147, 2020.

[13] Henry M Clever, Ariel Kapusta, Daehyung Park, Zackory Erickson, Yash Chitalia, and Charles C Kemp. 3d human pose estimation on a configurable bed from a pressure image. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 54–61. IEEE, 2018.

[14] H. Coskun, F. Achilles, R. DiPietro, N. Navab, and F. Tombari. Long short-term memory kalman filters: Recurrent neural estimators for pose regularization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5525–5533, Oct 2017.

[15] Pádraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers. 2007.

[16] Paul Dempsey. The teardown: Htc vive vr headset. *Engineering & Technology*, 11(7-8):80–81, 2016.

[17] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1801–1810, 2019.

[18] Looking Glass Factory. Looking glass factory · the world's first desktop holographic display. `https://lookingglassfactory.com/`. [Online; accessed 13-July-2019].

[19] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[20] Ian Fitz. The thrill of the fight - vr boxing, 2018.

[21] Werner Goebl and Caroline Palmer. Temporal control and hand movement efficiency in skilled music performance. *PLOS ONE*, 8(1):1–10, 01 2013.

[22] Perttu Hämäläinen, Tommi Ilmonen, Johanna Höysniemi, Mikko Lindholm, and Ari Nykänen. Martial arts in artificial reality. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 781–790. ACM, 2005.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Susan Higgins. Motor skill acquisition. *Physical therapy*, 71(2):123–139, 1991.

[26] Susan Higgins. Motor Skill Acquisition. *Physical Therapy*, 71(2):123–139, 02 1991.

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[28] Yuuki Horiuchi, Yasutoshi Makino, and Hiroyuki Shinoda. Computational foresight: Forecasting human body motion in real-time for reducing delays in interactive system. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*, pages 312–317. ACM, 2017.

[29] Atsuki Ikeda, Dong-Hyun Hwang, and Hideki Koike. Ar based self-sports learning system using decayed dynamic timewarping algorithm. In *ICAT-EGVE*, pages 171–174, 2018.

[30] 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.

[31] 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.

[32] Jintae Lee and T. L. Kunii. Model-based analysis of hand posture. *IEEE Computer Graphics and Applications*, 15(5):77–86, Sep. 1995.

[33] 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 *The IEEE International Conference on Computer Vision (ICCV)*, 2015.

[34] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[35] Makio Kashino. Understanding and shaping the athlete's brain using body-mind reading and feedback. In *Proceedings of the 1st International Workshop on Multimedia Content Analysis in Sports*, pages 1–8. ACM, 2018.

[36] Makio Kashino. Understanding and shaping the athlete's brain—ntt sports brain science project. *NTT Technical Review*, 16(3), 2018.

[37] L and L Technology. Virtual fighting championship, 2018.

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

[39] Cheng-Chang Lien and Chung-Lin Huang. Model-based articulated hand motion tracking for gesture recognition. *Image and Vision Computing*, 16(2):121 – 134, 1998.

[40] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[41] Chih-Yao Ma, Min-Hung Chen, Zsolt Kira, and Ghassan AlRegib. Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *arXiv preprint arXiv:1703.10667*, 2017.

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

[43] Mikihito Matsuura, Shio Miyafuji, Erwin Wu, Satoshi Kiyofuji, Taichi Kin, Takeo Igarashi, and Hideki Koike. Cv-based analysis for microscopic gauze suturing training. In *Augmented Humans Conference 2021*, AHs'21, page 169–173, New York, NY, USA, 2021. Association for Computing Machinery.

[44] 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.

[45] 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.

[46] 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 motion capture with a single RGB camera. volume 39, 2020.

[47] 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. volume 36, 2017.

[48] Takayuki Nozawa, Erwin Wu, and Hideki Koike. Vr ski coach: Indoor ski training system visualizing difference from leading skier. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, page D16. IEEE, 2019.

[49] Natural Point. Inc.: Optitrack-optical motion tracking solutions. `https://optitrack.com/`, 2009. [Online; accessed 13-July-2019].

[50] Patrik Puchert and Timo Ropinski. Human pose estimation from sparse inertial measurements through recurrent graph convolution. *arXiv preprint arXiv:2107.11214*, 2021.

[51] N. Dinesh Reddy, Minh Vo, and Srinivasa G. Narasimhan. Occlusion-net: 2d/3d occluded keypoint localization using graph networks. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[52] Daniel Roetenberg, Henk Luinge, and Per Slycke. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep*, 1, 2009.

[53] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[54] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[55] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[56] 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 *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[57] Hava T. Siegelmann and Eduardo D. Sontag. On the computational power of neural nets. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 440–449, New York, NY, USA, 1992. ACM.

[58] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[59] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[60] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, June 2018.

[61] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *CoRR*, abs/1902.09212, 2019.

[62] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[63] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[64] Denis Tome, Patrick Peluse, Lourdes Agapito, and Hernan Badino. xr-egopose: Egocentric 3d human pose from an hmd camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7728–7738, 2019.

[65] Sam Toyer, Anoop Cherian, Tengda Han, and Stephen Gould. Human pose forecasting via deep markov models. In *2017 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2017, Sydney, Australia, November 29 - December 1, 2017*, pages 1–8, 2017.

[66] Du-Ming Tsai, Wei-Yao Chiu, and Men-Han Lee. Optical flow-motion history image (of-mhi) for action recognition. *Signal, Image and Video Processing*, 9(8):1897–1906, 2015.

[67] Josien C. van den Noort, Henk G. Kortier, Nathalie van Beek, DirkJan H. E. J. Veeger, and Peter H. Veltink. Measuring 3d hand and finger kinematics—a

comparison between inertial sensing and an opto-electronic marker system. *PLOS ONE*, 11(11):1–16, 11 2016.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[69] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[70] E. Wu and H. Koike. Futurepose - mixed reality martial arts training using real-time 3d human pose forecasting with a rgb camera. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1384–1392, Jan 2019.

[71] Erwin Wu, Mitski Piekenbrock, Takuto Nakumura, and Hideki Koike. Spinpong - virtual reality table tennis skill acquisition using visual, haptic and temporal cues. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2566–2576, 2021.

[72] Xiao Xiao, Michael S. Bernstein, Lining Yao, David Lakatos, Lauren Gust, Kojo Acquah, and Hiroshi Ishii. Pingpong++: Community customization in games and entertainment. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, ACE '11, New York, NY, USA, 2011. Association for Computing Machinery.

[73] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[74] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future

person localization in first-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7593–7602, 2018.

[75] Yiding Yang, Zhou Ren, Haoxiang Li, Chunluan Zhou, Xinchao Wang, and Gang Hua. Learning dynamics via graph neural networks for human pose estimation and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8074–8084, 2021.

[76] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[77] Hui-Shyong Yeo, Erwin Wu, Juyoung Lee, Aaron Quigley, and Hideki Koike. Opisthenar: Hand poses and finger tapping recognition by observing back of hand using embedded wrist camera. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 963–971, New York, NY, USA, 2019. Association for Computing Machinery.

[78] Ye Yuan and Kris Kitani. 3d ego-pose estimation via imitation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 735–750, 2018.

[79] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time pd control. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10082–10092, 2019.

[80] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[81] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. Model-based deep hand pose estimation. In *Proceedings of the Twenty-*

Fifth International Joint Conference on Artificial Intelligence, IJCAI'16, page 2421–2427. AAAI Press, 2016.

[82] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. Monocular real-time hand shape and motion capture using multi-modal data, 2020.