UNIVERSITY OF CALIFORNIA,
IRVINE

Deep Learning Models On Hand Pose Estimation and Mesh Reconstruction From RGB Images

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Deying Kong

Dissertation Committee:
Professor Xiaohui Xie, Chair
Professor Charless C. Fowlkes
Professor Erik B. Sudderth

2022

DEDICATION

To my family.

TABLE OF CONTENTS

Page

LIST OF FIGURES vi

LIST OF TABLES viii

LIST OF ALGORITHMS ix

ACKNOWLEDGMENTS x

VITA xi

ABSTRACT OF THE DISSERTATION xiii

1 Introduction 1
- 1.1 Problem Definition . . . 1
- 1.2 Outline . . . 2

2 Background 6
- 2.1 Probabilistic Graphical Models . . . 6
- 2.2 Deep Convolutional Neural Networks . . . 7
  - 2.2.1 Convolutional Pose Machine . . . 7
  - 2.2.2 Stacked Hourglass . . . 8
- 2.3 Integrating CNNs and Graphical Models . . . 9
- 2.4 Graph Neural Networks . . . 10
- 2.5 Generative Hand Models . . . 11

3 Rotation-invariant Mixed Graphical Model Network 13
- 3.1 Introduction . . . 14
- 3.2 Related Work . . . 16
- 3.3 Methodology . . . 18
  - 3.3.1 Basic pipeline . . . 18
  - 3.3.2 Model . . . 19
  - 3.3.3 Detailed structure of the R-MGMN . . . 22
- 3.4 Learning . . . 26
  - 3.4.1 Train Rotation Net . . . 27
  - 3.4.2 Train Unary Branch . . . 28
  - 3.4.3 Train Soft Classifier . . . 28

3.4.4 Train Graphical Model Parameters . . . 29
3.4.5 Jointly Train All the Parameters . . . 29
3.5 Experiments . . . 30
3.5.1 Experimental settings . . . 30
3.5.2 Results . . . 31
3.5.3 Ablation study . . . 33
3.6 Conclusion . . . 34

4 Adaptive Graphical Model Network 35
4.1 Introduction . . . 35
4.2 Related Work . . . 38
4.3 Method . . . 39
4.3.1 Basic Framework of Adaptive Graphical Model Network . . . 39
4.3.2 Detailed Structure of AGMN . . . 42
4.4 Leaning . . . 45
4.5 Experiments . . . 47
4.5.1 Experimental settings . . . 47
4.5.2 Results . . . 48
4.6 Conclusion . . . 50

5 SIA-GCN: A Spatial Information Aware Graph Neural Network with 2D Convolutions 52
5.1 Introduction . . . 53
5.2 Related Work . . . 55
5.3 Methodology . . . 57
5.3.1 Revisiting Graph Convolutional Network . . . 58
5.3.2 SIA-GCN . . . 60
5.3.3 SiaPose and its training procedure . . . 61
5.4 Experiments . . . 63
5.5 Conclusion . . . 68

6 Identity-Aware Hand Mesh Estimation and Personalization from RGB Images 69
6.1 Introduction . . . 69
6.2 Related Work . . . 72
6.3 Method . . . 74
6.3.1 MANO Model . . . 74
6.3.2 Identity-aware Hand Mesh Estimation . . . 75
6.3.3 Baseline Method . . . 77
6.3.4 Personalization Pipeline . . . 77
6.3.5 Loss Functions . . . 79
6.4 Experiments . . . 81
6.4.1 Experimental Setups . . . 81
6.4.2 Quantitative Evaluation . . . 84
6.4.3 Ablation Study . . . 85

6.5 Conclusion 89

7 Conclusion **90**

Bibliography **92**

LIST OF FIGURES

Page

1.1 Definition of the problem. Left: input RGB image of the hand. Middle: detected 2D keypoints overlaid on the image. Right: 3D hand mesh reprojected onto the image. [141] 2

2.1 Architecture and receptive fields of convolutional pose machine in [124]. . . . 8
2.2 The staked hourglass network for pose estimation consists of multiple modules which allow for repeated bottom-up, top-down inference [75]. 9
2.3 Didactic example of message passing between the face and shoulder joints [107]. 10
2.4 PCA shape space of MANO model [90]. 12
2.5 PCA pose space of MANO model [90]. 12

3.1 Pipeline overview of the proposed Rotation Mixture Graphical Model Network (R-MGMN). 18
3.2 Configuration of the rotation net. 22
3.3 Configuration of the soft classifier. 23
3.4 Tree-structured graphical model for hand keypoints. 25
3.5 Illustration of the rotation. Left image courtesy to [96] 27
3.6 PCK performance on two public datasets. 31
3.7 Qualitative results. First row: CPM. Second row: our model. 33

4.1 Basic flow diagram of adaptive graphical model network. 40
4.2 More detailed inllustration of adaptive graphical model network. 42
4.3 Tree structured model and message passing schedule. 44
4.4 Model performance. 49
4.5 Predicted hand keypoint positions. For each pair of images, the top image shows the result of CPM and the bottom image shows that of AGMN. 51

5.1 System diagram of the SiaPose, utilizing SIA-GCN. 57
5.2 A simple illustration of SIA-GCN. 60
5.3 Qualitative results of baseline (top) and our model (bottom) on Panoptic and MPII. 67
5.4 Failure cases of our model. Each pair contains an input image and its prediction. 68

6.1 Overview of our proposed identity-aware hand mesh estimation model. The model mainly contains three parts, i.e., the iterative pose regressor, the 2D detector and the optimization module. Note that in our proposed model, along with the RGB image, we also feed the user's identity information, i.e. the ground truth or calibrated MANO shape parameters of the user. 75
6.2 Proposed personalization pipeline with attention mechanism. Images used for personalization capture the same subject who is never seen during training. . 78
6.3 Hand shape consistency comparison between our proposed method and the baseline. The x-axis corresponds to different subjects in the test dataset, while the y-axis corresponds to the length of the hand of each subject. 86
6.4 Impact of the number of images used in calibration. 86
6.5 Qualitative results. a) Left: calibrated hand model versus ground truth hand model. b) Right: visualization of our identity-aware hand mesh estimator. From top row to bottom row are the input RGB images, the projected ground truth meshes, the projected predicted meshes, and the predicted meshes viewed from two different angles. 88

LIST OF TABLES

Page

3.1 Detailed numerical results of PCK performance. 32
3.2 Numerical results for ablation study on CMU Panoptic Hand Dataset. 32

4.1 Detailed numerical results. 50

5.1 SHG based SiaPose on Panoptic Dataset. 65
5.2 CPM based SiaPose on Panoptic Dataset. 65
5.3 Comparison to state-of-the-art methods. 67
5.4 Domain generalization of our model to MPII+NZSL from Panoptic Dataset. 67

6.1 Numerical results on DexYCB and HUMBI datasets. 84
6.2 Comparison with existing methods on Dex-YCB. 84
6.3 Performance of hand model calibration. 84
6.4 Effectiveness of confidence-valued based attention mechanism. 87
6.5 Evaluating models trained with 3D keypoints instead of mesh supervision on DexYCB and HUMBI datasets. 87

LIST OF ALGORITHMS

		Page
1	Broadcast and Aggregation Matrices Construction	62

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my parents, my sister and Kiki, for their unwavering support and belief in me.

Also, I would like to thank my advisor, Professor Xiaohui Xie, for his invaluable advice, continuous support, and patience during my PhD study. His immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life.

Besides, I would like to express my sincere gratitude to my defense committee members, Professor Xiaohui Xie, Professor Charless C. Fowlkes and Professor Erik B. Sudderth. And also thanks to my advancement to candidacy committee members, Professor Roy Fox and Professor Tingting Nian.

Additionally, I would like to thank my friends, Zhe Wang, Zhanhang Liang, Junjie Shen, Hengjie Wang, Jie Zhu, Dahai Hao, Qiaoqian Hu, Yu Qin, Zhiheng Zuo, Yuting Yang for their support. And my labmates, Hao Tang, Liangjian Chen, Yingxin Cao, Haoyu Ma, Xiangyi Yan, Kun Han, Shanlin Sun, Junayed Naushad, Pooya Khosravi, Thanh-Tung Le, Hasan Celik for their insightful discussions and suggestions.

Finally, I am deeply grateful to Linguang Zhang, who was my mentor during my internship at Facebook Reality Labs, and Verse Zhou, my mentor during my Google internship.

VITA

Deying Kong

EDUCATION

Doctor of Philosophy in Computer Science **2022**
University of California, Irvine *Irvine, California, USA*

Master's Degree in Electrical Engineer **2016**
Huazhong University of Science and Technology *Wuhan, Hubei, China*

Bachelor's Degree in Electrical Engineer **2013**
Huazhong University of Science and Technology *Wuhan, Hubei, China*

RESEARCH EXPERIENCE

Graduate Research Assistant **2021**
University of California, Irvine *Irvine, California, USA*

Soft Engineer Intern **2022**
Google *Mountain View, California, USA*

Research Scientist Intern **2020**
Facebook *Redmond, Washington, USA*

Graduate Research Assistant **2018**
University of California, Irvine *Irvine, California, USA*

Legal Research Intern **2017**
Skyworks Solutions, Inc. *Irvine, California, USA*

TEACHING EXPERIENCE

Teaching Assistant **2017-2022**
University of California, Irvine *Irvine, California, USA*

REFEREED CONFERENCE PUBLICATIONS

• First-authored papers:

Adaptive Graphical Model Network for 2D Handpose Estimation **2019**
British Machine Vision Conference (BMVC'19)

Rotation-invariant Mixed Graphical Model Network for 2D Hand Pose Estimation **2020**
IEEE Winter Conference on Applications of Computer Vision (WACV'20)

Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation **2020**
British Machine Vision Conference (BMVC'20 Oral)

Identity-Aware Hand Mesh Estimation and Personalization from RGB Images **2022**
European Conference on Computer Vision (ECCV'22)

• Co-authored papers:

PPT: token-Pruned Pose Transformer for monocular and multi-view human pose estimation **2022**
European Conference on Computer Vision (ECCV'22)

Topology-Preserving Shape Reconstruction and Registration via Neural Diffeomorphic Flow **2022**
Conference on Computer Vision and Pattern Recognition (CVPR'22)

After-unet: Axial fusion transformer unet for medical image segmentation **2021**
IEEE Winter Conference on Applications of Computer Vision (WACV'21)

Transfusion: Cross-view fusion with transformer for 3d human pose estimation **2021**
IEEE Winter Conference on British Machine Vision Conference (BMVC'21)

Nonparametric structure regularization machine for 2d hand pose estimation **2020**
IEEE Winter Conference on Applications of Computer Vision (WACV'20)

Diffeomorphic Image Registration with Neural Velocity Field **2023**
IEEE Winter Conference on Applications of Computer Vision (WACV'23)

Representation Recovering for Self-Supervised Pretraining on Medical Images **2023**
IEEE Winter Conference on Applications of Computer Vision (WACV'23)

ABSTRACT OF THE DISSERTATION

Deep Learning Models On Hand Pose Estimation and Mesh Reconstruction From RGB Images

By

Deying Kong

Doctor of Philosophy in Computer Science

University of California, Irvine, 2022

Professor Xiaohui Xie, Chair

Estimating and reconstructing human hand pose is a crucial task involved in many real world AI applications, such as human-computer interaction, augmented reality and virtual reality. However, hand pose estimation is challenging because the hand is highly articulated and dexterous, and hand pose estimation suffers severely from self-occlusion. To address the challenges of hand pose estimation from RGB images, several algorithms would be proposed in this thesis. In the first part, the task of 2D hand pose estimation from RBG images would be investigated. We introduce new techniques that combine traditional graphical probabilistic models with deep convolutional neural networks, and use these techniques to incorporate structural constraints of the hand to improve hand pose estimation. Apart from that, a novel graph neural network, spatial information aware GCN, would be proposed, which can efficiently extract spatial information from heatmaps of hand keypoints and propagate them through graph convolution. In the second part, the more challenging problem of 3D hand mesh reconstruction would be tackled. We will introduce an identity-aware hand mesh estimation network and a novel method to perform hand model calibration from RGB images. Extensive experiments have been conducted on multiple large-scale public datasets, demonstrating the state-of-the-art performance.

Chapter 1

Introduction

Human hands play a very important role in our daily life. Understanding hand poses and movements are critical to many applications in VR/AR and human computer interaction. With the advent of deep neural networks, many algorithms have been proposed to solve the problem of hand pose estimation and mesh reconstruction. However, since the hand is highly articulated and often affected by object-occlusion or self-occlusion, the problem still remains challenging, when only single view RGB images are available. In this chapter, we will give a brief definition on the problem of interest and then present the outline of this dissertation.

1.1 Problem Definition

Given a RGB image of the hand, the goal of the 2D hand pose estimation is to find a mapping $f(\cdot)$ from the image to the 2D positions of $K = 21$ keypoints,

$$f : I \in \mathcal{R}^{3\times w\times h} \mapsto J \in \mathcal{R}^{K\times 2}, \tag{1.1}$$

where I is the input image with width w and height h, and J is the 2D hand keypoints. Similarly, the task of 3D hand mesh reconstruction is to derive a mapping $g(\cdot)$ from the image to 3D positions of vertices on the hand surface, as

$$g : I \in \mathcal{R}^{3\times w\times h} \mapsto V \in \mathcal{R}^{N\times 3}, \tag{1.2}$$

where I is the input image with width w and height h, and V is the coordinates of N predefined vertices on the hand mesh surface. The two tasks are illustrated in Fig. 1.1, where the 2D hand keypoints and the projection of the 3D mesh are displayed on top of the input RGB hand image.

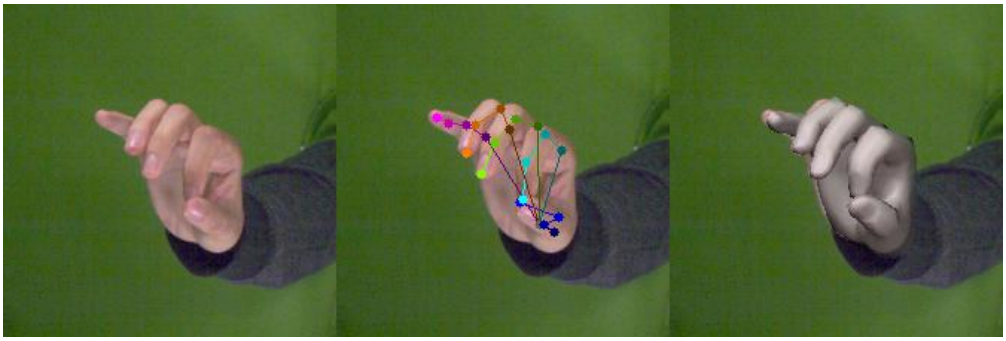


Figure 1.1: Definition of the problem. Left: input RGB image of the hand. Middle: detected 2D keypoints overlaid on the image. Right: 3D hand mesh reprojected onto the image. [141]

1.2 Outline

The thesis is organized as following. Chapter 1 gives a brief introduction of the problem of interest, i.e., human hand pose estimation, and the outline of the dissertation. Then, some backgrounds are given in Chapter 2. Afterwards, 2D hand pose estimation is tackled in Chapter 3, 4 and 5, by utilizing several techniques, e.g., probabilistic graphical models, deep convolutional neural networks, graph neural networks. In the following Chapter 6, the more challenging problem of 3D hand mesh reconstruction is investigated, where a novel

identity-aware network and a new hand model personalization pipeline are proposed. Finally, Chapter 7 concludes the dissertation.

Chapter 2. In this chapter, traditional probabilistic graphical models would be introduced first, followed by recent development of the powerful deep neural networks, including convolutional neural networks and graph neural networks. After that, a popular 3D hand mesh model, MANO, would be discussed.

Chapter 3. To alleviate the limitations of existing methods for 2D hand pose estimation from RGB images, we propose a new architecture named Rotation-invariant Mixed Graphical Model Network (RMGMN). Instead of using a single graphical model, we design a pool of graphical models to accommodate different hand pose categories. The pose categories are obtained via unsupervised learning through the algorithm of K-Means. Additionally, motivated by the fact that a rotation of the input image should not change the graphical model (or only to a rotational angle), we propose to use a rotation net to align the input image before applying the graphical models.

This chapter was previously published in IEEE Winter Conference on Applications of Computer Vision (WACV'20) [54].

Chapter 4. To further exploit the potential power of the graphical model, in this work, we make the graphical model fully adaptive. By "fully adaptive", we mean that the graphical model is conditional on individual input images other than categories as that in Chapter 3. The proposed Adaptive Graphical Model Network (AGMN) consists of two branches of deep convolutional neural networks for calculating unary and pairwise potential functions both conditioned on input images, followed by a graphical model inference module for integrating unary and pairwise potentials. Unlike existing architectures proposed to combine DCNNs with graphical models, our AGMN is novel in that the parameters of its graphical model are conditioned on and fully adaptive to individual input images.

This chapter was previously published in Proceedings of the British Machine Vision Conference (BMVC'19) [53].

Chapter 5. Still focused on the task of 2D hand pose estimation, in this chapter, we resort to another technology, Graph Neural Networks (GNN). GNNs generalize neural networks from applications on regular structures to applications on arbitrary graphs, and have shown success in many application domains such as computer vision, social networks and chemistry. In this chapter, we extend GNNs along two directions: a) allowing features at each node to be represented by 2D spatial confidence maps instead of 1D vectors; and b) proposing an efficient operation to integrate information from neighboring nodes through 2D convolutions with different learnable kernels at each edge. The proposed SIA-GCN can efficiently extract spatial information from 2D maps at each node and propagate them through graph convolution. By associating each edge with a designated convolution kernel, the SIA-GCN could capture different spatial relationships for different pairs of neighboring nodes. When applied on the task of 2D hand pose estimation, the nodes represent the 2D coordinate heatmaps of keypoints and the edges denote the kinetic relationships between keypoints.

This chapter was previously published in Proceedings of the British Machine Vision Conference (BMVC'20) [55].

Chapter 6. This chapter handles the more challenging task of reconstructing 3D hand mesh from RGB input images. This task has attracted increasing amount of attention due to its enormous potential applications in the field of AR/VR. Most state-of-the-art methods attempt to tackle this task in an anonymous manner. Specifically, the identity of the subject is ignored even though it is practically available in real applications where the user is unchanged in a continuous recording session. In this chapter, an identity-aware hand mesh estimation model is proposed, which can incorporate the identity information represented by the intrinsic shape parameters of the subject. The importance of the identity information is demonstrated by comparing the proposed identity-aware model to a baseline which treats

subject anonymously. Furthermore, to handle the use case where the test subject is unseen, a novel personalization pipeline is proposed to calibrate the intrinsic shape parameters using only a few unlabeled RGB images of the subject.

This chapter was previously published in European Conference on Computer Vision (ECCV'22) [56].

Chapter 7. This chapter concludes the dissertation.

Chapter 2

Background

2.1 Probabilistic Graphical Models

Probabilistic graphical models are a class of statistical models which combine the rigour of a probabilistic approach with the intuitive representation of relationships given by graphs [93]. In general, they are composed by two parts:

- A set $\mathcal{X} = \{X_1, X_2, \cdots, X_p\}$ of random variables describing the quantities of interest.
- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in which each node $v \in \mathcal{V}$ is associated with one of the random variables in $\mathcal{X}$. Edges $e \in \mathcal{E}$ are used to express the dependence structure of the data [93].

Many traditional algorithms on human/hand pose estimation/tracking have been developed by using the graphical models, several of which are introduced as follows.

In [103], a probabilistic method is developed for visual tracking of a three-dimensional geometric hand model from monocular image sequences. Using a graphical model of hand

kinematics, the hand's motion is tracked using the nonparametric belief propagation algorithm. In [24], a deformable template is proposed to detect and localize shapes in grayscale images. The template is formulated as a Bayesian graphical model of a two-dimensional shape contour, and it is matched to the image using a variant of the belief propagation algorithm used for inference on graphical models. In [95], the 3D human body is represented as a graphical model in which the relationships between the body parts are represented by conditional probability distributions. Then the pose estimation problem is formulated as one of probabilistic inference over a graphical model where the random variables correspond to the individual limb parameters. The famous framework of pictorial structures has been proposed in [29], whose key idea is to represent an object by a collection of parts arranged in a deformable configuration. The appearance of each part is modeled separately, and the deformable configuration is represented by spring-like connections between pairs of parts.

2.2 Deep Convolutional Neural Networks

Deep Convolutional Neural Networks (CNNs) have witnessed many successes in a range of fields, e.g., image classification, image segmentation, object detection, pose estimation, in recent years [98, 41, 57, 91, 87, 40, 89, 114**?**]. In this section, we will introduce two popular deep CNNs that are used in the field of 2D pose estimation, convolutional pose machine [124] and stacked hour glass [75]. They are initially proposed for human pose estimation, but can be also applied for the task of 2D hand pose estimation.

2.2.1 Convolutional Pose Machine

Convolutional Pose Machine (CPM) provides a sequential prediction framework for learning rich implicit spatial models [124]. As shown in Fig. 2.1, the CPM refines heatmaps sequen-

tially, with increasing receptive fields in later stages. In the first stage, the network only works on the image evidence, while in the following stages, the network operates on both image evidence and the heatmap output from the previous stage. By doing so, the CPM has the ability to implicitly model long-range dependencies between variables in structured prediction tasks, which enables it achieve then state-of-the-art performance on 2D human pose estimation.

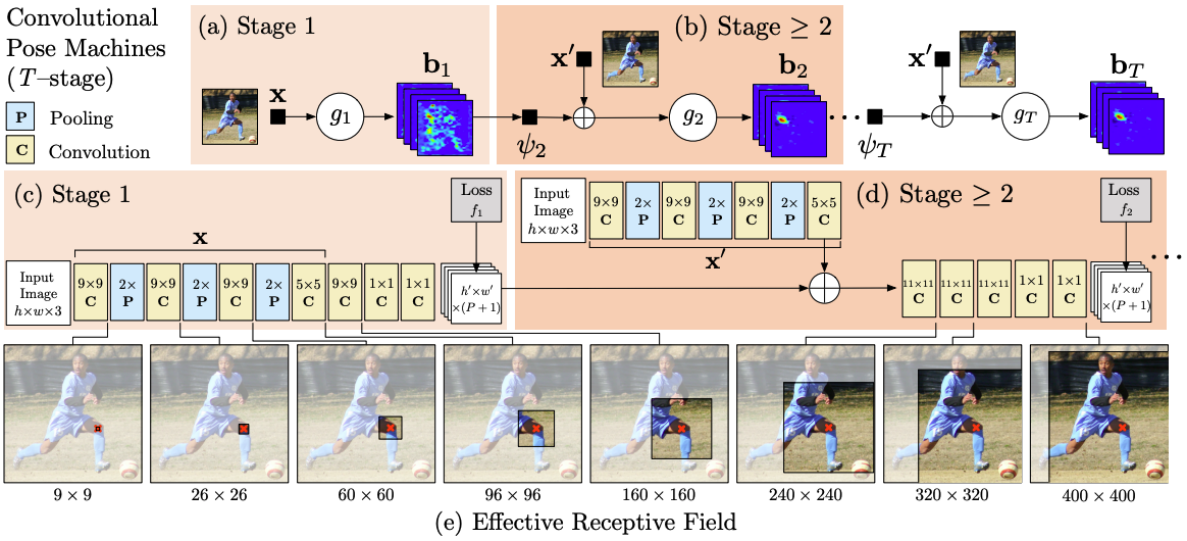


Figure 2.1: Architecture and receptive fields of convolutional pose machine in [124].

2.2.2 Stacked Hourglass

Apart from the CPM, there is another popular architecture, stacked hourglass, that has been proposed to tackle 2D pose estimation problem [75]. The motivation is that to understand the pose, not only is local information necessary, but the knowledge of global information is also essential. In the stacked hourglass networks, features are processed across all scales and consolidated to best capture the various spatial relationships associated with the human body, as shown in Fig. 2.2.

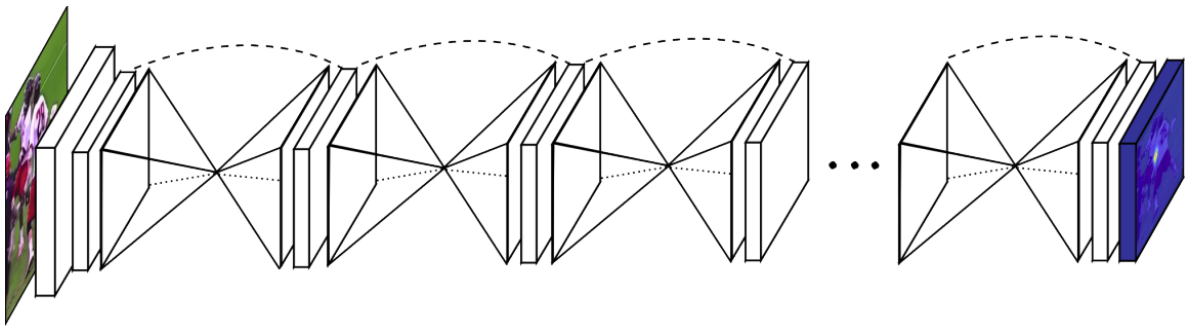


Figure 2.2: The staked hourglass network for pose estimation consists of multiple modules which allow for repeated bottom-up, top-down inference [75].

2.3 Integrating CNNs and Graphical Models

There is also a branch of works trying to combine deep CNNs and graphical models for pose estimation [107, 19, 132], which are trained independently or end-to-end via the coordination of back propagation and message-passing. The combination of DCNNs and GM has been studied in several scenarios, i.e., human pose estimation in a vedio[99], multi-person pose estimation [82, 43], multi-person pose tracking [42, 44]. Although these methods are mainly proposed for the task of human body pose estimation, they can also be applied to hand pose estimation.

The work in [107] is a pioneer in integrating deep neural networks and graphical models. In [107], a new hybrid architecture has been proposed, which consists of a deep convolutional network and a Markov random field. It has been shown that this architecture can be successfully applied to the challenging problem of articulated human pose estimation in monocular images. The architecture can exploit structural domain constraints such as geometric relationships between body joint locations. Fig. 2.3 illustrates how the message passing works between body joints and how the structual constraints are ultized.

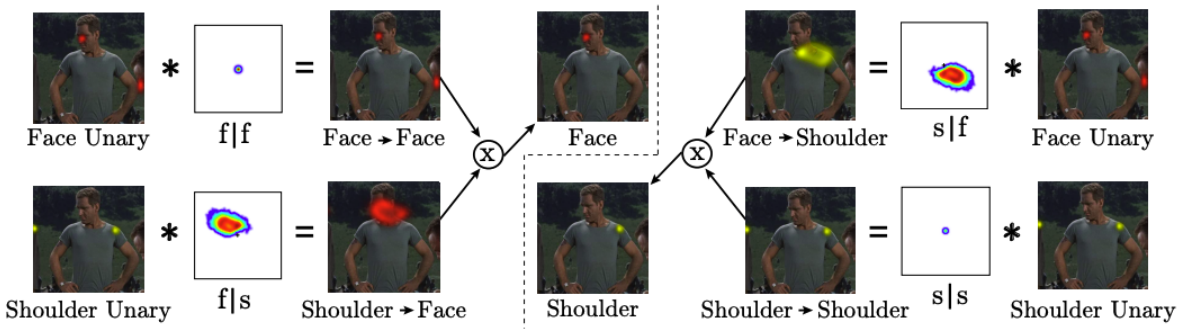


Figure 2.3: Didactic example of message passing between the face and shoulder joints [107].

2.4 Graph Neural Networks

Convolutional neural networks can only operate on regular Euclidean data, e.g, images (2D grids) and texts (1D sequences) while these data structures can be regarded as instances of graphs. Graph Neural Networks (GNNs) have been proposed to operate on unregular graphs with applications in the fields of physics, chemistry, social networks and so on [137, 52, 94, 113, 27]. Among the many popular types of GNNs, we would introduce one spectral-based GNN, Graph Convolutional Network (GCN) which is proposed by Kipf and Welling [52] which has enjoyed great success on a variety of applications since its advent.

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$, adjacency matrix $A \in \mathbb{R}^{N \times N}$, and a degree matrix $D \in \mathbb{R}^{N \times N}$ with $D_{ii} = \sum_j A_{ij}$, the layer-wise propagation rule of GCN is characterized by the following equation

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \tag{2.1}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with self-connections [52]. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(l)} \in \mathbb{R}^{N \times M}$ is the matrix of activations in the l^{th} layer, or input feature matrix of the l^{th} layer. The parameter $W^{(l)}$ is the trainable weight matrix of layer l.

2.5 Generative Hand Models

Many 3D hand models have been proposed in the past decades. For example, [76, 85, 88] are among the methods that approximate the hand with shape primitives, by doing this, fast evaluation of distances has been achieved. In [92], shape primitives are voxelized and signed distance functions for local coordinated are computed. Another line of research resort to triangulated mesh with linear blend skinning, which gives more realistic hand [92, 109, 90]. Recently, with the popularity of neural fields [127], the LISA hand model is proposed in [23], which is defined by an articulated implicit representation learned from multi-view RGB videos annotated with coarse 3D hand poses.

In this subsection, we would mainly introduce the widely used triangular mesh based MANO hand model [90], which is extended from the 3D human model SMPL [65]. MANO is learned from around 1000 high-resolution 3D scans of hands of 31 subjects in a wide variety of hand poses. The model is realistic, low-dimensional, captures non-rigid shape changes with pose, is compatible with standard graphics packages, and can fit any human hand [90].

The MANO model factorizes the hand mesh into two groups of parameters: the shape parameters and the pose parameters. The shape parameters control the intrinsic shape of the hand, e.g., size of the hand, thickness of the fingers, length of the bones, etc. The pose parameters represent the hand pose, i.e., how the hand joints are transformed, which subsequently deforms the hand mesh. Mathematically, the model is defined as below:

$$\mathcal{M}(\beta, \theta) = W(T_P(\beta, \theta), J(\beta), \theta, \mathcal{W}) \tag{2.2}$$

where a skinning function W is applied to an articulated mesh with shape T_P, joint locations J, pose parameter θ, shape parameter β, and blend weights $\mathcal{W}$ [90]. Fig. 2.4 and Fig. 2.5 shows the PCA shape and pose spaces of the MANO model respectively.

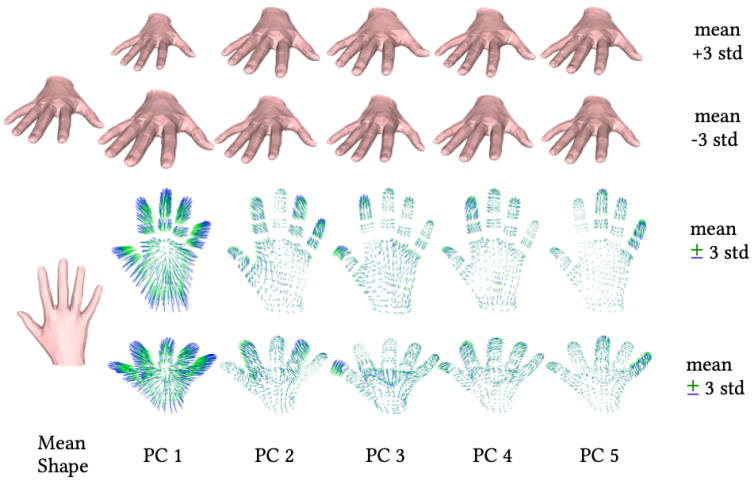


Figure 2.4: PCA shape space of MANO model [90].

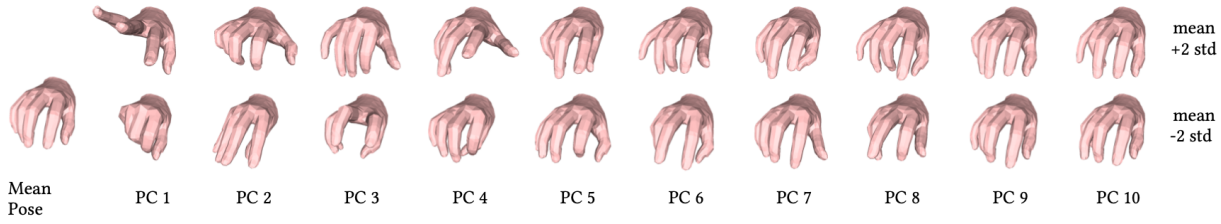


Figure 2.5: PCA pose space of MANO model [90].

Chapter 3

Rotation-invariant Mixed Graphical Model Network

In this chapter, we propose a new architecture named Rotation-invariant Mixed Graphical Model Network (R-MGMN) to solve the problem of 2D hand pose estimation from a monocular RGB image. By integrating a rotation net, the R-MGMN is invariant to rotations of the hand in the image. It also has a pool of graphical models, from which a combination of graphical models could be selected, conditioning on the input image. Belief propagation is performed on each graphical model separately, generating a set of marginal distributions, which are taken as the confidence maps of hand keypoint positions. Final confidence maps are obtained by aggregating these confidence maps together. We evaluate the R-MGMN on two public hand pose datasets. Experiment results show our model outperforms the state-of-the-art algorithm which is widely used in 2D hand pose estimation by a noticeable margin.

3.1 Introduction

Hands play a central role in almost all daily activities of human beings. Understanding hand pose is an essential task for many AI applications, such as gesture recognition, human-computer interaction [102], and augmented/virtual reality [59, 83]. The task of estimating hand pose has been investigated for decades, however, it still remains challenging due to the complicated articulation, high dexterity and severe self-occlusion.

To address these problems, one possible way is to resort to multi-view camera systems [47, 77, 96]. However, such systems are expensive and not practical. Meanwhile, with the popularization of low-cost depth sensors in recent years, a large number of RGB-D based approaches have been proposed for 3D hand pose estimation [3, 31, 33, 74, 111, 112, 134]. Nonetheless, RGB cameras are still the most popular and easily accessible devices. Researchers have started performing 3D hand pose estimation directly from RGB images [6, 7, 32, 45, 73, 78, 101, 139]. Many proposed approaches involve a two stage architecture, i.e., first performing 2D hand pose estimation and then lifting the estimated pose from 2D to 3D [6, 73, 78, 139], which makes 2D hand pose estimation itself still an important task. In this paper, we investigate the problem of 2D hand pose estimation from a monocular RGB image.

The research field of 2D hand pose estimation is related closely to that of human pose estimation. Spurred by developments in deep learning and large datasets publicly available [74, 96], deep convolutional neural network (DCNN)-based algorithms have make this field advance significantly. Convolutional Pose Machines (CPM) [124] is one of the most popular and well known algorithms for human pose estimation, and it has been widely applied in 2D hand pose estimation [96] yielding the state of the art performance.

Although the deep convolutional neural networks have the power to learn very good feature representations, they could only learn spatial relationships among joints or keypoints implicitly, which often results in joint inconsistency [49, 99]. To model the correlation among joints

explicitly, several studies investigate the combination of Graphical Model (GM) and DCNN in pose estimation. In most of the studies [99, 107, 132], a self-independent GM is imposed on top of the score maps regressed by DCNN. The parameters of the GM are learned during end-to-end training, then these parameters are fixed during prediction. However, pose can be varied in different scene, a fixed GM is unable to model diverse pose. This shortage could be even worse in hand pose estimation. In [19], image-dependent pairwise potentials are introduced, however, the model does not support end-to-end training and the pairwise potential is restrained to quadratic function.

In this paper, we propose a novel architecture for 2D hand pose estimation from monocular RGB image, namely, the Rotation-invariant Mixed Graphical Model Network (R-MGMN). We argue that different hand shapes should be associated with different spatial relationships among hand keypoints, resulting to graphical models with different parameters. Also, a powerful graphical model should have the ability to capture the same shape of the hands when viewed from a different angle, i.e., the graphical model should be rotation-invariant.

The proposed R-MGMN consists of four parts, i.e., a rotation net, a soft classifier and a pool which contains several different graphical models. The rotation net is inspired by the Spatial Transformer Networks [46]. The goal of the rotation net is to rotate the input image such that the hand would be in a canonical direction. Then, the soft classifier outputs a soft class assignment vector (which sums up to 1), representing the belief on possible shapes of the hand. Meanwhile, the unary branch generates heatmaps which would be fed into the graphical models as unary functions. After that, inference is performed via message passing on each graphical model separately. The inferred marginals are aggregated by weighted averaging, using the soft assignment vector. This procedure could be viewed as a soft selection of graphical models. The final scoremap is obtained by rotating the aggregated marginal backwards to align with the original coordinate of the input image.

We demonstrate the performance of the R-MGMN on two public datasets, the CMU Panoptic

Dataset [96] and the Large-scale 3D Multiview Hand Pose Dataset [35]. Our approach outperforms the popularly used algorithm CPM by a noticeable margin on both datasets. Qualitative results indicate our model could alleviate geometric inconsistency among hand keypoints even when severe occlusion exists.

The main contributions of this paper are summarized as follows:

- We propose a new model named R-MGMN which combines graphical model and deep convolutional neural network efficiently.
- Instead of only having one graphical model, the proposed R-MGMN has several independent graphical models which can be selected softly, depending on input image. And it could be trained end-to-end.
- Our R-MGMN could alleviate the spatial inconsistency among predicted hand keypoints greatly and outperform the popularly used CPM algorithm by a notable margin.

3.2 Related Work

Human pose estimation from single RGB image. Studies on hand pose estimation have been benefiting from that on human pose estimation for a long time. Since DeepPose [108] pioneered the application of DCNN in pose estimation, DCNN-based algorithms have dominated the field [118]. For example, the network proposed by Sun et al. [104] has achieved the state-of-the-art score in many human pose estimation datasets [1, 63]. Early DCNN-based algorithms try to regress the 2D coordinates of keypoints [115, 12]. Later algorithms estimate keypoint heatmaps [124, 75, 22], which usually achieve better performance. The main body of DCNN mainly adopts the high-to-low and low-to-high framework, optionally augmented with multi-scale fusion and intermediate supervision. However, the structure information among the body joints captured by DCNN is implicit. Some approaches try to

learn extra information besides heatmaps of joint position to provide structural constraints, i.e. compound heatmaps [49] and offset fields [79]. Nonetheless, these methods still could not fully exploit structural information.

Hand pose estimation. Recently, most studies of hand pose focus on 3D hand pose estimation, which is much more challenging than body pose estimation, due to self-occlusion, dexterity and articulation of the hand. The mainstream approaches usually resort to either multi-view camera system [47, 77, 96] or depth data [3, 33, 111, 112, 134]. Nevertheless, There is also a rich literature on 3D hand pose and reconstruction from single color image using deep neural networks [6, 7, 32, 45, 73, 78, 101, 139] . Some studies fit their 3D hand models from the estimated 2D joint locations [6, 73, 78, 139]. Thus the accuracy of 2D hand pose estimation has a great impact on the performance of 3D hand pose.

Among a variety of DCNN-based models, CPM [124] is commonly used in 2D hand pose estimation [96, 116, 139]. This architecture estimates the score maps via intermediate supervision and the most likely location is selected as the maximum confidence of the corresponding position in the confidence maps. In this paper, we choose CPM as the baseline for comparison with our proposed model.

Graphical model in pose estimation. Graphical model has also been exploited in solving human pose estimation tasks. By using GM, spatial constraints among body parts can be modeled explicitly.

Recently, there is also a trend to combine DCNN and GM for pose estimation [107, 19, 99, 132]. The GM and the backbone DCNN are trained either independently or end-to-end via the combination of back propagation and message-passing. However, studies in this field usually apply a GM with fixed parameters, which limits its ability to model a variety of pose, especially in hand pose estimation. The most recent work in [53] proposes to generate adaptive GM parameters conditioning on individual input images.

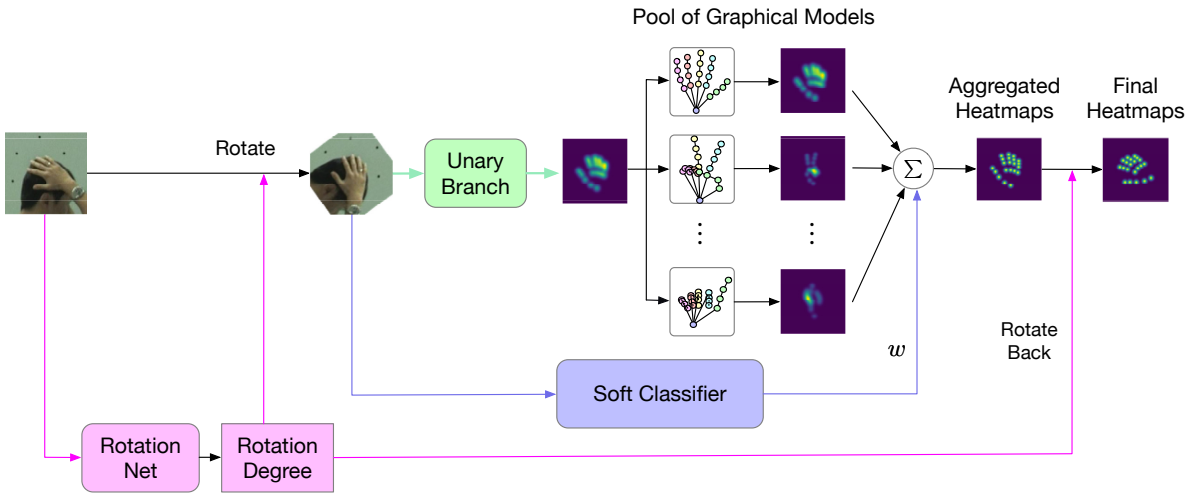


Figure 3.1: Pipeline overview of the proposed Rotation Mixture Graphical Model Network (R-MGMN).

3.3 Methodology

3.3.1 Basic pipeline

The proposed Rotation-invariant Mixture Graphical Model Network (R-MGMN) mainly consists of four components, i.e., the rotation net, the soft classifier, the unary branch and the pool of graphical models, as shown in Fig. 3.1. The pipeline of the MGMN is given as follows.

- The rotation net regresses a rotation degree from the input image.
- Then, using the obtained rotation degree, the image is rotated such that the hand in the image would be in a canonical direction (e.g. the hand is upright).
- After that two parallel branches follow.

  Branch 1:

  – A deep neural network referred to as unary branch is applied onto the rotated image. The output of the unary branch is a set of 2D heatmaps which represent

the confidence of the hand keypoint positions.

  - As unary potential functions, these 2D heatmaps are fed into the pool of graphical models. Each graphical model performs inference separately. Then, the pool outputs several sets of marginal probabilities of the keypoint positions.

  Branch 2:

  - The parallel branch contains a soft classifier which outputs a weight vector whose entries sum up to one.

- Aggregated heatmaps are obtained as the weighted average of the marginal probabilities, using the weight vector.
- Rotate heatmaps backwards according to previous rotation degree.

3.3.2 Model

Our R-MGMN could be broken down into two parts:

- The rotation part which controls the rotation of the image and the backward rotation of the heatmaps.
- The MGMN, which performs handpose estimation on the rotated images.

Image rotation

The rotation angle α is regressed from the rotation net $\mathcal{RT}$ as

$$\alpha = \mathcal{RT}(I; \theta_{rt}), \tag{3.1}$$

where θ_{rt} is the set of parameters of the rotation net, I is the input image. Then the rotated image is given by

$$I_{rt} = f_{rt}(I, \alpha), \tag{3.2}$$

where f_{rt} is the rotation function.

MGMN

Given the rotated image I_{rt}, the handpose estimation problem could be formulated by using a graph and it could be solved via probabilistic tools.

Let $\mathcal{V} = \{v_1, v_2, \cdots, v_K\}$ denote the set of all the hand keypoints, each of which is associated with a random variable $x_i \in \mathbb{R}^2$ representing its 2D position in image I_{rt}. And let $\mathcal{E}$ represent the set of pairwise relationships among the keypoints in $\mathcal{V}$, to be more specific, $(i, j) \in \mathcal{E}$ if and only if v_i and v_j $(i < j)$ are neighbours. Then we could define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ being its vertices and $\mathcal{E}$ being the edges. A basic probabilistic model of the handpose task could be formulated by the following equation.

$$p^{\texttt{basic}}(X|I_{rt}) = \prod_{v_i \in V} \phi(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi(x_j, x_k|I_{rt}), \tag{3.3}$$

where $\phi(x_i) \in \mathbb{R}$ is usually called the unary function, $\psi(x_j, x_k) \in \mathbb{R}$ is the pairwise function and X denotes the positions of all hand keypoints, i.e., $X = (x_1, x_2, \cdots, x_K)$.

The naive model in Eq. (3.3) could be generalized to a mixed graphical model as

$$p(X|I_{rt}) = \sum_{l=1}^{L} w_l \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}), \tag{3.4}$$

where L graphical models are aggregated together, w_l is the weight corresponding to the l-th graphical model.

Our proposed MGMN is obtained when the same unary function is shared for all L graphical models, i.e.,

$$\phi_l(x_i|I) = \eta(x_i|I),\ l = 1, 2, \cdots, L \tag{3.5}$$

where $\eta(x_i|I)$ is the output of the unary branch $\mathcal{U}$ with parameters θ_u in Fig. 3.1,

$$\eta(x_i|I) = \mathcal{U}(I, \theta_u). \tag{3.6}$$

The marginal probability $p(x_i|I_{rt})$ could be calculated by summing up the marginal probabilities $p_l(x_i|I_{rt})$ of each individual graphical models, as validated by the following equation,

$$p(x_i|I_{rt}) = \sum_{\sim x_i} p(X|I_{rt}) \tag{3.7}$$

$$= \sum_{\sim x_i} \sum_{l=1}^{L} w_l \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}) \tag{3.8}$$

$$= \sum_{l=1}^{L} w_l \sum_{\sim x_i} \prod_{v_i \in V} \phi_l(x_i|I_{rt}) \prod_{(j,k)\in\mathcal{E}} \psi_l(x_j, x_k|I_{rt}) \tag{3.9}$$

$$= \sum_{l=1}^{L} w_l\ p_l(x_i|I_{rt}), \tag{3.10}$$

where $\sum_{\sim x_i}$ means to summing over all $x_k, k = 1, 2, \cdots, K$ except x_i. The marginal $p_l(x_i|I_{rt})$ of each graphical model could be calculated exactly or approximately using message passing efficiently.

The marginal $p(x_i|I_{rt})$ could be taken as a confidence of the joint v_i being located at the specific position. For each keypoint v_i, a confidence map or score map S_i, which is a 2D matrix, could be constructed by assigning the (m, n)-th entry of S_i to be

$$S_i[m, n] = p(x_i = (m, n)|I_{rt}), \tag{3.11}$$

where (m, n) is the 2D coordinate.

Inverse Rotation of Confidence Maps

The final confidence map of the keypoint v_i's position is given by

$$S_i^{\mathrm{F}} = f_{rt}(S_i, -\alpha), \tag{3.12}$$

where α is given by the rotation net as in Eq. (3.1).

The predicted position of keypoint v_i is obtained by maximizing the confidence map, as

$$x_i^* = (m^*, n^*) = \underset{m,n}{\operatorname{argmax}} \; S_i^{\mathrm{F}}[m, n]. \tag{3.13}$$

3.3.3 Detailed structure of the R-MGMN

In this subsection, we would describe the detailed structure of each component of the R-MGMN.

Rotation Net

The rotation net consists of a ResNet18 and two additional layers to regress the rotation degree α, as shown in Fig. 3.2.

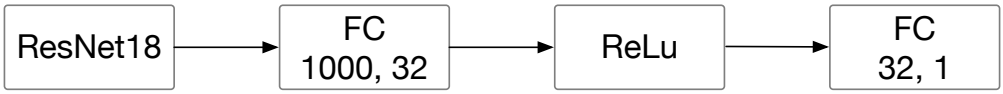


Figure 3.2: Configuration of the rotation net.

The output of ResNet18 is a 1000-dimensional vector, then it's fed into two fully connected (FC) layers with a ReLu function in between. Finally, a scalar representing the rotation degree is obtained.

Unary Branch

The Convolutional Pose Machine (CPM) is adopted as our unary branch. To be more specific, we follow the same architecture used in [96]. The convolutional stages of a pre-initialized VGG-19 network up to conv4_4 are utilized as a feature extractor. Then, six cascaded stages are deployed to regress the confidence maps repeatedly. Moreover, as in [9], convolutions of kernel size 7 are replaced with 3 layers of convolutions of kernel 3 which are concatenated at their end.

Soft Classifier

For the soft classifier we adopt the ResNet-152 followed by a softmax layer as in Fig. 3.3. The output dimension of the ResNet-152 is set to be 20, which means we would like to expect there are 20 clusters among the hands.

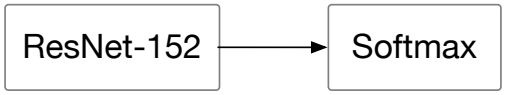


Figure 3.3: Configuration of the soft classifier.

Pool of Graphical Models

There are $L = 20$ tree-structured graphical models integrated in the pool of graphical models. Each of the graphical model shares the same structure, but every single graphical model is

associated with a different set of parameters. Marginal probabilities are inferred on each individual graphical model, and then aggregated via a weight vector which comes from the soft classifier.

Belief propagation. Sum-product message passing is a well known algorithm for performing inference on graphical models. It could calculate marginals of the random variables efficiently. During the inference, vertices on the graph receive messages from and send messages to their neighbors iteratively, as in the following equation,

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j)\phi_i(x_i) \prod_{k \in \mathtt{Nbd}(i) \setminus j} m_{ki}(x_i)\,, \tag{3.14}$$

where $m_{ij} \in \mathbb{R}$ is the message sent from vertex v_i to vertex v_j, which is the belief from the vertex v_i on the position of the j-th keypoint.

The message passing process in the above equation would be performed several iterations until convergence or satisfaction of some other stop criteria. The estimated marginal distribution $\hat{p}_i(x_i)$ is given by

$$\hat{p}_i(x_i) \propto \phi_i(x_i) \prod_{k \in \mathtt{Nbd}(i)} m_{ki}(x_i) \tag{3.15}$$

$$= \frac{1}{Z'}\phi_i(x_i) \prod_{k \in \mathtt{Nbd}(i)} m_{ki}(x_i), \tag{3.16}$$

where Z' is a normalization term such that the probabilities sum up to 1.

Message passing on tree-structured graphs. When the graph is tree-structured, the estimated marginal equals the exact marginal. In our R-MGMN, tree-structured models are utilized, as illustrated in Fig. 4.3. Each branch consisting of four same-colored circles corresponds to a single finger.

By using a tree-structure, exact marginals could be inferred very efficiently by only two

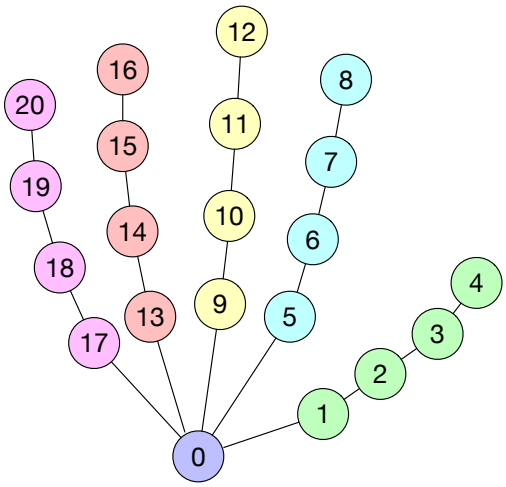


Figure 3.4: Tree-structured graphical model for hand keypoints.

passes of message passing. In the first step, starting from the leaf nodes, variables pass messages sequentially towards the root node. Then in the second step, messages are passed sequentially towards the leaf nodes, beginning at the root node.

Message passing as 2D convolution. For each iteration, the message $m_{ij}(x_j)$ in Eq. (3.14) could be rewritten as

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j) h_i(x_i), \tag{3.17}$$

where

$$h_i(x_i) \triangleq \phi_i(x_i) \prod_{k \in \mathtt{Nbd}(i) \setminus j} m_{ki}(x_i). \tag{3.18}$$

If the pairwise potential function $\varphi_{i,j}(x_i, x_j)$ only depends on the relative position between the two neighboring keypoints, i.e.,

$$\varphi_{i,j}(x_i, x_j) = \gamma_{i,j}(x_i - x_j)\,. \tag{3.19}$$

By compacting $m_{i,j}(\cdot)$ and $h_i(\cdot)$ into 2D matrices M_{ij} and H_i (this is reasonable since x_i

corresponds to a 2D location), Eq. (4.16) is transformed to

$$M_{ij} = \Gamma^{i,j} \circledast H_i \,, \tag{3.20}$$

where $\Gamma^{i,j}$ is a 2D matrix encoding the pairwise potential function $\gamma_{i,j}(x_i - x_j)$, and the notation $\circledast$ denotes convolution.

Thus, the set of parameters for each graphical model is given by

$$\Theta^{\mathtt{gm}} = \{\Gamma^{i,j} | (i,j) \in \mathcal{E} \text{ or } (j,i) \in \mathcal{E}\} \,. \tag{3.21}$$

The whole set of parameters of the pool of graphical models are

$$\Theta^{\mathtt{GM}} = \{\Theta_l^{\mathtt{gm}} \mid l = 1, 2, \cdots, L\} \,, \tag{3.22}$$

where $\Theta_{l_1}^{\mathtt{gm}}$ is independent of $\Theta_{l_2}^{\mathtt{gm}}$ for $l_1 \neq l_2$.

3.4 Learning

Since our R-MGMN contains several components, we follow a step-by-step training procedure. First, the rotation net is trained. Then, while keeping the rotation net fixed, we train the unary branch and the soft classifier separately. After that, the parameters of graphical models are learned while keeping other parts frozen. Finally, the whole R-MGMN is jointly trained. More details are given as following.

3.4.1 Train Rotation Net

The rotation net is trained alone in the first phase of the training. The aim of the rotation net is to rotate the input image such that the hand in the resulted image is upwards, i.e., the directional line connecting the 1-st keypoint and the 10-th keypoint is pointing upwards as illustrated in Fig. 3.5.

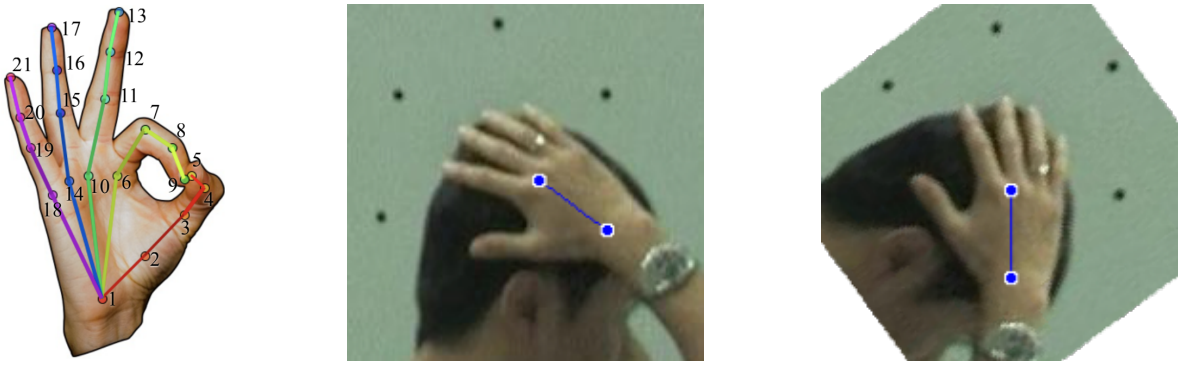


Figure 3.5: Illustration of the rotation. Left image courtesy to [96] .

Almost no public dataset provides the ground truth rotation degree directly, however, it could be obtained easily given the ground truth positions of hand keypoints. The ground truth rotation degree α^* could be derived by calculating the directional angle between the vector v_1 and v_2, where

$$v_1 = x_{10} - x_1, \tag{3.23}$$

with x_{10} and x_1 representing the positions of the keypoints, and v_2 is the unit vector whose direction is vertically upwards.

During training, squared error is used for the loss function, which is

$$L^{\mathtt{rn}} = (\alpha - \alpha^*)^2\,, \tag{3.24}$$

where α is the regressed rotation degree from the rotation net.

3.4.2 Train Unary Branch

The unary branch is trained with the help of the rotation net, while the rotation net is fixed during this training phase. The unary branch is actually the convolutional pose machine, which produces and refines the confidence maps repeatedly. Rotated image is fed into the unary branch, which outputs a set of confidence maps. These confidence maps are then rotated back so as to be aligned with the original coordinate of the input image before the rotation net.

Denote $S_k^t \in \mathbb{R}^{h_u \times w_u}$ as the aligned output confidence map of the k-th keypoint at the t-th stage of the unary branch, the loss function used in this training phase is designed as

$$L^{\texttt{unary}} = \sum_{t=1}^{T} \sum_{k=1}^{21} ||S_k^t - S_k^*||_F^2 \,, \tag{3.25}$$

where T is the number of stages in the unary branch, $S_k^* \in \mathbb{R}^{h_u \times w_u}$ is the ground truth confidence map of the k-th keypoint, and $||\cdot||_F$ represents the Frobenius norm. The ground truth S_k^* is obtained by by putting a Gaussian peak at the keypoint's ground truth location.

3.4.3 Train Soft Classifier

Again, there is no ground truth class label for the classification subtask. Thus, we resort to unsupervised learning, especially the K-means clustering algorithm. To be fair, only training dataset is utilized in this phase.

Given the pretrained rotation net in the first phase, we rotate the images and the keypoints' position labels according to the estimated rotation degrees. Then, the K-means algorithm is applied on the rotated images. The feature vector used in K-means is obtained by concatenating the relative positions of neighbouring keypoints. The number of the clusters is set to

be 20.

The training set is further split into 70/30, on which the soft classifier is trained on. Standard cross entropy is used for the loss function.

3.4.4 Train Graphical Model Parameters

Keeping all the other parts fixed, in this phase, we only train the parameters of the graphical models, with the whole R-MGMN. The loss function is

$$L^{\texttt{GM}} = \sum_{k=1}^{21} ||\tilde{S}_k - \tilde{S}_k^*||_F^2, \tag{3.26}$$

where $\tilde{S}_k \in \mathbb{R}^{h_o \times w_o}$ is the k-th channel of the output of the R-MGMN. Since the confidence map $\tilde{S}_k$ is actually a normalized probability distribution, the ground truth $\tilde{S}_k^* \in \mathbb{R}^{h_o \times w_o}$ is also normalized ($\tilde{S}_k^*$ is the normalized version of S_k^* from Eq. (3.25)).

3.4.5 Jointly Train All the Parameters

For the last phase, we use the same loss function as that in training the graphical model paramters,

$$L^{\texttt{Joint}} = L^{\texttt{GM}}. \tag{3.27}$$

3.5 Experiments

We verify our approach on two public handpose datasets, i.e., the CMU Panoptic Hand Dataset (CMU Panoptic) [96] and the Large-scale Multiview 3D Hand Pose Dataset (Large-scale 3D) [30]. A comprehensive analysis of the proposed model is also carried out.

3.5.1 Experimental settings

Datasets.

The CMU Panoptic dataset contains 14817 annotations of hand images while the Large-scale 3D dataset contains 82760 anotations in total. The Large-scale 3D dataset provides a simple interface to generate 2D labels from the 3D labels which come with the dataset. For both datasets, we split them into training set (70%), validation set (15%) and test set (15%). Since we focus on handpose estimation in this paper, we crop image patches of annotated hands off the original images, thus leaving out the task of hand detection. A square bounding box which is 2.2 times the size of the hand is used during the cropping.

Evaluation metric.

Probability of Correct Keypoint (PCK) [96] is a popular metric, which is defined as the probability that a predicted keypoint is within a distance threshold σ of its true location. In this paper, we use normalized threshold σ with respect to the size of hand bounding box, and mean PCK (mPCK) with $\sigma = \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$.

Implementation Details.

All input images are resized to 368×368, then scaled to [0,1] and further normalized using mean of (0.485, 0.456, 0.406) and standard derivation of (0.229, 0.224, 0.225). Batch size is set to 32 for all training phases. Adam is used as the optimizer, and the initial learning rate is set to be lr =1e-4 for each training phase. The rotation net is only trained for 6 epochs at the fist training phase. With best models being selected basing on the performance of the validation set, the unary branch and soft classifier are both trained for 100 epochs, after which the parameters of graphical models are trained for 40 epochs, and finally the whole network are trained end-to-end for 150 epochs.

3.5.2 Results

The PCK performance of our proposed model on two public datasets, i.e., the CMU Panoptic dataset and the Large-scale 3D dataset, are shown in Fig. 3.6. It is seen that our model outperforms the CPM consistently on both datasets. Detailed numerical results are given in Table 4.1.

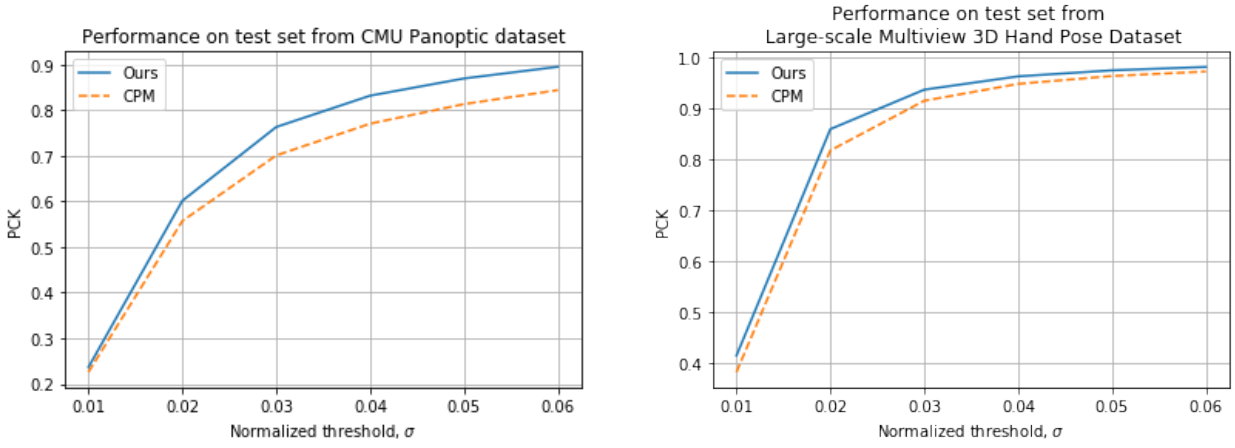


Figure 3.6: PCK performance on two public datasets.

On CMU Panoptic dataset, our model achieves a significant PCK improvement comparing

Threshold of PCK, σ	0.01	0.02	0.03	0.04	0.05	0.06	mPCK
CMU Panoptic Hand Dataset							
CPM Baseline (%)	22.60	55.69	70.06	77.01	81.30	84.36	65.17
Ours	23.67	60.12	76.28	83.14	86.91	89.47	69.93
Improvement	1.07	4.43	**6.22**	**6.13**	**5.61**	**5.11**	**4.76**
Large-scale Multiview 3D Hand Pose Dataset							
CPM Baseline (%)	38.27	81.78	91.54	94.84	96.39	97.27	83.35
Ours	41.51	85.97	93.71	96.33	97.51	98.17	85.53
Improvement	**3.24**	**4.19**	2.17	1.49	1.12	0.90	2.18

Table 3.1: Detailed numerical results of PCK performance.

Threshold of PCK, σ	0.01	0.02	0.03	0.04	0.05	0.06	mPCK	improvement
CPM Baseline (%)	22.60	55.69	70.06	77.01	81.30	84.36	65.17	-
CPM + Single GM	22.58	55.78	70.14	77.05	81.34	84.41	65.21	0.04
CPM + Mixture of GMs	23.39	57.53	71.95	78.49	82.28	85.02	66.44	1.27
Rotaion + CPM [1]	22.70	57.91	72.95	79.94	83.90	86.71	67.35	2.18
Rotaion + CPM [2]	21.97	57.59	74.53	81.98	86.21	88.83	68.52	3.35
R-MGMN	23.67	60.12	76.28	83.14	86.91	89.47	69.93	4.76

Table 3.2: Numerical results for ablation study on CMU Panoptic Hand Dataset.

to CPM. An absolute improvement of 6.22 percent is observed at threshold $\sigma = 0.03$. In average, the mPCK is improved by 4.76 percent. The experiment result on Large-scale 3D dataset also validates the advantage of our model. At threshold of $\sigma = 0.02$, there is a 4.19 percent improvement in PCK.

The reason why the improvement on Large-scale 3D dataset is not as much as that on the CMU Panoptic dataset, probably lies in the fact that annotation settings of these two datasets are slightly different. In Large-scale 3D dataset, the center of the palm is considered as the root keypoint instead of the wrist. This would cause the reference vector v_1 in Eq. (3.23) to be relatively short, which in turn would cause the calculated rotation degree to be prone to erroneous when noise exists.

Qualitive results are shown in Fig. 5.3. Images in the top row are the predicted results by

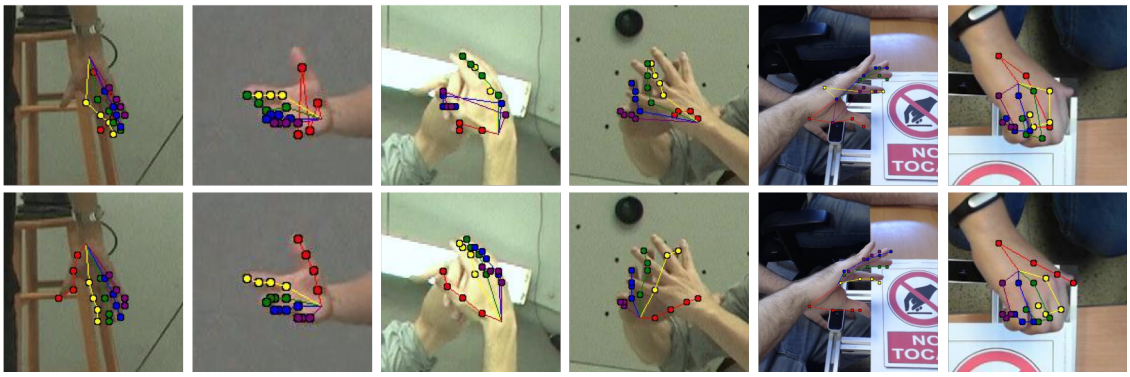


Figure 3.7: Qualitative results. First row: CPM. Second row: our model.

CPM, while the bottom row corresponds to the prediction of our model. The results show that our proposed R-MGNM could greatly reinforce the keypoints consistency, and generate much more reasonable predictions than CPM.

Our model succeed to predict well even if the hand is severely occluded, as in the 4-th column in Fig. 5.3. In this example, half of the right hand is occluded by the left hand. The CPM fails to recover many of the keypoints. However, our R-MGMN correctly recovers the index finger and thumb, even they are totally occluded.

3.5.3 Ablation study

To understand the proposed model, ablation study is also performed. Several experiments are conducted as follows.

- CPM+Single GM. In this experiment, we only keep the unary branch and one single graphical model from the R-MGMN. Both the rotation net and the soft classifier are removed.
- CPM+Mixture of GMs. The rotation net is removed from the R-MGMN.
- Rotaion+CPM[1]. Only keep the rotation net and the unary branch, jointly trained using the loss function in Eq. (3.25).

- Rotaion+CPM^2. First train the rotation net, then jointly the train the rotation net and the unary branch.

All of the above experiments support end-to-end training. Numerical results are given in Table 3.2. As indicated by the results, adding a single graphical model on top of CPM has very little effect on the PCK performance. By adding a mixture of graphical models, there is an improvement of 1.28 percent in mPCK. Properly tuned, the rotation net would help improve the performance by 3.35 percent. By integrating the rotation net and the mixture of graphical models together into our R-MGMN, final improvement of 4.76 percent is achieved.

3.6 Conclusion

A new architecture called Rotation-invariant Mixed Graphical Model Network (R-MGMN) is proposed in this chapter. The R-MGMN combines the graphical model and deep convolutional neural network in a new way, where a pool of graphical models could be selected softly depending on input image. The R-MGMN could be trained end-to-end. Experiment results validate that the proposed R-MGMN outperforms the widely used CPM algorithm on two public datasets. Ablation study is also performed to see the functionality of each part of the R-MGMN model.

Chapter 4

Adaptive Graphical Model Network

In this chapter, we propose a new architecture called Adaptive Graphical Model Network (AGMN) to tackle the challenging task of 2D hand pose estimation from a monocular RGB image. The AGMN consists of two branches of deep convolutional neural networks (DCNN) for calculating unary and pairwise potential functions, followed by a graphical model inference module for integrating unary and pairwise potentials. Unlike existing architectures proposed to combine DCNN with graphical models, our AGMN is novel in that the parameters of its graphical model are adaptive to individual input images. Experiments show that our approach outperforms state-of-the-art methods used in 2D hand keypoints estimation by a notable margin on two public datasets.

4.1 Introduction

Understanding human hand pose is a critical task for many real world AI applications, such as human-computer interaction, augmented reality and virtual reality. However, hand pose estimation remains very challenging because the hand is highly articulated and dexterous,

and hand pose estimation suffers severely from self-occlusion. An intuitive approach is to resort to multi-view RGB cameras [96, 48], which unfortunately requires expensive hardware and strict environment configurations. For practical daily applications, many researchers have also explored the problem under monocular RGB [139, 78, 7] or RGB-Depth [134, 3, 112] scenarios. Solving 3D pose estimation problem [139, 7] often relies on 2D hand pose estimation, making 2D hand pose estimation itself an important task. In this paper, we focus on the task of 2D hand pose estimation from a monocular RGB image.

The advent of Deep Convolutional Neural Networks (DCNNs) has enabled this field to make big progress in recent years. For example, the Convolutional Pose Machine (CPM) [124] is one of the most successful DCNNs that have been applied to 2D hand pose estimation [96], although it was originally proposed for the task of human pose estimation. However, despite the fact that DCNNs like CPM have the power to learn good feature representations, they often fail to learn geometric constraints among joints, resulting in joint inconsistency in the final prediction as observed in human pose estimation tasks [99, 50]. For 2D hand pose estimation, the situation could be even worse, since there are more articulations and self-occlusion is severer.

To model the relationships among joints, several studies have also explored the possibility of combining DCNN and the Graphical Model (GM) in pose estimation tasks. Existing methods [107, 19, 132, 99] all impose a self-independent GM on top of the score maps regressed by DCNNs. The parameters of the GM are learned during end-to-end training, then these parameters are fixed during prediction.

In this paper, we propose the Adaptive Graphical Model Network (AGMN), which is a brand new framework for combining DCNNs and GM. By "adaptive", we mean that the parameters of the GM should be able to adapt to different input images, instead of being fixed after training procedure. We argue that a hand in the shape of a fist and a hand which is widely open should have different spacial constraints among hand keypoints. Hands from different

views should also have different geometric models. The adaptivity of the GM is achieved by setting the parameters of the GM to be the output of a DCNN whose input is the image. Another DCNN is used to regress score maps of each hand joint location. Then the regressed score maps, which are treated as unary potential functions, are fed into the GM module. Final score maps are inferred by the GM using techniques like message passing. The whole AGMN architecture could be trained end-to-end.

We show the efficiency of our proposed framework on two public datasets: the CMU Panoptic Hand Dataset [96] and the Large-scale Multiview 3D Hand Pose Dataset[30]. Our approach outperforms the popularly used algorithm CPM by a noticeable margin on both datasets. Qualitative results show our model could alleviate geometric inconsistency among predicted hand keypoints significantly when severe occlusion exists.

The main contributions of this work are:

- We propose a novel framework integrating DCNNs and GM, making GM adaptive to different input images. In our proposed AGMN, parameters of the GM depends on individual input images directly, which distinguishes AGMN from existing architectures that also combine CNNs and GM.
- By implementing the message passing algorithm as a sequence of 2D convolutions, the inference is performed efficiently and the AGMN could be trained end-to-end.
- Our AGMN could reduce the inconsistency and ambiguity of hand keypoints significantly in scenarios of severe occlusion, as shown by experiments on two real world hand pose datasets.

4.2 Related Work

Human pose estimation. Research on hand pose estimation has benefited from the progress in the study of human pose estimation. On one hand, DCNNs have been successfully applied to human pose estimation [9, 40, 28] in recent years. The DCNN-based algorithms are typically equipped with well crafted deep architectures [98, 41] and/or multi-stage training technique[124, 75]. Since DCNNs have large receptive fields, they could learn salient and expressive feature representations. However, DCNNs could only capture structural constraints among body parts implicitly, resulting in limited performance in practice when severe occlusion and cluttering exist [99, 50]. Some approaches try to learn extra tasks (*e.g.*, offset fields [79], compound heatmaps [50]) besides heatmaps of joint positions, with the purpose of providing more additional structural information. Nevertheless, these methods still could not fully exploit structural information.

On the other hand, graphical model has also been exploited in solving human pose estimation tasks [19]. By using GM, one can model spatial constraints among body parts explicitly. Recently, there is also a trend to combine DCNN and GM for pose estimation [107, 19, 132]. The combination of DCNNs and GM has been studied in several scenarios, i.e., human pose estimation in a vedio[99], multi-person pose estimation [82, 43], multi-person pose tracking [42, 44]. However, graphical models in all of these approaches are not adaptive to individual input images.

Hand pose estimation. The 3D hand pose estimation is a challenging task due to strong articulation and heavy self-occlusion of hands. Some researcher try to solve the task efficiently with the help of multi-view RGB cameras [96, 48]. However, this kind of approaches are impractical for daily applications as they require expensive hardware and strict environment configurations. To circumvent this limitation, other studies have been focused on depth-based solutions [134, 3, 112] where RGB-D cameras are used. Due to the ubiquitous-

ness of regular RGB cameras, researchers also have a great interest in solving hand pose estimation from monocular RGB images [139, 78, 7].

2D hand pose estimation plays an important role in the task of estimating 3D hand pose, since 3D estimation is often inferred from 2D estimation[139, 78, 7]. Current algorithms on 2D hand pose estimation often directly deploy DCNN-based human pose estimators. Among a variety of DCNN-based models, CPM is commonly used in 2D hand pose estimation[96, 139, 78, 7], yielding state-of-art performance. Thus, in this work, we choose CPM as the baseline for comparison with our proposed model.

4.3 Method

4.3.1 Basic Framework of Adaptive Graphical Model Network

As shown in Fig. 4.1 (a), due to lack of explicit structural information, CPM fails when the hand is occluded severely, resulting in hand keypoints' spatial inconsistency. To alleviate this problem, we propose the novel adaptive graphical model network (AGMN), the efficiency of which could be seen from the right image of Fig. 4.1 (a). The model contains two DCNN branches, the *unary branch* and the *pairwise branch*, and a graphical model inference module, as depicted in Fig. 4.1 (b). The unary branch would output intermediate score maps of K hand keypoints. Any existing DCNN-based pose estimator that regresses score maps could be used as the unary branch. The pairwise branch produces parameters that characterize pairwise spatial constraints among K hand keypoints. These parameters would be later used in the graphical model. It is the pairwise branch that makes our model distinguish from existing models [107, 19, 132, 99] which also try to combine graphical model with DCNNs. In our approach, the parameters of the graphical model are not independent parameters. Instead, they are closely coupled with the input image via a DCNN and they are adaptive

to different input images. In approaches from [107, 19, 132, 99], once the graphical model parameters are learned, they would be fixed and used for different input images in future prediction.

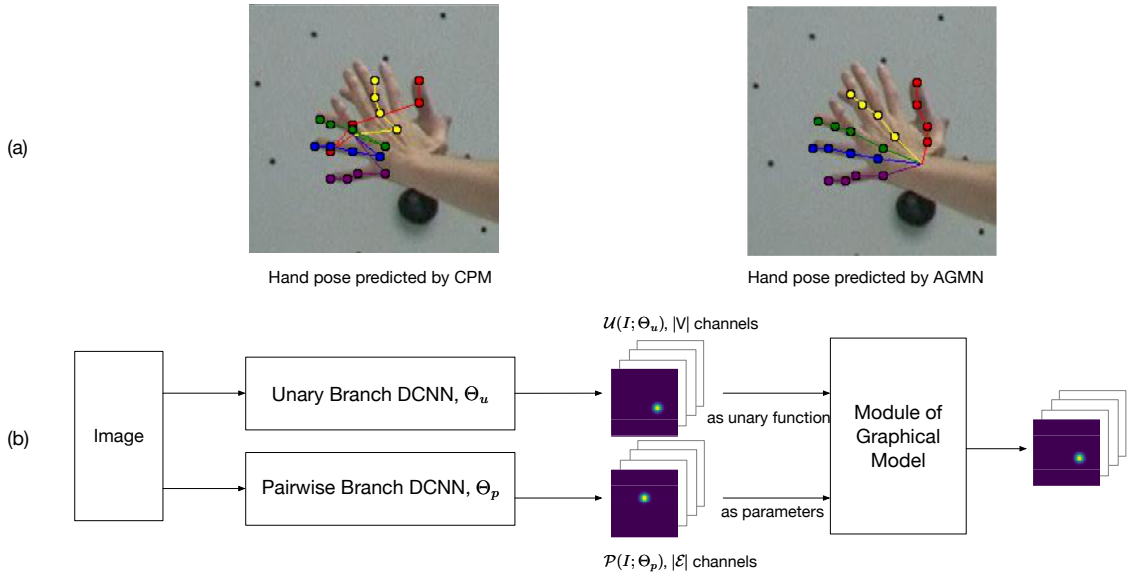


Figure 4.1: Basic flow diagram of adaptive graphical model network.

The hand pose estimation problem could be formulated by using a graph. Let $G = (V, \mathcal{E})$ denote a graph with a vetex set V and an edge set $\mathcal{E}$, where $V = \{v_1, v_2, \cdots, v_K\}$ corresponds to the set of hand keypoints and $\mathcal{E} \subseteq V \times V$ is the set of edges between neighboring keypoints. Let the discrete variable $x_i \in \mathbb{R}^2$ denote the 2D position of the keypoint associated with v_i.

The joint probability distribution of a hand pose configuration is given by

$$p(X|I;\Theta) = \frac{1}{Z} \prod_i \phi_i(x_i|I;\Theta_u) \prod_{(i,j)\in\mathcal{E}} \varphi_{i,j}(x_i, x_j|I;\Theta_p), \tag{4.1}$$

where $X = \{x_1, x_2, \cdots, x_K\}$ represents positions of all the keypoints, I stands for the input image and Z is the partition function. The whole set of AGMN's prameters Θ consists of two components, parameters for the unary branch and that for the pairwise branch, i.e., $\Theta = \{\Theta_u, \Theta_p\}$.

Unary Terms. The non-negative term $\phi_i(x_i|I;\Theta_u) \in \mathbb{R}$ is the local confidence of the appearance of the i-th keypoint at location x_i. Let $\mathcal{U}(I;\Theta_u) \in \mathbb{R}^{|V|\times h_u \times w_u}$ denote the output of the unary branch in Fig. 4.2, where $|V|$ is the cardinality of the set V, w_u and h_u are the width and height of the output heatmap. We define

$$\phi_i(x_i|I;\Theta_u) = \max\left(0, \mathcal{U}^i_{x_i}(I;\Theta_u)\right), \tag{4.2}$$

where $\mathcal{U}^i_{x_i}(I;\Theta_u) \in \mathbb{R}$ is the value of the i-th channel of $\mathcal{U}(I;\Theta_u)$ evaluated at location x_i.

Pairwise Terms. The term $\varphi(x_i, x_j|I,\Theta_p) \in \mathbb{R}$ represents the pairwise potential function between the i-th and j-th keypoints, if (i,j) forms an edge in the graphical model. It encodes spatial constraints between two neighboring keypoints. The pairwise term is given by

$$\varphi_{i,j}(x_i, x_j|I;\Theta_p) = \mathcal{F}(x_i, x_j; \theta^{(i,j)}), \tag{4.3}$$

$$\theta^{(i,j)} = \max(0, \mathcal{P}^{(i,j)}(I;\Theta_p)), \tag{4.4}$$

where $\mathcal{P}(I;\Theta_p) \in \mathbb{R}^{|\mathcal{E}|\times h_p \times w_p}$ is the output of the pairwise branch in Fig. 4.2, $\mathcal{P}^{(i,j)}(I;\Theta_p) \in \mathbb{R}^{h_p \times w_p}$ is a channel of $\mathcal{P}(I;\Theta_p)$ corresponding to the pair of the i-th and j-th keypoints. Function $F(\cdot)$ is defined as $\mathcal{F}(x_i, x_j; \theta^{(i,j)}) = \theta^{(i,j)}_{x_i - x_j}$, which is an entry of the matrix $\theta^{(i,j)} \in \mathbb{R}^{h_p \times w_p}$, indexed by the relative position of the i-th keypoint with respect to the j-th keypoint.

One can also design $\theta^{(i,j)}$ as a set of parameters of a spring model , and then define $\mathcal{F}(\cdot)$ as a quadratic function as in [19, 132, 99] . In this work we follow the idea in [107] and design $\theta^{(i,j)}$ to be a 2D matrix, which has a much larger parameter space.

Inference. The final score maps generated by AGMN are the marginal distributions of $p(X|I;\Theta)$ given in Eq.(5.1). The marginals are defined as

$$p_i(x_i|I;\Theta) = \sum_{V\setminus x_i} p(X|I;\Theta), \tag{4.5}$$

which is computed in the module of graphical model. Finally, the predicted position of hand keypoint i is obtained by maximizing its marginal probability as

$$x_i^* = \underset{x_i}{\operatorname{argmax}}\, p_i(x_i|I;\Theta). \tag{4.6}$$

In summary, the complete parameters in the AGMN model is given by $\Theta = \{\Theta_u, \Theta_p\}$, consisting the parameters from the unary branch and pairwise branch.

4.3.2 Detailed Structure of AGMN

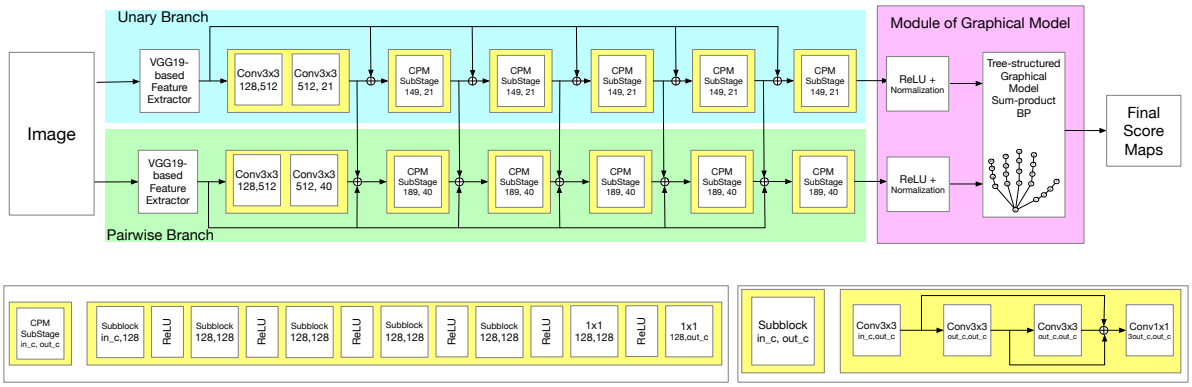


Figure 4.2: More detailed inllustration of adaptive graphical model network.

The detailed structure of AGMN is shown in Fig. 4.2.

Unary branch. The structure of the unary branch is the same as the CPM detection architecture used in [96]. A pre-initialized VGG-19 network [98] up to conv4_4 and additional convolutions are used to produce the 128-channel features, then several prediction stages follows. The output of the unary branch is a 21-channel score map, each channel corresponding to one keypoint of the hand.

Pairwise branch. The pairwise branch follows the similar structure of the unary branch. The only difference is that the pairwise branch outputs a 40-channel kernel instead of a 21-

channel score map. This 40-channel kernel would later be utilized in the module of graphical model. The reason why the channel of this kernel is designed to be 40 would be clear after we talk about message passing later. There are also some information flowing from the unary branch to the pairwise branch, as indicated by the arrows between the unary branch and pairwise branch in Fig. 4.2. We found that adding such information flows would benefit the performance.

Inference. *Message Passing.* Sum-product algorithm is widely used for efficient calculation of marginals in a graphical model. Vertices receive messages from and send messages to their neighbors. The sum-product algorithm updates the message sent from hand keypoint i to keypoint j as follows:

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j)\phi_i(x_i) \prod_{k \in Nbd(i) nj} m_{ki}(x_i). \tag{4.7}$$

Let M_{ij} denote the complete message sent from keypoint i to j, then $M_{ij} \in \mathbb{R}^{h_u \times w_u}$, since since x_j could take values from a set of grid points which has the size of $h_u \times w_u$. After several iterations or convergence, the marginal probabilities are approximated by

$$p_i(x_i) \approx \frac{1}{Z'}\phi_i(x_i) \prod_{k \in Nbd(i)} m_{ki}(x_i), \tag{4.8}$$

where Z' is just a normalization term.

Tree Structured Graphical Model. In our implemented AGMN, a tree-structured graphical model is used. One advantage of tree-structured model is that exact marginal probability in Eq.(4.8) could be obtained by belief propagation. The tree-structured hand model is shown in Fig. 4.3. By passing messages from leaves to root (upwards) and then from root to leaves (downwards), the exact marginal can be reached. The numbers along side each arrow in Fig. 4.3 indicates the schedule of massage updates. In total, we only need pass messages 40

times to obtain exact marginals.

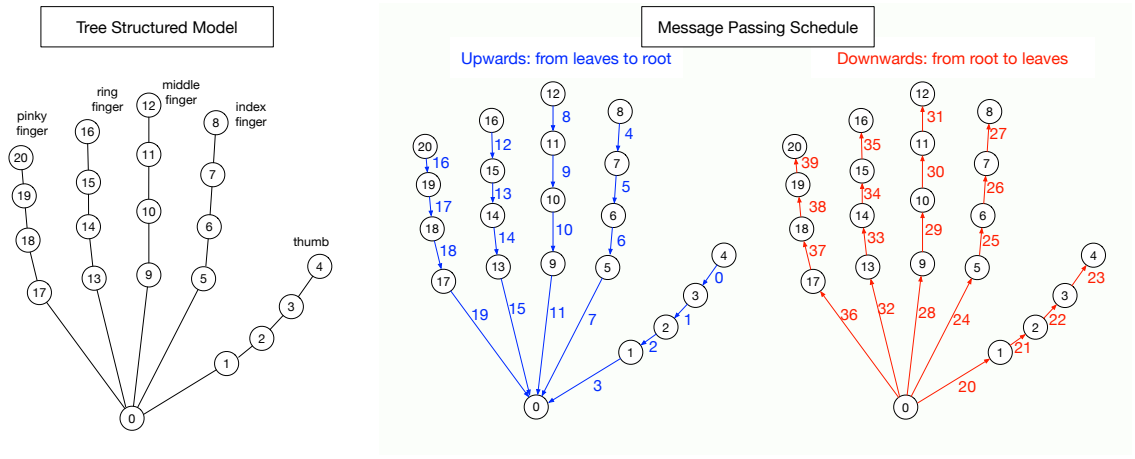


Figure 4.3: Tree structured model and message passing schedule.

Message updates as convolution operations. When implementing Eq. (4.7), one way to avoid the for loop in the summation is to use matrix product. However, if we write $\varphi_{i,j}(x_i, x_j)$ compactly in to a matrix, the dimension of this matrix is huge. Since x_i and x_j could both take $h^u \times w^u$ different values, The matrix storing $\varphi_{i,j}(x_i, x_j)$ would have the size of $(h^u \times w^u)^2$. To save memories during the inference, we resort to convolution operations when performing message passing.

The message update formula in Eq. (4.7) could be rewrittern as

$$m_{ij}(x_j) = \sum_{x_i} \varphi_{i,j}(x_i, x_j) h_i(x_i), \tag{4.9}$$

$$h_i(x_i) = \phi_i(x_i) \prod_{k \in Nbd(i) nj} m_{ki}(x_i). \tag{4.10}$$

We could rewrite Eq. (4.9)in a form of 2D convolution, if $(i, j) \in \mathcal{E}$,

$$M_{ij} = \theta^{i,j} * H_i, \; M_{ji} = \left(\theta^{i,j}\right)^T * H_j, \tag{4.11}$$

where $M_{ij} \in \mathbb{R}^{h_u \times w_u}$, $\theta^{i,j} \in \mathbb{R}^{h_p \times w_p}$, $\theta \in \mathbb{R}^{|\mathcal{E}| \times h_p \times w_p}$, $H_i \in \mathbb{R}^{h_u \times w_u}$. The matrix H_i is the

compact matrix formed by values of $h_i(x_i)$. Appropriate zeroing padding is required on H_i to make the shape of M_{ij} is the same as that of H_i. The similar idea is also used in [107]. Kernel $\theta^{i,j}$ could be interpreted as the probability of where keypoint j would be with respect to keypoinjt i, and it encodes the information of relative positions between the keypoint i and j. In our implementation in Fig. 4.2, to avoid the transpose operation in Eq.(4.11), we let the pairwise branch produce an output $Q \in \mathbb{R}^{2|\mathcal{E}|\times h_p \times w_p}$ which has $2 \times |\mathcal{E}| = 40$ channels.

4.4 Leaning

Since there are two branches of DCNN in the proposed AGMN, we utilize a 3-stage training strategy. Firstly, the unary branch is trained. Then, the pairwise branch is trained with the unary branch fixed. Finally, the whole AGMN is finetuned end-to-end.

Train unary branch. The unary branch is trained alone first. As in [124, 96], intermediate supervision is used during the training. Each stage of the unary branch is trained to repeatedly produce the score maps (or belief maps) for the locations of each of the hand keypoints. The ground truth score map of keypoint i, denoted as $S_i^* \in \mathbb{R}^{h_u \times w_u}$, is created by putting a Gaussian peak at its ground truth location. The cost function at each stage t of the unary branch is defined by

$$f_t = \sum_{k=1}^{21} ||S_k^t - S_k^*||_F^2, \tag{4.12}$$

where S_k^t is the score map of keypoint k generated by the t-th stage in the unary branch. Notation $||\cdot||_F$ represents the Frobenius norm which is defined as the square root of the sum of the squares of its elements. If we have T stages, then $S_k^T = \mathcal{U}^k(I; \Theta_u)$ in Eq.(4.2). By adding up the cost functions at each stage, the final loss function of the unary branch is

$$L^{unary} = \sum_{t=1}^{T} f_t. \tag{4.13}$$

Train pairwise branch. The pairwise branch is trained with the help of the unary branch, since there are some information flowing from the unary branch to the pairwise branch as shown in Fig. 4.2. Parameters of the unary branch are frozen during this training phase.

The goal of the pairwise branch is to learn relative positions between hand keypoints. The pairwise branch produces an output Q of 40 channels, with each channel corresponding to one directed edge in the message passing schedule. The ideal output (ground truth) Q^* of the pairwise branch is computed from relative positions of each pair of neighboring hand keypoints which share a common edge in the tree structure. For example, if the k-th directed edge (right side of Fig. 4.3) incidents on two hand keypoints i and j, say starting from i to j, the relative position of these two keypoints is computed as $r_k = l_j - l_i$, where l_i and l_j are length-2 vectors representing the ground truth positions of the keypoints. Then, the ground truth of the k-th channel of Q^*, i.e., Q_k^*, is created by putting a Gaussian peak at the location which is r_k away from the center of the 2D matrix.

We use a similar loss function as that in training the unary branch

$$L^{pairwise} = \sum_{t=1}^{T} \sum_{k=1}^{40} ||Q_k^t - Q_k^*||_F^2. \tag{4.14}$$

Fine tune the whole AGMN. Since the final outputs of the AGMN are marginal probabilities, the ground truth for the final score map of keypoint k, FS_k^* is set to be the normalized version of S_k^* used in Eq.(4.12). The loss function defined by the final score maps is given by

$$L^{last} = \sum_{k=1}^{21} ||FS_k - FS_k^*||_F^2, \tag{4.15}$$

where FS_k is the k-th channel of the output of the AGMN.

The whole AGMN is fine tuned with a loss function which is a weighted sum of loss functions

from the unary branch, pariwise branch and module of graphical model as following

$$L = \alpha_1 L^{unary} + \alpha_2 L^{pairwise} + \alpha_3 L^{last}. \tag{4.16}$$

4.5 Experiments

In this section, we demonstrate the performance of our proposed algorithm on two real-world hand pose datasets. Comparative analysis is also carried out.

4.5.1 Experimental settings

Datasets. We evaluate our model on two public datasets, the CMU Panoptic Hand Dataset (referred to as "CMU Panoptic")[96], and the Large-scale Multiview 3D Hand Pose Dataset (referred to as "Large-scale 3D") [30]. (i) The CMU Panoptic dataset contains 14817 annotations of right hands in images of persons from Panoptic Studio. Since our focus is on hand pose estimation other than hand detection, we cropped image patches of annotated hands off the original images using a square bounding box which is 2.2 times the size of the hand. Then, we randomly split the whole dataset into training set (80%), validation set (10%) and test set (10%). (ii) The Large-scale 3D dataset contains 82760 images in total. We follow the same preprocessing procedure on this dataset and take care of the keypoints ordering. Although this is a 3D dataset, it provides an interface to get 2D annotations by performing projection. The whole Large-scale 3D dataset is split it into training set (60000 images), validation set (10000 images) and test set (12760 images).

Evaluation metric. We consider the "normalized" Probability of Correct Keypoint (PCK) metric from [96]: the probability that a predicted keypoint is within a distance threshold σ of its true location. We use a normalized threshold σ which ranges from 0 to 1, with respect

to the size of hand bounding box.

All images are resized to 368×368 before fed into the AGMN, yielding a final score map of size 46×46 for each keypoint. Also, after being scaled to [0,1], all the images are then normalized using mean = (0,485, 0,456, 0,406) and std = (0,229, 0,224, 0,225). During training, the batch size is set to 32. The gaussian peaks used to generate ground truth during training all have standard deviation of 1. Learning rate is set to 1e-4 when training the unary branch and pairwise branch. When finetuning the whole AGMN, learning rate is set to 1e-5 and the coefficients in Eq.(4.16) are set to $\alpha_1 = 1, \alpha_2 = 0.1, \alpha_3 = 0.1$.

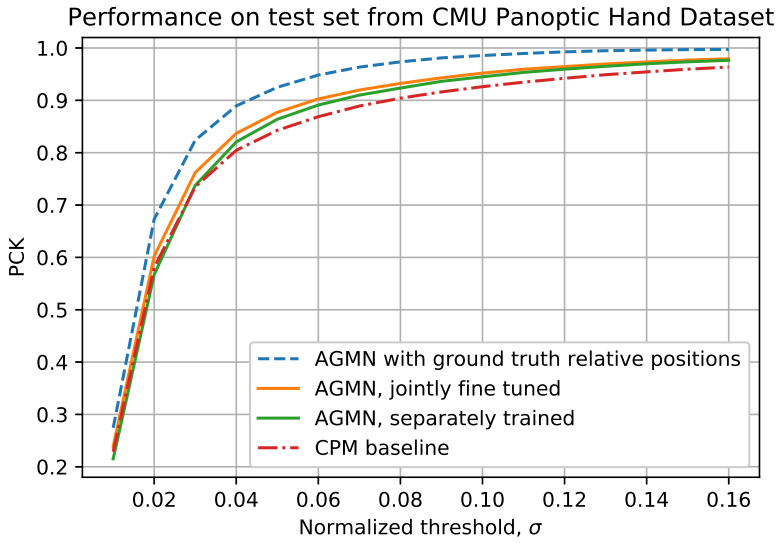
4.5.2 Results

Fig. 4.4 shows our model's performance on above mentioned datasets. Detailed numerical results are summarized in Table. 4.1. It is seen that our model outperforms CPM consistently.

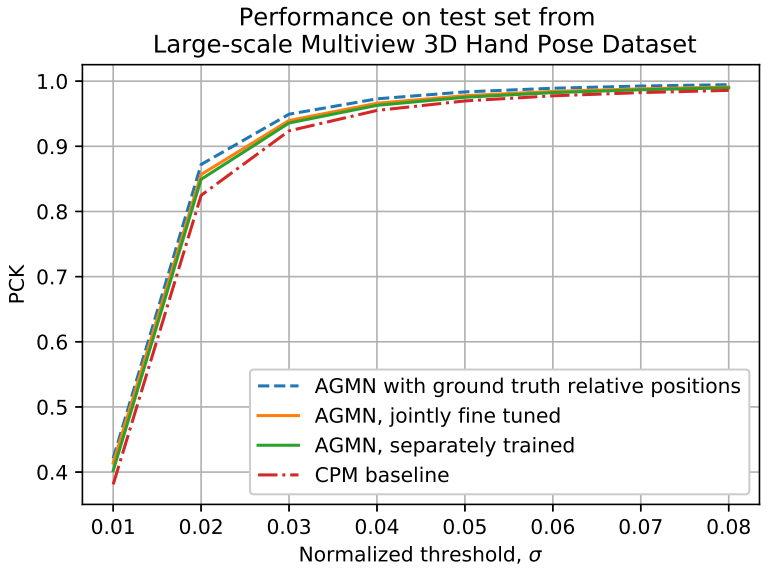
On CMU Panoptic dataset, by training the unary branch and pairwise branch separately, we see an absolute PCK improvement of 2.12% at threshold $\sigma = 0.05$. A final improvement of 3.45% is obtained after finetuning the unified AGMN. On Large-scale 3D dataset, our AGMN obtains its highest improvement 3.27% at thresholds $\sigma = 0.01$. The authors in [107] stated that "Spatial-Model has little impact on accuracy for low radii threshold". However, based on the results in Fig. 4.4(b), it is observed that our adapitve spatial model has the power of increasing accuracy for low radii threshold.

The reason why AGMN achieves highest improvement on CMU Panoptic dataset at higher threshold σ than that of Large-scale 3D dataset, probably lies in the fact that CMU Panoptic dataset is a much harder dataset where a lot more occlusions exist.

We also conducted an experiment where the ground truth of the relative positions among hand keypoints (Q^* in Eq.(4.14)) are given to the AGMN, with pre-trained unary branch.



(a) CMU Panoptic.



(b) Large-scale 3D Dataset.

Figure 4.4: Model performance.

The result of this experiment is actually the upper bound of our AGMN's performance given specific unary branch. The result is drawn as the blue dashed lines in Fig. 4.4.

Threshold of PCK, σ	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
CMU Panoptic Hand Dataset										
CMP Baseline (%)	22.88	58.10	73.48	80.45	84.27	86.88	88.91	90.42	91.61	92.61
AGMN Sep. Trained	21.52	56.73	73.75	82.06	86.39	89.10	91.00	92.35	93.63	94.50
AGMN Finetuned	23.90	60.26	76.21	83.70	87.72	90.27	91.97	93.23	94.30	95.20
Improvement	1.02	2.16	2.73	**3.25**	**3.45**	**3.39**	**3.06**	2.81	2.69	2.59
Large-scale Multiview 3D Hand Pose Dataset										
CMP Baseline (%)	38.11	82.48	92.37	95.50	96.97	97.75	98.24	98.58	98.84	99.02
AGMN Sep. Trained	40.22	84.94	93.57	96.29	97.53	98.24	98.68	98.97	99.17	99.34
AGMN Finetuned	41.38	85.67	93.96	96.61	97.77	98.42	98.82	99.10	99.29	99.43
Improvement	**3.27**	**3.19**	1.59	1.11	0.80	0.67	0.58	0.52	0.45	0.41

Table 4.1: Detailed numerical results.

Examples in Fig. 4.5 show that our AGMN could greatly reinforce the keypoints consistency and reduce ambiguities in prediction. Note that the first keypoint in Large-scale 3D dataset is the center of the palm.

4.6 Conclusion

This chapter provides a new direction on how deep convolutional neural networks can be combined and integrated with graphical models. We propose an adaptive framework called AGMN for hand pose estimation, which contains two branches of DCNN, one for regressing the score maps of hand keypoint positions, the other for regressing the parameters of graphical model, followed by a graphical model for inferring the final score maps through message passing. Experiment results show that the proposed AGMN outperforms the commonly used CPM algorithm on two public hand pose datasets. The proposed framework is general and

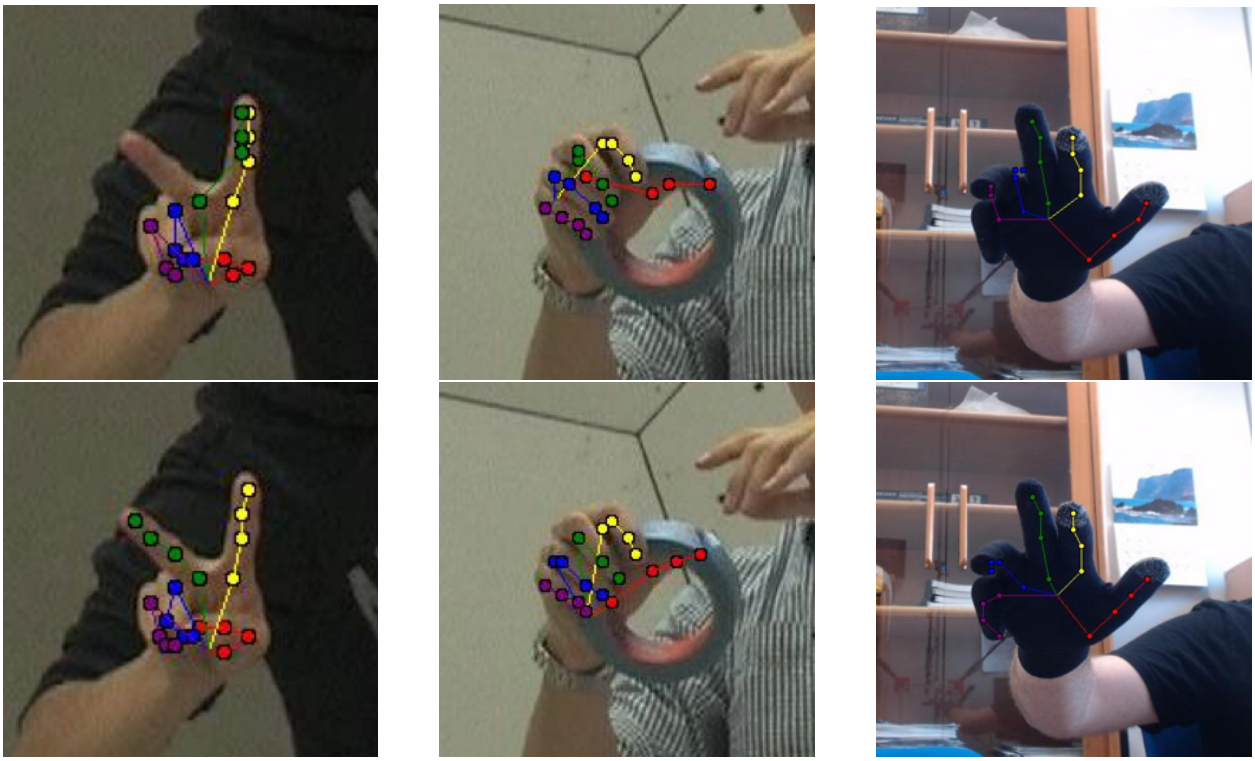


Figure 4.5: Predicted hand keypoint positions. For each pair of images, the top image shows the result of CPM and the bottom image shows that of AGMN.

can also be applied to other deep learning applications where performance can benefit by considering structural constraints.

Chapter 5

SIA-GCN: A Spatial Information Aware Graph Neural Network with 2D Convolutions

Graph Neural Networks (GNNs) generalize neural networks from applications on regular structures to applications on arbitrary graphs, and have shown success in many application domains such as computer vision, social networks and chemistry. In this paper, we extend GNNs along two directions: a) allowing features at each node to be represented by 2D spatial confidence maps instead of 1D vectors; and b) proposing an efficient operation to integrate information from neighboring nodes through 2D convolutions with different learnable kernels at each edge. The proposed SIA-GCN can efficiently extract spatial information from 2D maps at each node and propagate them through graph convolution. By associating each edge with a designated convolution kernel, the SIA-GCN could capture different spatial relationships for different pairs of neighboring nodes. We demonstrate the utility of SIA-GCN on the task of estimating hand keypoints from single-frame images, where the nodes represent the 2D coordinate heatmaps of keypoints and the edges denote the kinetic relationships

between keypoints. Experiments on multiple datasets show that SIA-GCN provides a flexible and yet powerful framework to account for structural constraints between keypoints, and can achieve state-of-the-art performance on the task of hand pose estimation.

5.1 Introduction

Hand pose estimation is a long standing research area in computer vision, given its vast potential applications in computer interaction, augmented reality, virtual reality and so on [25]. It aims to infer 2D or 3D positions of hand keypoints from a single input image or a sequence of images, which could possibly take the form of RGB, RGB-D or grayscale. Although 3D hand pose estimation is drawing increasing attention in the research community [119, 68, 128, 111, 134, 33], 2D hand pose estimation still remains a valuable and challenging problem [96, 116, 53]. A plentiful of 3D hand pose estimation algorithms rely on their 2D counterparts [7, 139], attempting to lift 2D predictions to 3D space. In this paper, we investigate the problem of 2D handpose estimation from single RGB image.

The progress in hand pose estimation research has been boosted greatly by the invention of deep Convolutional Neural Networks (CNNs). Deep CNN models like Convolutional Pose Machine [124] and Stacked Hourglass [75] have been successfully applied to 2D hand pose estimation, though they are originally proposed to solve the task of human pose estimation. Some methods [54, 53, 19] also integrate deep CNNs with probabilistic graphical model to harvest both the powerful representation ability of deep CNNs and the capability of explicitly expressing spatial relationships attributed to graphical model.

In contrast to CNN, graph neural network has the ability to handle irregular structured data. The joints of a human body, and keypoints of a hand can be conveniently considered as irregular graphs, giving possibilities of applying Graph Convolutional Network (GCN) [52]

on human/hand pose estimation tasks. However, in the vanilla GCN [52], all the nodes share the same one-hop propagation weight matrix, which makes it unready to be applied to pose estimation task because different human body joints and bones should have different semantics. Authors in [26, 136, 8] have proposed different variants of the vanilla GCN from [52] for the purpose of human or hand pose estimation. However, all these methods take as input a one dimensional vector for each node, and the node feature at each layer is always a one dimensional vector. Thus, they are not ready to process 2D confidence map. Although, in [26, 136, 8], modifications are made to vanilla GCN, they still do not allow full independence among the edges.

In this paper we propose the Spatial Information Aware Graph Neural Network with 2D convolutions (SIA-GCN). In SIA-GCN, the feature of each node is a two dimensional matrix, and the information propagation to neighboring nodes are carried out via 2D convolutions along each edge. By using 2D convolutions instead of flattening the 2D feature map to a 1D vector and then performing linear multiplications, the spatial information encoded in the feature map is reserved and appropriately exploited. We also propose to use different 2D convolutional kernels on different edges, aiming to capture different spatial relationships for different pairs of neighboring nodes. The SIA-GCN is very flexible and could be easily combined with off-the-shelf 2D pose estimators. In this work, we demonstrate the efficacy of SIA-GCN on 2D hand pose estimation. For this application, the 2D feature maps at the nodes are actually the confidence maps of the hand keypoint positions. With a designated matrix for each edge, the SIA-GCN has the ability to capture various spatial relationships between different pairs of hand keypoints.

Our main contributions are threefold:

- We propose the novel SIA-GCN which can process 2D confidence maps for each node efficiently and effectively, by integrating graph neural networks and 2D convolutions. Using 2D convolutions, our SIA-GCN can exploit and harvest the spatial information

provided in the 2D feature maps.

- By assigning different convolutional kernels on different edges, the SIA-GCN has the property of full edge-awareness. Distinct spatial relationships can be learned on different edges.
- We deploy SIA-GCN in the task of hand pose estimation. Utilizing SIA-GCN, the constructed neural network can achieve state-of-the-art performance.

5.2 Related Work

There exists a vast amount of research focusing on topics of human/hand pose estimation [96, 119, 68, 128, 111, 122, 134, 3, 112, 33, 74, 31] and graph neural networks [96, 26, 136, 8]. In the related work, we focus on 2D hand pose estimation from single RGB images and graph convolutional network [96]'s applications to pose estimation tasks.

2D hand pose estimation. Studies of RGB image based 2D hand pose estimation has long benefited from that of human pose estimation, where deep Convolutional Neural Networks (CNNs) have enjoyed great success [108, 124, 75, 126, 21, 104]. Among these deep CNN models, Convolutional Pose Machines [124] and Stacked Hourglass [75] are commonly used in various RGB-based 2D hand pose estimation methods [96, 54, 53, 20, 116]. Compared with deep CNNs, Graphical Model (GM) has also played a significant role in solving the pose estimation task. GM has the power of modeling spatial constraints among the joints explicitly. Recently, several works in pose estimation combine GM and neural network to fully exploit the structural information [107, 19, 99, 132, 53, 54]. Traditionally, GM with fixed parameters [107, 99, 19] are applied to the pose estimation task, while the most recent work in [53, 54] propose to adopt GM with adaptive parameters conditioning on input images. Although all take advantage of structural information, our proposed method is based on graph convolutional network while these previous works [53, 54] are based on graphical

models.

Graph convolutional network. Graph Convolutional Network (GCN), which generalizes deep CNNs to graph structured data, have attracted increasing attention in recent years. One main research direction is to define graph convolutions from the spectral perspective [94], while the other works on the spatial domain [52]. For a comprehensive survey on GCN, we refer readers to [125]. The most related works to ours are [26, 136, 8], in which variants of spatial GCNs have been proposed and applied to human/hand pose estimation tasks in the computer vision field. In the following, we discuss the key differences between our SIA-GCN and those in [26, 136, 8].

In [8], the authors have proposed to classify neighboring nodes according to their semantic meanings and use different kernels for different neighboring nodes. The purpose of their proposed GCN is to regress 3D position vectors from 2D position vectors, and the input to the GCN for each node is a one dimensional $\mathbb{R}^2$ vector, representing predicted 2D position of a corresponding body joint. However, our proposed SIA-GCN aims to handle two dimensional confidence maps for each node. The confidence map inherently contains much more information than the two-element position vector. Our goal is to refine final 2D predictions, other than lifting 2D predictions to 3D space. Besides, instead of classifying nodes into different classes, we treat every edge independently and attach a designate weight kernel to each edge.

In [26], the authors directly adopt the propagation rule from [52] with the modification that, instead of using a predefined adjacency matrix, they have proposed to use an adaptive adjacency matrix which could be learned from data. The feature for each node is a one dimensional vector. Our method differs from [26] in that edge-dependent weights are considered explicitly and our SIA-GCN works on 2D confidence maps for each node.

In [136], the proposed Semantic Graph Convolution (SemGConv) adds a learnable weighting

matrix to conventional graph convolutions from [52]. The weight matrix serves as a weighting mask on the edges of a node when information aggregation is performed. The SemGConv is inherited from ST-GCN [129], but is equipped with additional important features such as softmax non-linearity and channel wise masks. The weighting mask adds a scalar importance weight (or a vector if it's channel wise) to each edge. However, in SIA-GCN, we directly attach to each edge a fully independent convolution matrix. Besides, our SIA-GCN works on 2D node features with spatial information awareness.

5.3 Methodology

In this section, we present the SIA-GCN, and its application to hand pose estimation. We refer to the resulted pose estimator as SiaPose, which is illustrated in Fig 5.1.

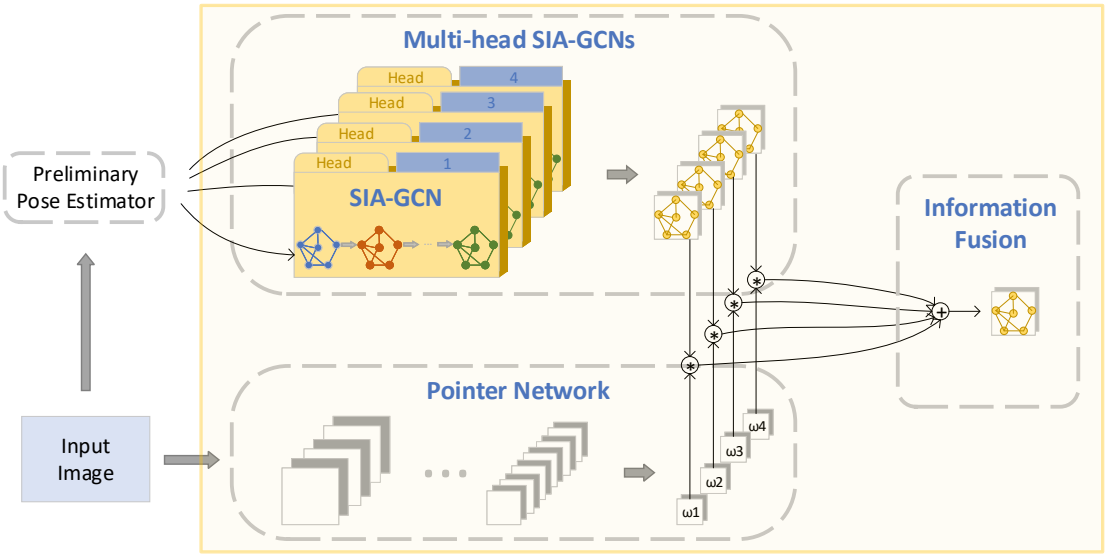


Figure 5.1: System diagram of the SiaPose, utilizing SIA-GCN.

The SiaPose takes as input a RGB image, to which a preliminary pose estimator is applied. The preliminary pose estimator could be any 2D pose estimator, such as the famous Convolutional Pose Machine [124] and Stacked Hourglass [75], which would output a set of

confidence maps of keypoint positions. Then, at the top branch, the confidence maps are fed into a block of multi-head SIA-GCNs. Each SIA-GCN processes a copy of the confidence maps parallelly and independently. Meanwhile at the bottom branch, the input image goes through a pointer network, which gives a weight vector, indicating which head is important in the multi-head SIA-GCNs. Finally, at the information fusion stage, confidence maps output from the multi-head SIA-GCNs are aggregated according to the weight vector.

In the following subsections, we revisit the graph convolutional network first, and discuss the motivation for our SIA-GCN. Then, we present a compact formulation of our proposed edge-aware graph convolutional layers in SIA-GCN, and demonstrate how to implement it efficiently using 2D convolutional operations. Finally, we describe the training procedure of the SiaPose.

5.3.1 Revisiting Graph Convolutional Network

The Graph Convolutional Network (GCN) proposed in [52] has enjoyed great success on a variety of applications since its advent. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes $v_i \in \mathcal{V}$, edges $(v_i, v_j) \in \mathcal{E}$, adjacency matrix $A \in \mathbb{R}^{N \times N}$, and a degree matrix $D \in \mathbb{R}^{N \times N}$ with $D_{ii} = \sum_j A_{ij}$, the layer-wise propagation rule is characterized by the following equation

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \tag{5.1}$$

where $\tilde{A} = A + I_N$ is the adjacency matrix of the undirected graph $\mathcal{G}$ with self-connections [52]. I_N is the identity matrix, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(l)} \in \mathbb{R}^{N \times M}$ is the matrix of activations in the l^{th} layer, or input feature matrix of the l^{th} layer. The parameter $W^{(l)}$ is the trainable weight matrix of layer l.

In the scenario of human and hand pose estimation, it is well studied that probabilistic

graphical models could be deployed to enhance structural consistency [107, 54, 19]. The graphical model could take in some preliminarily generated 2D confidence maps of each body joint or hand points. These confidence maps are usually considered as the unary potential functions by the graphical model. Then the graphical model could impose some learned pairwise potential functions on the initial confidence maps, thus enforcing spatial consistency of the body joints/keypoints. Can we also apply GCN to the confidence maps and then enhance spatial consistency?

The answer is positive, but it's not trivial. To apply the above GCN to pose estimation, some modifications are needed due to the dimensionality. In Eq. (5.1), the activation matrix $H^{(l)} \in \mathbb{R}^{N \times M}$ is a two dimensional matrix, corresponding to N nodes and each node is associated with a 1-d feature of size M. However, for the case of 2D pose estimation, each graph node (usually corresponding to a joint or keypoint) can be associated with a two dimensional confidence map. This discrepancy could be handled by flattening the two dimensional confidence map to a single long vector and then perform layer propagation according to Eq. (5.1). However, this would result in very large feature size, significantly increase the computational complexity (imagine that a 64×64 matrix would result in a one dimensional vector of size 4069). Besides, by flattening the confidence map, spatial information encoded in the confidence map would be corrupted. Thus, we propose to use 2D convolutional operations directly on 2D confidence maps when propagating information along the edges.

Moreover, in Eq. (5.1), since all the node share the same weight matrix $W^{(l)}$ and information aggregation is only controlled by the adjacency relationships between nodes, it would be difficult for the propagation rule in Eq. (5.1) to characterize different positional relationships for different pairs of neighboring joints. For example, the positional information propagation between two neighboring thumb joints should be different from that between the neighboring joints on the middle finger. One simple reason is that the bones from the thumb and middle

finger actually have different lengths.

5.3.2 SIA-GCN

To resolve the above mentioned concerns, we propose the spatial information aware graph neural network with 2D convolutions (SIA-GCN), where each edge of the graph is associated with an individual learnable 2D convolutional kernel. A toy example of a graph consisting of four nodes is shown in Fig. 5.2, where green matrices represent 2D features (heatmaps) at each node and red matrices represent designated 2D kernels associated with each edge.

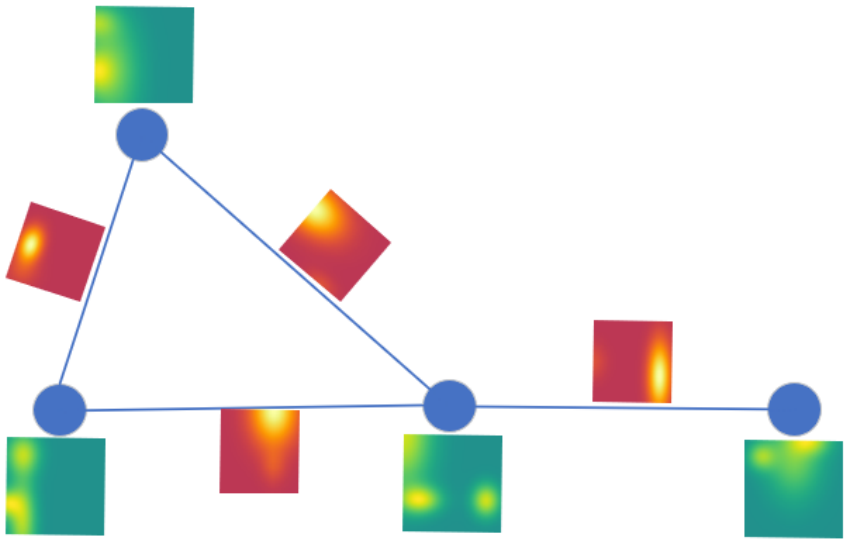


Figure 5.2: A simple illustration of SIA-GCN.

For the task of hand pose estimation, we could define a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \{v_1, v_2, \cdots v_K\}$ is the set of nodes corresponding to K hand keypoints, and $\mathcal{E}$ is the set of edges encoding the neighboring relationships among the keypoints. Each node v_i is associated with a 2D confidence map $X_i \in \mathbb{R}^{h \times w}$, which encodes the positional information of i^{th} keypoint. We could stack all $\{X_i\}$ for $i = 1, 2, .., K$ in a 3D matrix, and denote it as $X \in \mathbb{R}^{K \times h \times w}$.

One important feature of our SIA-GCN is that each edge in $\mathcal{E}$ is associated with an individual weight matrix or 2D convolutional kernal, $F_j \in \mathbb{R}^{h' \times w'}$, $j = 1, 2, \cdots, |\mathcal{E}|$. Again, we compact

all $\{F_j\}$ into a single matrix $F \in \mathbb{R}^{|\mathcal{E}| \times h' \times w'}$, which is actually the set of learnable parameters of the edge-aware graph convolutional layer. The information propagated from node i to node j along edge $e_{i,j}$ is obtained by calculating the 2D convolution of $X_i \circledast F_{e_{i,j}}$. Then, all the information propagated into node i are aggregated according to the adjacency matrix. The propagation rule could be presented compactly in matrix multiplications and convolutions as

$$X^{(l+1)} = \sigma\left(\hat{A}\left((BX^{(l)})\tilde{\circledast}F^{(l)}\right)\right), \tag{5.2}$$

where the superscript l and $l+1$ denote the l^{th} layer and $l+1^{th}$ layer respectively, $\tilde{\circledast}$ is the channel-wise 2D convolution operator, and $\sigma(\cdot)$ is the non-linear activation function. The matrix $B \in \mathbb{R}^{|\mathcal{E}| \times K}$ is the broadcast matrix, which broadcasts node features to its outgoing edges. Note that the matrix multiplication $BX^{(l)}$ results in a shape of $|\mathcal{E}| \times h \times w$, whereas originally the dimension of $X^{(l)}$ is $K \times h \times w$. In other words, the operation $BX^{(l)}$ simply prepares the input along each edge for the following channel-wise convolution, $(BX^{(l)})\tilde{\circledast}F^{(l)}$. Finally, the matrix $\hat{A} \in \mathbb{R}^{K \times |\mathcal{E}|}$ is the aggregation matrix, which harvests all the information from the incoming edges to the graph nodes.

It is worth pointing out that, in Eq. (5.2), only $F^{(l)}$ is the learnable parameter, while the broadcast matrix B and the aggregation matrix $\hat{A}$ are both determined and constructed from the graph's adjacency matrix A by Algorithm 1. In Algorithm 1, we assume the input adjacency matrix A is already included with self connections.

5.3.3 SiaPose and its training procedure

With SIA-GCN, we propose the SiaPose for 2D hand pose estimation, as in Fig. 5.1. The preliminary pose estimator could be any off-the-shelf 2D hand pose estimator. Multiple heads of SIA-GCN would benefit capturing different positional informations due to different hand shapes in the input images. Assume there are M heads in the multi-head SIA-GCNs,

Algorithm 1 Broadcast and Aggregation Matrices Construction

procedure ConstructMatrices(A) ▷ Input A is the adjacency matrix
  Find the number of directed edges, $|\mathcal{E}|$, from A
  Find the number of nodes, K, from A
  Initialize ▷ Initialization for B and $\hat{A}$
    B as a zero matrix of size $|\mathcal{E}| \times K$
    $\hat{A}$ as a zero matrix of size $K \times |\mathcal{E}|$
    e as a zero vector of size $|\mathcal{E}|$
    $m = 1$
  for i in $1, 2, \cdots, K$ **do** ▷ Calculate for B
    for j in $1, 2, \cdots, K$ **do**
      if $A_{j,i} == 1$ **then** ▷ If j is the starting node of edge m
        $B_{m,j} = 1$
        $e[m] = i$ ▷ Record the end node of edge m
        $m = m + 1$
  for m in $1, 2, \cdots, |\mathcal{E}|$ **do** ▷ Calculate for $\hat{A}$
    $\hat{A}_{e[m],m} = 1$
  Construct the diagonal degree matrix D, with $D_{ii} = \sum_j \hat{A}_{ij}$.
  Set $\hat{A} = D^{-1}\hat{A}$ ▷ Normalize $\hat{A}$
  return B, $\hat{A}$

then, we could denote the output of the multi-head SIA-GCNs as $Y \in \mathbb{R}^{M \times K \times h \times w}$ and the output at the m^{th} SIA-GCN as $Y_m \in \mathbb{R}^{K \times h \times w}$. The pointer network, whose input is the image, is a regression network which generate a soft pointer vector $w \in \mathbb{R}^M$. The weight vector w actually indicates the importance of the information generated at different heads. Finally, at the information fusion stage, the aggregated confidence map is given by

$$\bar{Y} = w \cdot Y = \sum_{m=1}^{M} w_m Y_m, \tag{5.3}$$

which is a weighted sum of Y_m. The final predictions of the keypoint positions are obtained by taking the argmax of $\bar{Y}$.

The training procedure of the SiaPose is simple and could be conducted in an end-to-end

fashion. The total loss function is defined as

$$L = \alpha L_1 + L_2 = \alpha \sum_{t=1}^{T} \sum_{k=1}^{K} \left\| S_k^t - Y_k^* \right\|_F^2 + \sum_{k=1}^{K} \left\| \bar{Y}_k - Y_k^* \right\|_F^2 . \tag{5.4}$$

The first loss L_1 is responsible for the output of the preliminary pose estimator, while the second loss L_2 is added at the final output. The preliminary pose estimator itself (e.g. CPM and Stacked Hourglass) might consist of T multiple stages. The term $S_k^t \in \mathbb{R}^{h \times w}$ is the confidence map of k^{th} keypoint generated by the t^{th} stage of the preliminary pose estimator, while $\bar{Y}$ is the final confidence output of the SiaPose as in Eq.(5.3). Besides, $Y_k^* \in \mathbb{R}^{h \times w}$ is the ground truth confidence map of k^{th} keypoint, created by placing a Gaussian peak at its ground truth position. The coefficient α serves as a balancing weight between the two loss functions.

5.4 Experiments

Datasets. We evaluate our proposed method on three public hand pose datasets, the CMU Panoptic Hand Dataset (Panoptic) [96], the MPII+NZSL Hand Dataset [96] and the Large-scale Multiview 3D Hand Pose Dataset (MHP) [30]. For Panoptic (˜15k images) and MHP (˜82k images), we follow the setting of [54] and randomly split all samples into training set (70%), validation set (15%) and test set (15%). Since our contribution mainly focus on pose estimation instead of detection, we crop square image patches of annotated hands off the original images. A square bounding box which is 2.2 times the size of the hand is applied for cropping as in [96, 54, 53].

Evaluation metrics. The Probability of Correct Keypoint (PCK) [96] is utilized as our evaluation metric. In this paper, we use normalized threshold with respect to the size of square bounding box. We report the performance under different thresholds, $\delta = \{0.01, 0.02,$

0.03, 0.04, 0.05, 0.06}, and also their average (mPCK). More formally, for a single cropped input image of size $s \times s$, the PCK at δ can be defined as

$$\text{PCK}(\delta) = N(\delta)/K, \tag{5.5}$$

where $N(\delta)$ is the number of predicted keypoints which are within an interval threshold $\delta \cdot s$ of its correct location and K is the total number of keypoints.

Implementation details. In the experiments, two baselines, i.e., six-staged Convolutional Pose Machine (CPM) as in [96] and eight-staged Stacked Hourglass (SHG) are used as preliminary pose estimators in our SiaPose. For the SIA-GCN, we use 5 edge-aware graph convolutional layers defined in Eq. (5.2), which adopts a tree structured graph according to the kinematic structure of the hand skeleton, adding self connections. The size of the convolutional kernels in Eq. (5.2) is set to 45. ResNet-18 is used as the backbone of the pointer network. The input image is resized to 368×368 and 256×256 for the cases of CPM and SHG, respectively. Images are then scaled to [0,1], and normalized with mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225). We use Adam as our optimizer. For SHG-based SiaPose, the initial learning rate is set to 7.5e-4 while for the CPM-based SiaPose, we set it to 1e-4. For both cases, we train the model for 100 epochs, with learning rate reduced by a factor of 0.5 at milestones of the 60-th and 80-th epoch. The weight coefficient α in loss function Eq. (5.4) is set to drop from 1.0 to 0.1 at the 40th epoch.

Comparison with baselines. In Table 5.1 and Table 5.2, we compare the performance of our SiaPose with two baselines, CPM and SHG. (1) First, we conduct an experiment where *edge-unaware* GCN is utilized, where a shared weight matrix is used for all the edges. Interestingly, it performs worse than the baseline models. This is reasonable, because it's not appropriate to assume that relative positions of neighboring keypoints are always the same.

For example, index finger and thumb naturally have bones with different lengths. (2) Then we conduct experiments with our *edge-aware* SIA-GCNs, where different numbers of heads are explored. The results demonstrate that our proposed SiaPose could consistently improve both baselines noticeably. The ablative study on different numbers of heads validates the benefit of multi-heads and the effectiveness of the proposed SIA-GCN. For SHG, there is a 2.12 percent improvement at threshold $\delta = 0.01$ and for CPM, a 1.95 percent improvement is seen at threshold $\delta = 0.04$. (3) Also, inspired by the state-of-the-art algorithm [54], by adding a rotation network into our SiaPose (R-SiaPose) and using a similar training strategy, the performance of our method is further boosted, leading to significant improvements from baselines. Improvements of about 5 percent for SHG and nearly 4 percent for CPM are observed. We would also compare our model with that proposed in [54] in next subsection.

Table 5.1: SHG based SiaPose on Panoptic Dataset.

PCK@	0.01	0.02	0.03	0.04	0.05	0.06	mPCK
SHG Baseline	35.85	71.47	83.15	88.21	91.10	92.92	77.12
SharedWeight GCN	34.76	69.66	81.33	86.19	89.14	90.95	75.34
1-head SiaPose	35.78	71.16	83.57	88.98	92.00	93.84	77.55
5-head SiaPose	37.53	73.07	84.60	89.51	92.14	93.85	78.45
10-head SiaPose	37.97	73.53	84.95	89.70	92.26	93.91	78.72
Improvement	2.12	2.06	1.80	1.49	1.16	0.99	1.60
10-head R-SiaPose	39.46	77.22	88.45	92.97	94.85	96.09	81.48
Improvement	3.61	5.75	5.30	4.76	3.75	3.17	4.36

Table 5.2: CPM based SiaPose on Panoptic Dataset.

PCK@	0.01	0.02	0.03	0.04	0.05	0.06	mPCK
CPM Baseline	25.73	62.77	77.80	84.35	88.11	90.57	71.55
SharedWeight GCN	25.14	61.76	77.13	83.60	86.97	89.20	70.63
1-head SiaPose	25.90	63.36	78.98	85.69	89.44	91.90	72.55
5-head SiaPose	26.36	64.05	79.11	85.74	89.38	91.78	72.74
10-head SiaPose	26.45	64.19	79.67	86.30	89.83	92.20	73.11
Improvement	0.72	1.42	1.87	1.95	1.72	1.63	1.56
10-head R-SiaPose	26.62	65.80	81.60	88.02	91.39	93.36	74.47
Improvement	0.89	3.03	3.80	3.67	3.28	2.79	2.92

Comparison with state-of-the-art methods. We further compare our approach with

the current state-of-the-art methods [54, 53]. Probabilistic graphical models are deployed in [54] and [53], where the output confidence maps from CPM are utilized as unary potential functions. The CPM used in [54] and [53] is the version where 7×7 convolutional kernels are replaced by three 3×3 convolutional kernels. To make fair comparison, we follow their configurations and use their version of CPM as our preliminary pose estimator. The fundamental difference between our method and [54] is that we have adopted our SIA-GCN instead of graphical models. As observed from Table 5.3, our method outperforms both [54, 53] on the Panoptic dataset. On the MHP dataset, our SiaPose also achieves the state-of-the-art level performance. The size of the MHP dataset is about five times the size of the Panoptic, making the MHP dataset an easier task and allows less room for improvement. Methods focused on modeling structural relationships between keypoints would benefit more from smaller and challenging datasets that require models to extrapolate beyond pose templates seen in the training data.

Complexity analysis. Regarding the size of the proposed models, the 5-head and 10-head models increase the model size by about 30% and 40%, respectively, compared to the 1-head model. The increment of the model size from 1-head to multiple heads is primarily due to the added pointer network, which is drawn in Fig. 5.1. However, going from 5-head to 10-head does not significantly increase model complexity. This is because the pointer network only needs to output 5 more scalers and the overall overhead mostly comes from adding more GCN layers, which are shallow and not associated with too many parameters (note that we use "channel-wise" 2D convolutions). It's also worth to point out that, using a 10-head SIA-GCN, our model is about 80% and 60% the size of those in [53] and [54], respectively.

Domain generalization of our model. Table 5.4 demonstrates the domain generalization ability of our model. All the models in Table 5.4 are pretrained on Panoptic dataset, and then finetuned for about 40 epochs on the MPII+NZSL dataset. Consistent improvements over baselines are seen for all the ranges of PCK thresholds.

Qualitative results. Some qualitative examples are given in Fig. 5.3, which indeed shows that the SIA-GCN helps to enhance structural consistency and alleviate the spatial ambiguity. For example, in the third column, although the right hand is partially occluded by the earphone, our model could still correctly predict the position of all keypoints. We also show some failure cases of our model in Fig. 5.4, which are due to very heavy occlusion and foreshortened view of a fist.

Table 5.3: Comparison to state-of-the-art methods.

PCK@	0.01	0.02	0.03	0.04	0.05	0.06	mPCK
CMU Panoptic Hand Dataset							
R-MGMN [54]	23.67	60.12	76.28	83.14	86.91	89.47	69.93
AGMN [53]	23.90	60.26	76.21	83.70	87.72	90.27	70.34
R-SiaPose (Ours)	24.94	62.08	77.83	84.91	88.78	91.34	71.65
Large-scale Multiview 3D Hand Pose Dataset (MHP)							
R-MGMN [54]	41.51	85.97	93.71	96.33	97.51	98.17	85.53
AGMN [53]	41.38	85.67	93.96	96.61	97.77	98.42	85.63
R-SiaPose (Ours)	41.27	85.89	93.82	96.43	97.61	98.29	85.56

Table 5.4: Domain generalization of our model to MPII+NZSL from Panoptic Dataset.

PCK@	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08
CPM	8.05	23.78	37.74	48.00	55.65	61.68	66.58	70.82
R-SiaPose (Ours)	8.40	24.71	39.33	50.31	59.04	66.01	71.29	75.63
Improvement	0.35	0.93	1.59	2.31	3.39	4.33	4.71	4.81
SHG	11.72	30.85	44.82	54.71	62.35	68.48	73.47	77.61
R-SiaPose (Ours)	12.19	33.34	49.13	59.86	67.83	73.69	78.26	81.72
Improvement	0.47	2.49	4.31	5.15	5.48	5.21	4.79	4.11

Figure 5.3: Qualitative results of baseline (top) and our model (bottom) on Panoptic and MPII.

Figure 5.4: Failure cases of our model. Each pair contains an input image and its prediction.

5.5 Conclusion

In this chapter, we propose a novel spatial information aware graph neural network with 2D convolutions (SIA-GCN), which has the advantage of processing 2D spatial features for each node, with additional capability of learning different spatial relationships for different pair of neighboring nodes. We show the efficacy of our SIA-GCN in the 2D hand pose estimation task, by implementing a network which achieves the state-of-the-art performance. The SIA-GCN has the potential to generalise well to other tasks.

Chapter 6

Identity-Aware Hand Mesh Estimation and Personalization from RGB Images

6.1 Introduction

Reconstructing 3D hand meshes from monocular RGB images has attracted increasing amount of attention due to its enormous potential applications in the field of AR/VR. Most state-of-the-art methods attempt to tackle this task in an anonymous manner. Specifically, the identity of the subject is ignored even though it is practically available in real applications where the user is unchanged in a continuous recording session. In this paper, we propose an *identity-aware* hand mesh estimation model, which can incorporate the identity information represented by the intrinsic shape parameters of the subject. We demonstrate the importance of the identity information by comparing the proposed *identity-aware* model to a baseline which treats subject anonymously. Furthermore, to handle the use case where

the test subject is *unseen*, we propose a novel personalization pipeline to calibrate the intrinsic shape parameters using only a few unlabeled RGB images of the subject. Experiments on two large scale public datasets validate the state-of-the-art performance of our proposed method.

Hand pose estimation has been one of the most popular computer vision problems because of its critical role in many applications, including hand gesture recognition, virtual and augmented reality, sign language translation and human-computer interaction [5]. With recent advances in deep learning techniques [41, 97, 75] and development of large hand pose datasets [96, 133, 141, 140], 2D hand pose estimation has been extensively investigated and deployed in real-time applications with compelling results [10, 53, 96]. However, 3D hand pose estimation still remains a challenging problem due to the diversity of hand shapes, occlusion and depth ambiguity when monocular RGB image is used.

Current state-of-the-art methods for 3D hand reconstruction from RGB images either try to directly regress 3D vertices of the hand mesh [32, 58, 18, 61, 62], or utilize the parametric MANO model [90] by regressing the low-dimensional parameters [6, 135, 4, 39, 131]. While these methods could generalize reasonably across different subjects, nearly all of them estimate the 3D hand pose in an anonymous manner. The identity information of the subject, which is practically available in real applications, is typically ignored in these methods. In many real-world use cases, such as virtual and augmented reality, the device is often personal and the user is typically identifiable.

We ask the question, can 3D hand reconstruction from RGB images be further improved with the help of identity information? If so, how should we calibrate the personalized hand model for *unseen* subjects during the test phase, using only RGB images? In depth image based hand tracking systems, the hand model personalization has been well studied and its benefits on improving hand tracking performance has been demonstrated [105, 106]. However, using only RGB images to perform personalization is underexplored.

To close this gap and answer the above question, we investigate the problem of hand model personalization from RGB images and design a simple yet effective network to incorporate the identity information. Specifically, we propose an *identity-aware* hand mesh estimation model, which can take in the personalized hand model along with the input RGB image. Motivated by MANO [90], we choose to use MANO shape parameters to represent the hand model. To enable a fair comparison, we then construct a strong baseline by adapting our proposed identity-aware network slightly. Instead of being given the groundtruth hand shape parameters, the baseline regresses the shape parameters directly from the input image via a multi-layer perceptron. We show through experiments that with ground truth shape parameters, more accurate 3D hand reconstruction can be obtained. Lastly, we propose a novel personalization method which can calibrate the hand model for *unseen* subjects, using only unannotated RGB images. The calibrated hand model can then be utilized in our identity-aware network. Our main contributions are summarized as follows:

- Our work is the first to systematically investigate the problem of hand mesh personalization from RGB images and demonstrate its benefits to hand mesh and keypoints reconstruction.
- For unknown subjects that are *not seen* in training, we develop a novel hand model personalization method that is capable of calibrating the hand model using a few unannotated images of the same subject.
- We demonstrate that our method outperforms existing methods on two large-scale public datasets, showing the benefit of utilizing the identity information, which is an underexplored topic in the field.
- We design a simple but competitive baseline that features the same optimization augmented inference step and further validate the effectiveness of leveraging the identity information.

6.2 Related Work

There are many research works on human/hand pose estimation [123, 121, 120, 61, 62, 70, 69, 66, 117, 15, 14, 16, 17, 67], including well-developed 2D hand pose estimation algorithms [116, 53, 10, 96, 54, 20, 55] and fast developing 3D hand pose estimation algorithms [2, 139, 34, 7, 100]. In this section, we will mainly discuss literature on 3D hand mesh reconstruction.

Model-based methods. The popular model-based method usually rely on the MANO model [90], developed from the SMPL human model [65]. As a parameterized model, the MANO model factorizes the hand mesh into shape and pose parameters, by utilizing principal component analysis. Massive literature has tried to predict the MANO parameters in order to reconstruct the hand mesh. Boukhayma *et al* [6] regressed the MANO shape and pose parameters from 2D keypoint heatmaps. This was the first end-to-end deep learning based method that can predict both 3D hand shape and pose from RGB images in the wild. Zhang *et al* [135] proposed to use an iterative regression module to regress the MANO parameters in a coarse-to-fine manner. Baek *et al* [4] also exploited iterative refinement. In addition to that, a differentiable renderer was also deployed, which can be supervised by 2D segmentation masks and 3D skeletons. Hasson *et al* [39] exploited the MANO model to solve the task of reconstructing hands and objects during manipulation. Yang *et al* [131] proposed a multi-stage bisected network, which can regress the MANO params using 3D heatmaps and depth map.

Model-free methods. In [70], Moon *et al* designed I2L-MeshNet, an image-to-lixel prediction network. Many other works are based on graph convolutional network, directly regressing the vertex locations. In [32], Ge *et al* proposed a graph neural network based method to reconstruct a full 3D mesh of hand surface. In [60], Lim *et al* proposed an efficient graph convolution, SpiralConv, to process mesh data in the spatial domain. Leveraging spiral mesh convolutions, Kulon *et al* [58] devised a simple and effective network architec-

ture for monocular 3D hand pose estimation consisting of an image encoder followed by a mesh decoder. Most recently, Chen *et al* [18] exploited the similar architecture, with more advanced designs. They divide the camera-space mesh recovery into two sub-tasks, i.e., root-relative mesh recovery and root recovery. To estimate the root-relative mesh, the authors proposed a novel aggregation method to collect effective 2D cues from the image, and then are decoded by a spiral graph convolutional decoder to regress the vertex coordinates. Apart from graph neural network, Transformer [110] has also been introduced into the field of computer vison, solving different tasks [64, 11, 130]. Several methods [61, 62, 37, 80] have been proposed for hand pose and mesh reconstruction.

Hand model personalization. Tan *et al* [105] and Tkach *et al* [106] studied hand model personalization in the scenario where multiple *depth* images are available, and successfully demonstrated its importance in hand tracking. Hampali *et al* [36] used the same method to generate annotations when creating a new dataset. However, hand model personlization from RGB images has been underexplored. Qian *et al* [86] focused on hand *texture* personalization from RGB images. While hand model (mesh) personalization is also performed, the effectiveness of mesh personalization is not validated by quantitative results. There is also no investigation on whether the personalized mesh model can be used to improve hand pose estimation. Moon *et al* [71] proposed to personalize each subject using a randomly generated Gaussian vector. The subject ID vectors were generated prior to training and experiments were performed where all subjects in the test set were already seen in the training set. The trained model is only applicable to known subjects and there exists no principle way to handle unseen subjects during the testing phase. MEgATrack [38] is a multi-view monochrome egocentric hand tracking system that calibrates the hand model for unseen users, but the calibration is limited to a single hand scaling factor.

To our best knowledge, our work is the *first* to systematically investigate the hand model personalization from RGB images and its benefits to 3D hand pose estimation and mesh

reconstruction.

6.3 Method

We first review the MANO hand model which is used extensively in this work, and then propose our identity-aware hand mesh estimation method that takes as input the identity information represented by the hand MANO shape parameters along with the input image. Next, by a slight modification of our method, we propose the baseline which would be compared with. Lastly, to address the practical use case where the hand model is not provided for the test subject, we propose a novel personalization pipeline that estimates the hand model for an *unseen* subject using only a few unannotated images.

6.3.1 MANO Model

MANO [90] is a popular parameterized hand model extended from the 3D human model SMPL [65]. The MANO model factorizes the hand mesh into two groups of parameters: the shape parameters and the pose parameters. The shape parameters control the intrinsic shape of the hand, e.g., size of the hand, thickness of the fingers, length of the bones, etc. The pose parameters represent the hand pose, i.e., how the hand joints are transformed, which subsequently deforms the hand mesh. Mathematically, the model is defined as below:

$$\mathcal{M}(\beta, \theta) = W(T_P(\beta, \theta), J(\beta), \theta, \mathcal{W}) \tag{6.1}$$

where a skinning function W is applied to an articulated mesh with shape T_P, joint locations J, pose parameter θ, shape parameter β, and blend weights $\mathcal{W}$ [90].

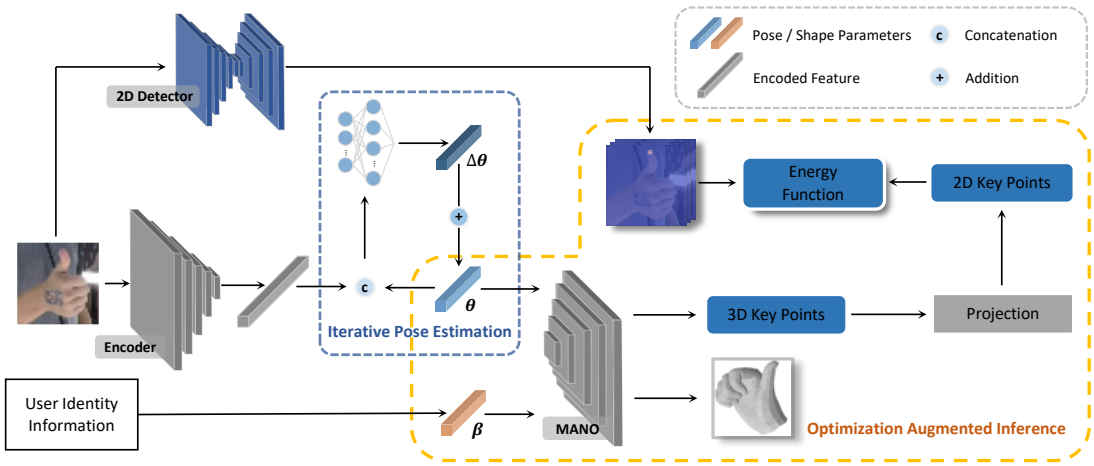


Figure 6.1: Overview of our proposed identity-aware hand mesh estimation model. The model mainly contains three parts, i.e., the iterative pose regressor, the 2D detector and the optimization module. Note that in our proposed model, along with the RGB image, we also feed the user's identity information, i.e. the ground truth or calibrated MANO shape parameters of the user.

6.3.2 Identity-aware Hand Mesh Estimation

Existing methods assume that the subject in every image frame is anonymous, even though the input is recorded in a continuous session. To fully leverage the fact that the subject is often fixed within each recording session in real applications, we propose a new hand mesh estimation pipeline. In addition to the input image, we also feed the user's identity information into the network.

There are various ways to represent the identity of a subject. The most straightforward method is to label each subject with a unique identifier, such as a high-dimensional random vector [71]. However, identity information that does not have physical meaning can be hard for the model to utilize. More importantly, models trained with this type of identity information usually only generalize to *known* subjects included in the training set. In this work, we are interested in an identity representation that allows generalizing to *unseen* subjects.

Inspired by MANO model [90], we utilize the MANO shape parameters as the identity information for a specific subject. As shown in Fig. 6.1, our proposed identity-aware hand mesh estimator takes in directly the ground truth or calibrated MANO shape parameters, enabling the network to be subject-aware. The main parts of our proposed model is explained as follows.

MANO pose parameter regressor. Motivated by [135], the pose parameter θ is obtained by using an iterative pose regressor. We include the global rotation in θ, and use the 6D rotation representation [138] to represent the rotation of each joint. With 15 hand joints and the global rotation, θ is a vector in $\mathbb{R}^{96}$. Let $\mathcal{F} \in \mathbb{R}^N$ denote the image feature after the encoder and $\theta^{(i)}$ denote the estimated pose after i iterations. Initially, we set $\theta^{(0)}$ as the rotation 6D representation of identity matrices. Then, the pose is predicted iteratively as follows

$$\Delta\theta^{(i)} = MLP_\theta\left(\text{cat}(\mathcal{F}, \theta^{(i-1)})\right) \tag{6.2}$$

$$\theta^{(i)} = \Delta\theta^{(i)} \oplus \theta^{(i-1)}, \tag{6.3}$$

where $\oplus$ means adding the new rotation increment onto the predicted rotation from the previous iteration. The operator $\oplus$ is implemented by transforming both $\Delta\theta^{(i)}$ and $\theta^{(i-1)}$ from rotation 6D representations to rotation matrices, then multiplying them, and finally converting the result back to rotation 6D representation. We adopt three iterations in the experiments.

Optimization Augmented Inference. During inference time, we can further improve the estimated hand mesh by enforcing the consistency between the 3D pose and the 2D pose predictions. The 2D predictions are obtained via a stacked hourglass-style neural network [75].

Let $\mathbf{x}^{\text{d}} \in \mathbb{R}^{21\times 2}$ denote the 2D keypoints predictions, $f_{\text{MANO}}(\cdot)$ represent the mapping function from (β, θ) to 3D keypoints positions, $\mathcal{P}(\cdot)$ denote the projection operator from 3D

space to image space, and $\mathbf{r} \in \mathbb{R}^{\mathbf{3}}$ denote the root-position of the hand. We aim to optimize the following energy function

$$\mathcal{E}(\theta, \mathbf{r}, \beta) = \left\| \mathbf{x}^{\mathrm{d}} - \mathcal{P}(f_{\mathrm{MANO}}(\beta, \theta) + \mathbf{r}) \right\|_2 . \tag{6.4}$$

We adopt a two-stage optimization procedure. In the first stage, we optimize $\mathbf{r}$ only. In the second stage, we optimize θ and $\mathbf{r}$ jointly. Note that the MANO parameters are not optimized from scratch. The prediction from the MANO parameter regressor is used as the initial guess.

6.3.3 Baseline Method

To further validate that the accuracy improvement of hand mesh reconstruction is a result of leveraging the identity information, we construct a baseline by slightly modifying our identity-aware model. Instead of feeding ground truth/calibrated shape parameters into the model, we use an extra MLP to regress the shape parameters from the input image. For a fair comparison, all the other modules from our identity-aware model are kept the same in this baseline model. Formally, let $\mathcal{F} \in \mathbb{R}^N$ denote the image feature produced by the encoder. The MANO shape parameter $\beta \in \mathbb{R}^{10}$ is directly regressed by a multilayer perceptron from $\mathcal{F}$ as

$$\beta = MLP_{\beta}(\mathcal{F}). \tag{6.5}$$

6.3.4 Personalization Pipeline

In most practical applications, the test subject is usually *unknown* and there is no corresponding hand model (shape parameters) available for the proposed identity-aware hand mesh estimation pipeline. To handle this practical issue, we propose a novel hand model

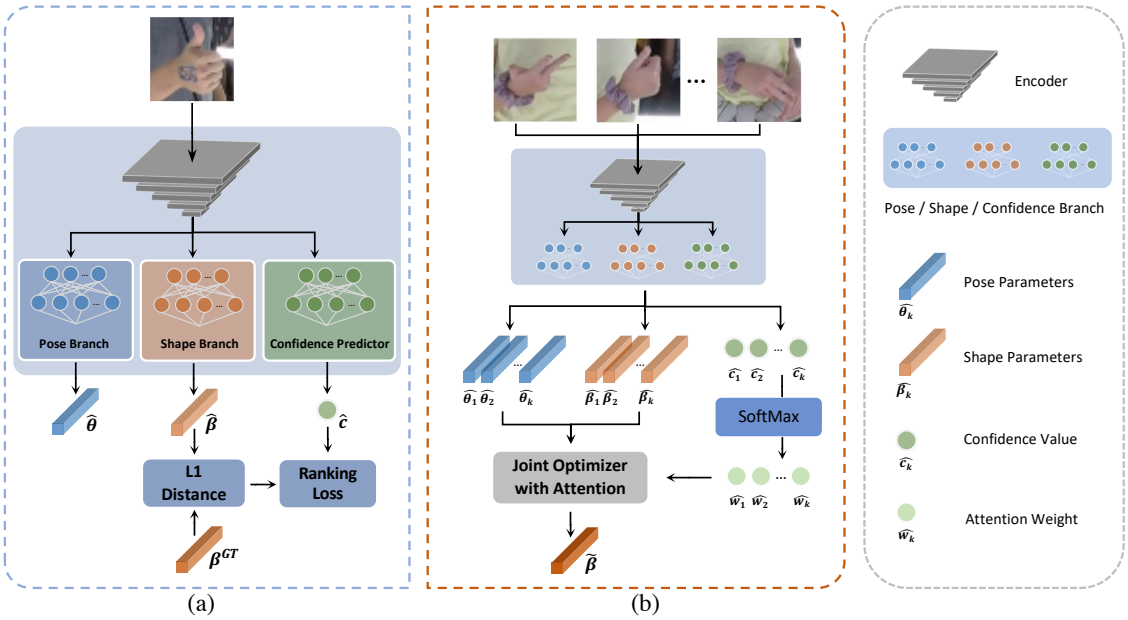


Figure 6.2: Proposed personalization pipeline with attention mechanism. Images used for personalization capture the same subject who is never seen during training.

personalization method, which could calibrate the hand model from a few *unannotated* RGB images.

Confidence Predictor.

Our personalization pipeline takes in multiple images of a same subject and perform a joint attention-based optimization to get the personalized shape parameter. Naively, the images can be treated equally and contribute the same weight during the optimization. However, images usually differ from each other in terms of quality, view angles, occlusions and so on. Thus, the images should be attended with different importance. To achieve this goal, we propose a light weight confidence predictor on top of the baseline network, as shown in Fig. 6.2 (a). The confidence predictor takes as input the feature extracted by the ResNet50 encoder and outputs a scalar via one fully connected layer. The predicted confidence value indicates the quality of the predicted shape parameter from the input image. Note that our confidence predictor is only trained on the training split. Subjects in the test split are

different from the training split and are **not seen** during the training phase.

Joint Optimization with Attention.

Fig. 6.2 (b) illustrates the whole process during the personalization phase. Denote the collection of K unannotated images from the same user as $\mathcal{I} = \{I_1, I_2, \cdots, I_K\}$. The images are fed into the baseline model equipped with confidence predictor, which outputs $\{c_i, \hat{\beta}_i, \hat{\theta}_i\}$ for each image I_i, where $c_i \in \mathbb{R}$ is the confidence value, $\hat{\beta}_i, \hat{\theta}_i$ are the predicted MANO shape and pose parameters. The confidence values $\{c_i\}_{i=1}^K$ then go through a SoftMax layer, which generates the attention weights $\{w_i\}_{i=1}^K$ as following

$$w_i = \frac{e^{c_i/T}}{\sum_{k=1}^K e^{c_k/T}}, \tag{6.6}$$

where T is the temperature parameter. Afterwards, $\{w_i, \hat{\beta}_i, \hat{\theta}_i\}_{i=1}^K$ are sent into the attention based optimization module, where the following optimization is solved

$$\min_{\tilde{\beta}} \sum_{k=1}^{K} w_k \cdot \|\mathcal{M}(\tilde{\beta}, \hat{\theta}_k) - \mathcal{M}(\hat{\beta}_k, \hat{\theta}_k)\|_F, \tag{6.7}$$

where $\mathcal{M}(\cdot)$ is the MANO model. Note that, now all the K images from the same subject share same shape parameter $\tilde{\beta}$. After the personalization process, $\tilde{\beta}$ would be used as the identity information for the subject.

6.3.5 Loss Functions

The baseline. To train the baseline, we apply loss terms on the predicted 3D hand mesh, following [18], and also on the predicted MANO shape and pose parameters.

a) Loss functions on hand mesh. Denote the vertices and faces of the hand mesh as $\mathcal{V}$ and

Ω. We impose $L1$ loss on the predicted hand mesh, and also deploy edge length loss and normal loss, following [18]. The loss functions on the mesh can be expressed as

$$
\begin{aligned}
L_{\text{mesh}} &= \sum_{i=1}^{N} \|\hat{\mathcal{V}}_i - \mathcal{V}_i\|_1 \\
L_{\text{norm}} &= \sum_{\omega \in \Omega} \sum_{(i,j) \subset \omega} \left| \frac{\hat{\mathcal{V}}_i - \hat{\mathcal{V}}_j}{\|\hat{\mathcal{V}}_i - \hat{\mathcal{V}}_j\|_2} \cdot \mathbf{n}_\omega \right| \\
L_{\text{edge}} &= \sum_{\omega \in \Omega} \sum_{(i,j) \subset \omega} \left| \|\hat{\mathcal{V}}_i - \hat{\mathcal{V}}_j\|_2 - \|\mathcal{V}_i - \mathcal{V}_j\|_2 \right|,
\end{aligned}
\tag{6.8}
$$

where the $\mathbf{n}_\omega$ is the unit normal vector of face $\omega \in \Omega$.

b) Loss function on MANO parameters. We use $L_{\text{pose}} = \|\hat{\theta} - \theta\|_1$ and $L_{\text{shape}} = \|\hat{\beta} - \beta\|_1$, where θ and β are ground truth MANO pose and shape parameters. The $\hat{\theta}$ is the predicted pose parameter from the last iteration of the iterative pose regressor.

c) Loss function on 2D heatmap. A binary cross entropy function is imposed on 2D heatmaps of hand keypoints as in $L_{\text{pose}_{\text{2D}}} = \text{BCE}(\hat{U}, U)$, where $\hat{U}$ and U are the predicted and ground truth 2D heatmaps of each keypoint, respectively. The groud-truth heatmap U is generated with a Gaussian distribution.

The 2D detector is trained by using $L_{\text{pose}_{\text{2D}}}$. The other parts are trained under the following loss function

$$
L_{\text{total}} = L_{\text{mesh}} + 0.1 \cdot L_{\text{norm}} + L_{\text{edge}} + L_{\text{pose}} + L_{\text{shape}}. \tag{6.9}
$$

Our identity-aware model. Since for our identity-aware model, the subject identity information (the MANO shape parameter) is provided, either ground truth or calibrated, the loss function is given by Eq. (6.10) with the shape loss removed,

$$
L^{'}_{\text{total}} = L_{\text{mesh}} + 0.1 \cdot L_{\text{norm}} + L_{\text{edge}} + L_{\text{pose}}. \tag{6.10}
$$

Confidence Predictor. We use margin ranking loss for training of the confidence predictor. Given N_b images in the batch, the baseline model equipped with confidence predictor would output confidence values $\{c_i\}_{i=1}^{N_b}$ and MANO shape parameter predictions $\{\hat{\beta}_i\}_{i=1}^{N_b}$. With ground truth shape parameters $\{\beta_i\}_{i=1}^{N_b}$, the difference l_i between the predicted and ground truth shape parameters can be calculated as $l_i = |\beta_i - \hat{\beta}_i|_1$. We generate $N_b \times (N_b - 1)/2$ pairs of $\{(c_i, l_i), (c_j, l_j)\}$, and calculate ranking loss on each pair [84]. The total loss is the sum of ranking losses from all pairs.

6.4 Experiments

6.4.1 Experimental Setups

Datasets. We conduct experiments on two large-scale public hand pose datasets, i.e., HUMBI [133] and DexYCB [13]. There are two major reasons why these two datasets are chosen. First, they both have a diverse collection of subjects, which allows us to split the datasets into different subject groups for training and evaluating our identity-aware pipeline. More importantly, they annotate the shape parameters of the same subject in a consistent way. Each hand image in the dataset is associated with a subject ID and all the hands from the same subject share the same MANO shape annotation. Note that our method cannot be directly evaluated on other popular benchmarks such as FreiHAND [141] or InterHand [72] because they either do not associate images with subject IDs or guarantee consistent shape parameters for the same subject.

HUMBI is a large multiview image dastaset of human body expressions with natural clothing. For each hand image, the 3D mesh annotation is provided, along with the fitted MANO parameters. The shape parameters are fitted across all instances of the same subject. This means that the same shape parameters are *shared* among all the hand meshes from the same

subject. In our experiments, we use all the right hand images from the released dataset. We split the dataset into training (90%) and test (10%), by subjects. The split results into 269 subjects (474,472 images) in the training set and 30 subjects (50,894 images) in the test set. Note that none of the subjects in the test set appear in the training set.

DexYCB is a large dataset capturing hand grasping of objects. The dataset consists of 582K RGB-D frames over 1,000 sequences of 10 subjects from 8 views. It also provides MANO parameters for each hand image. Same as the HUMBI dataset, the hand shape parameters for each subject are calibrated and fixed throughout each subject's sequences. While object pose estimation is beyond the scope of this work, extra occlusions introduced by the objects makes the DexYCB dataset more challenging for hand mesh estimation. In our experiments, similar to the set up for the HUMBI dataset, we use the provided split in [13] which splits the dataset by subjects. In this set up, there are 7, 1, 2 subjects in the training, validation and test set, respectively.

Metrics for 3D Hand Estimation. Following the protocol used by existing methods, we use the following two metrics, both in millimeter.

a) *Mean Per Joint Position Error* (MPJPE) measures the Euclidean distance between the root-relative prediction and ground truth 3D hand keypoints.

b) *Mean Per Vertex Position Error* (MPVPE) measures the Euclidean distance between the root-relative prediction and groud-truth 3D hand mesh.

Metrics for Hand Shape Calibration. We propose three metrics to evaluate the performance of the calibrated hand shape.

a) MSE_{mano} measures the mean square error between the estimated MANO shape parameters and the ground truth values.

b) *W-error* measures the mean hand width error between the calibrated hands and the ground truth hands at the flat pose, which is defined as the distance between the metacarpophalangeal joints of index finger and ring finger.

b) *L-error* measures the mean hand length error between the calibrated hands and the ground truth hands at the flat pose, which is defined as the distance between the wrist joint and the tip of middle finger as the hand length.

Implementation Details. We implement our model in PyTorch [81] and deploy ResNet50 [41] as our encoder. Input images are resized to 224×224 before being fed into the network. We use the Adam optimizer [51] and a batch size of 32 to train all the models except for the confidence predictor. For a fair comparison, both the baseline model and our proposed identity-aware model are trained using the same learning rate schedule. On the HUMBI dataset, both models are trained for 15 epochs, with an initial learning rate of 1e-4 which is dropped by a factor of 10 at the 10-th epoch. On the DexYCB dataset, models are also trained for 15 epochs, with the same initial learning rate, while the learning rate is dropped at the 5-th and 10-th epochs. With the baseline model trained and frozen, the lightweight confidence predictor is trained with a batch size of 128, with the intuition that larger batch size allows more image pairs to train the ranking loss. The temperature parameter is set to 0.33 in Eq. (6.6). During all the training, input images are augmented with random color jitter and normalization. In the inference stage, we use the Adam optimizer in PyTorch to optimize Eq. (6.4). Specifically, 200 and 60 iterations are performed with learning rate of 1e-2 and 1e-3 in the first and second optimization stages, respectively. On one Titan RTX graphics card, it takes 8 minutes to process all test images (50k) in HUMBI dataset, and 7.5 minutes for those (48k) in DexYCB dataset. We emphasize again that all our experiments are conducted in the scenarios where there is **no overlap** between the subjects in the test set and the training set.

Table 6.1: Numerical results on DexYCB and HUMBI datasets.

Method	DexYCB		HUMBI	
	MPJPE ↓	MPVPE ↓	MPJPE ↓	MPVPE ↓
CMR-PG [18]	20.34	19.88	11.64	11.37
Without Optimization at Inference Time				
Baseline	21.58	20.95	12.13	11.82
Ours, GT shape	18.83	18.27	11.41	11.11
Ours, Calibrated	18.97	18.42	11.51	11.21
With Optimization at Inference Time				
Baseline	18.03	17.92	10.75	10.60
Ours, GT shape	16.60	16.29	10.17	9.94
Ours, Calibrated	**16.81**	**16.55**	**10.31**	**10.28**

Table 6.2: Comparison with existing methods on DexYCB.

Methods	MPJPE↓	MPVPE ↓
Boukhayma et al. [6]	27.94	27.28
Spurr *et al* [100] + ResNet50	22.71	-
Spurr *et al* [100] + HRNet32	22.26	-
Boukhayma et al. [6] †	21.20	21.56
CMR-PG [18]	20.34	19.88
Metro [61]	19.05	17.71
Ours, Calibrated	**16.81**	**16.55**

Table 6.3: Performance of hand model calibration.

Metrics	HUMBI	DexYCB
MSE_{mano}	0.07	0.04
W-error (mm)	0.88	1.02
L-error (mm)	1.71	1.20

6.4.2 Quantitative Evaluation

3D Hand Estimation. We evaluate the benefit of our pipeline under two settings, i.e., with and without the optimization module during inference time. As shown in Table 6.1, our proposed pipeline improves the baseline consistently across different datasets. With calibrated hand model, our proposed method can achieve close performance to that with ground truth hand model, which validates the effectiveness of our personalization pipeline. Furthermore, our method also achieves the state-of-the-art performance, as shown by Table 6.1 and Table 6.2. To ensure fair comparison, same data augmentation are applied to all

the methods, i.e., random color jitter and normalization. All the models are trained for 15 epochs including ours, with the exception of Metro [61] which is trained for 70 epochs, as transformers are much harder to converge. The superscript † in Table 6.2 means adding our optimization module on top of the original method. It shows that our optimization module can be generalized to other model-based methods efficiently. We emphasize that, *none* of the existing methods produces consistent shape estimation across images originating from the same subject. In contrast, our method guarantees shape consistency with zero hand shape variation.

Hand Shape Calibration. Table 6.3 reports the performance of our personalization pipeline, which achieves less than 2 mm in terms of hand width and hand length errors, by calibrating on 20 unannotated images.

Our proposed method inherently guarantees the hand shape consistency among different images from the same subject. Fig. 6.3 demonstrates this advantage of our method over the baseline model. As shown in Fig. 6.3, for a specific subject, the baseline model outputs hand meshes with big variations in terms of hand length, up to 20 mm. This is because the baseline model is subject-agnostic and predicts hand shape parameters based on a single input image. Even if the input images are from the same user, the baseline model could predict hand meshes with big variations in size. In contrast, our proposed method outputs consistent hand shape inherently, with *zero* hand shape variation across images from the same user. Also shown in Fig. 6.3, the hand size calibrated by our proposed method stays close to the ground truth hand size in most cases.

6.4.3 Ablation Study

Number of images used for personalization. Fig. 6.4 shows hand size errors (in mm) when different number of images are utilized during calibration. With $K = 20$ images, the

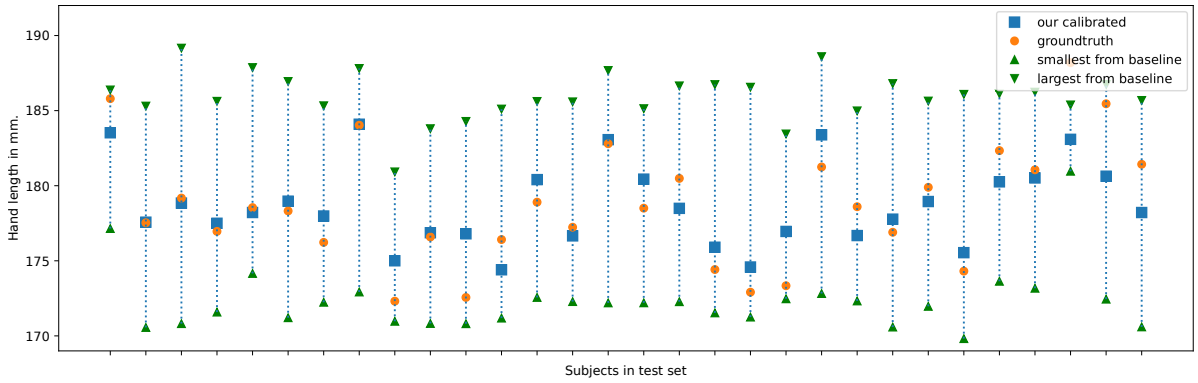


Figure 6.3: Hand shape consistency comparison between our proposed method and the baseline. The x-axis corresponds to different subjects in the test dataset, while the y-axis corresponds to the length of the hand of each subject.

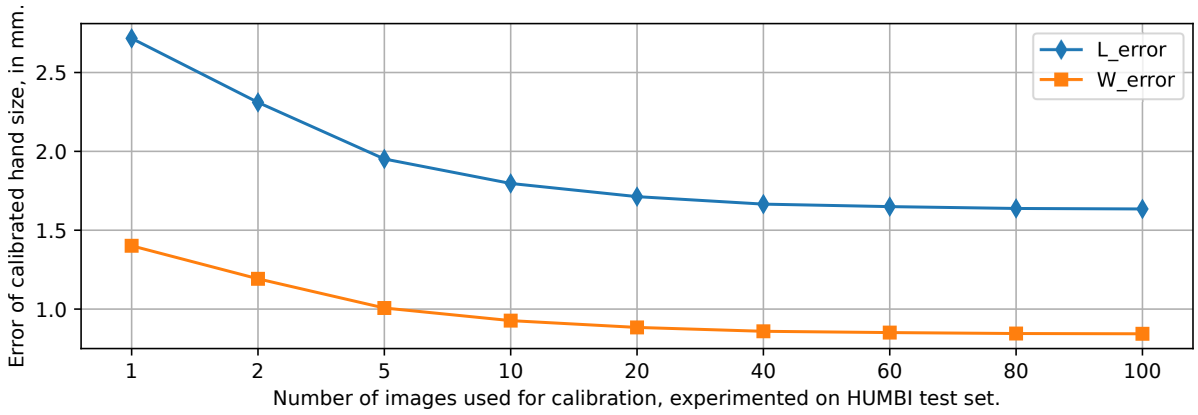


Figure 6.4: Impact of the number of images used in calibration.

hand model can already be well calibrated with length error less than 2 mm and width error less than 1 mm. In all the other experiments, we use $K = 20$ images for hand model calibration.

Attention during calibration. During the calibration, different weights are imposed across the input images according to their confidence values, as formulated in Eq. (6.7). We compare the calibration performance of our attention-based method with the non-attention method, as shown in Table. 6.4. Specifically, non-attention means to treat each image equally and set $w_i = 1/K$ in Eq. (6.7) for all images. As shown by Table. 6.4, our attention-based calibration can improve the performance by a noticeable margin comparing to the naive

Table 6.4: Effectiveness of confidence-valued based attention mechanism.

Metrics	MSE_{mano}	W-error (mm)	L-error (mm)
No attention	0.084	1.00	1.93
Ours, with attention	0.070	0.88	1.71
Improvement	16%	12%	11%

Table 6.5: Evaluating models trained with 3D keypoints instead of mesh supervision on DexYCB and HUMBI datasets.

Method	DexYCB		HUMBI	
	MPJPE↓	MPVPE ↓	MPJPE ↓	MPVPE ↓
Without Optimization at Inference Time				
Baseline	21.85	20.26	12.34	12.02
Ours, GT Shape	18.92	18.35	11.61	11.30
With Optimization at Inference Time				
Baseline	17.71	17.58	10.80	10.95
Ours, GT Shape	**16.63**	**16.32**	**10.37**	**10.12**

calibration.

Optimization augmented inference from scratch. In this experiment, we remove the MANO parameter regressor from the model in Fig. 6.1. Without being initialized by the MANO parameter regressor, the initial pose is set to the neutral pose prior to optimization. This pure optimization procedure results in an MJPJE $>$ 50mm on both DexYCB and HUMBI datasets. This validates the necessity of the MANO parameter regressor, which can give good initial values of MANO parameters for later optimization.

Training model with 3D keypoints instead of 3D mesh supervision. In Table. 6.5, we report the performance of the baseline and our proposed identity-aware model when trained with 3D keypoints supervision, instead of 3D mesh. Under this setting, our identity-aware method still improves the accuracy.

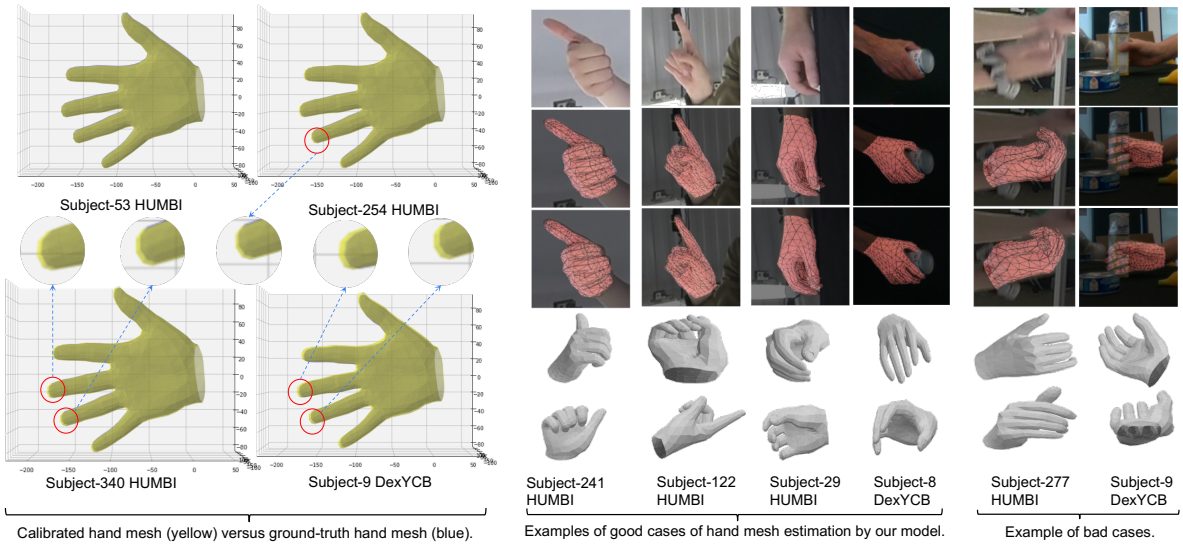


Figure 6.5: Qualitative results. a) Left: calibrated hand model versus ground truth hand model. b) Right: visualization of our identity-aware hand mesh estimator. From top row to bottom row are the input RGB images, the projected ground truth meshes, the projected predicted meshes, and the predicted meshes viewed from two different angles.

Qualitative Results. The qualitative results of our personalization method and the identity-aware hand mesh estimator are shown in Fig. 6.5. On the left side, it can be seen that the calibrated hand mesh is very close to the ground truth hand mesh. On the right side, qualitative results of our identity-aware model are demonstrated. When generating the third row, we align the predicted mesh with ground truth root position before projecting the mesh back to the image space. As seen from Fig. 6.5, our model can robustly recover the hand mesh under moderate occlusion and can handle a wide range of hand poses.

Limitations. The guarantee of consistent hand shape primarily comes from explicitly incorporating a 3D hand model i.e., the MANO in our pipeline. A future direction is to explore model free approaches to enforce shape consistency at inference time. We also observe that images with severe occlusions and blurs may affect the quality of shape calibration. We currently mitigate this issue by predicting confidence values, which helps lower the importance of these suboptimal images greatly. A better approach might be to detect and remove these

images prior to the calibration step.

6.5 Conclusion

In this chapter, we propose an *identity-aware* hand mesh estimation pipeline for 3D hand mesh recovery from monocular images. Different from existing methods which estimate the hand mesh anonymously, our method leverages the fact that the user is usually unchanged in real applications and identity information of the subject can be utilized for 3D hand mesh recovery. More specifically, our model not only takes as input the RGB image, but also the identity information represented by the intrinsic shape parameters of the subject. We also design a novel personalization pipeline, through which the intrinsic shape parameters of an *unknown* subject can be calibrated from a few RGB images. With the personalization pipeline, our model can operate in scenarios where ground truth hand shape parameters of subjects are not provided, which are common in real world AR/VR applications. We experimented on two large-scale public datasets, HUMBI and DexYCB, demonstrating the state-of-the-art performance of our proposed method.

Chapter 7

Conclusion

In this thesis, several algorithms are proposed to tackle the problem of human hand pose estimation and 3D mesh reconstruction from RGB images. To explicitly enforce the structural constraints among the hand keypoints, we firstly propose to utilize the probabilistic graphical models. On top of combining the graphical models with newly developed deep CNNs, we propose to make the graphical models adaptive to input images, either fully adaptive in the sense that each input image has its own graphical model or semi adaptive when images are clustered and a pool of a fixed number of graphical models are used. Further more, we also resort to graph neural networks, which has a relaxed constraints comparing to graphical models. A novel graph convolutional network has been proposed, where the spatial information along each edge of the graph is utilized. When tackling the more challenging hand mesh reconstruction problem, we propose to utilize the identity information of the subject, which enables the hand shape consistency among different images from the same person. Additionally, a personalization method is proposed for the use case in practical scenarios. Extensive experiments have been conducted to validate the efficacy of our proposed methods.

Future research. Apart from the progress discussed above, there are still many open problems that are very interesting for future research.

- Hand-object interaction. In the scenario of hand-object interaction, reconstructing the hand mesh and the object mesh together is much more challenging since additional factors have to be taken into consideration, for example, the penetration issue and the dedicate mesh deformation around the contact area.

- Generalization of the models. The ability of models to generalize to wild images is still somewhat limited. Improving the generalization ability of the models is key to applications in real world scenarios. This might be achieved by designing new pose estimation algorithms or new techniques of collecting hand data in the wild.

- More advanced hand models. Currently, MANO is the de facto state-of-the-art hand model which is widely used in the research field. With the popularity of neural fields and other new techniques like diffusion model, more advanced hand models may be proposed.

Bibliography

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. 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.

[2] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–432. IEEE, 2003.

[3] S. Baek, K. In Kim, and T.-K. Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8330–8339, 2018.

[4] S. Baek, K. I. Kim, and T.-K. Kim. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1067–1076, 2019.

[5] D. R. Beddiar, B. Nini, M. Sabokrou, and A. Hadid. Vision-based human activity recognition: a survey. *Multimedia Tools and Applications*, 79(41):30509–30555, 2020.

[6] A. Boukhayma, R. d. Bem, and P. H. Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10843–10852, 2019.

[7] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 666–682, 2018.

[8] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2272–2281, 2019.

[9] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018.

[10] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 43(1):172–186, 2019.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

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

[13] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9044–9053, 2021.

[14] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, W. Fan, and X. Xie. Dggan: Depth-image guided generative adversarial networks for disentangling rgb and depth images in 3d hand pose estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[15] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, and X. Xie. Mvhm: A large-scale multi-view hand mesh benchmark for accurate 3d hand pose estimation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 836–845, 2021.

[16] L. Chen, S.-Y. Lin, Y. Xie, Y.-Y. Lin, and X. Xie. Temporal-aware self-supervised learning for 3d hand pose and mesh estimation in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1050–1059, January 2021.

[17] L. Chen, S. Y. Lin, Y. Xie, H. Tang, Y. Xue, Y. Y. Lin, X. Xie, and W. Fan. Tagan: Tonality-alignment generative adversarial networks for realistic hand pose synthesis. In *30th British Machine Vision Conference, BMVC 2019*, 2020.

[18] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng. Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13274–13283, 2021.

[19] X. Chen and A. L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Advances in neural information processing systems*, pages 1736–1744, 2014.

[20] Y. Chen, H. Ma, D. Kong, X. Yan, J. Wu, W. Fan, and X. Xie. Nonparametric structure regularization machine for 2d hand pose estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 381–390, 2020.

[21] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.

[22] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017.

[23] E. Corona, T. Hodan, M. Vo, F. Moreno-Noguer, C. Sweeney, R. Newcombe, and L. Ma. Lisa: Learning implicit shape and appearance of hands. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20533–20543, 2022.

[24] J. M. Coughlan and S. J. Ferreira. Finding deformable shapes using loopy belief propagation. In *European Conference on Computer Vision*, pages 453–468. Springer, 2002.

[25] B. Doosti. Hand pose estimation: A survey. *CoRR*, abs/1903.01013, 2019.

[26] B. Doosti, S. Naha, M. Mirbagheri, and D. Crandall. Hope-net: A graph-based model for hand-object pose estimation. *arXiv preprint arXiv:2004.00060*, 2020.

[27] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in neural information processing systems*, 28, 2015.

[28] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu. RMPE: regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

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

[30] S. O.-E. Francisco Gomez-Donoso and M. Cazorla. Large-scale multiview 3d hand pose dataset. *ArXiv e-prints 1707.03742*, 2017.

[31] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3593–3601, 2016.

[32] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10833–10842, 2019.

[33] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 475–491, 2018.

[34] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[35] F. Gomez-Donoso, S. Orts-Escolano, and M. Cazorla. Large-scale multiview 3d hand pose dataset. *arXiv preprint arXiv:1707.03742*, 2017.

[36] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. Honnotate: A method for 3d annotation of hand and object poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3196–3206, 2020.

[37] S. Hampali, S. D. Sarkar, M. Rad, and V. Lepetit. Keypoint transformer: Solving joint identification in challenging hands and object interactions for accurate 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11090–11100, 2022.

[38] S. Han, B. Liu, R. Cabezas, C. D. Twigg, P. Zhang, J. Petkau, T.-H. Yu, C.-J. Tai, M. Akbay, Z. Wang, et al. Megatrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (TOG)*, 39(4):87–1, 2020.

[39] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11807–11816, 2019.

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

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

[42] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele. ArtTrack: Articulated multi-person tracking in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6457–6465, 2017.

[43] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. DeeperCut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016.

[44] U. Iqbal, A. Milan, and J. Gall. PoseTrack: Joint multi-person pose estimation and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2020, 2017.

[45] U. Iqbal, P. Molchanov, T. Breuel Juergen Gall, and J. Kautz. Hand pose estimation via latent 2.5 d heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 118–134, 2018.

[46] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[47] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[48] H. Joo, T. Simon, X. Li, H. Liu, L. Tan, L. Gui, S. Banerjee, T. Godisart, B. Nabbe, I. Matthews, et al. Panoptic Studio: A massively multiview system for social interaction capture. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):190–204, 2019.

[49] L. Ke, M.-C. Chang, H. Qi, and S. Lyu. Multi-scale structure-aware network for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 713–728, 2018.

[50] L. Ke, M.-C. Chang, H. Qi, and S. Lyu. Multi-scale structure-aware network for human pose estimation. In *European Conference on Computer Vision*. Springer, 2018.

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

[52] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[53] D. Kong, Y. Chen, H. Ma, X. Yan, and X. Xie. Adaptive graphical model network for 2d handpose estimation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2019.

[54] D. Kong, H. Ma, Y. Chen, and X. Xie. Rotation-invariant mixed graphical model network for 2d hand pose estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1546–1555, 2020.

[55] D. Kong, H. Ma, and X. Xie. Sia-gcn: A spatial information aware graph neural network with 2d convolutions for hand pose estimation. *arXiv preprint arXiv:2009.12473*, 2020.

[56] D. Kong, L. Zhang, L. Chen, H. Ma, X. Yan, S. Sun, X. Liu, K. Han, and X. Xie. Identity-aware hand mesh estimation and personalization from rgb images. In *European Conference on Computer Vision*, pages 536–553. Springer, 2022.

[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[58] D. Kulon, R. A. Guler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4990–5000, 2020.

[59] T. Lee and T. Hollerer. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 15(3):355–368, 2009.

[60] I. Lim, A. Dielen, M. Campen, and L. Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3d meshes. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[61] K. Lin, L. Wang, and Z. Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021.

[62] K. Lin, L. Wang, and Z. Liu. Mesh graphormer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12939–12948, 2021.

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

[64] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[65] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015.

[66] H. Ma, L. Chen, D. Kong, Z. Wang, X. Liu, H. Tang, X. Yan, Y. Xie, S.-Y. Lin, and X. Xie. Transfusion: Cross-view fusion with transformer for 3d human pose estimation. *arXiv preprint arXiv:2110.09554*, 2021.

[67] H. Ma, Z. Wang, Y. Chen, D. Kong, L. Chen, X. Liu, X. Yan, H. Tang, and X. Xie. Ppt: token-pruned pose transformer for monocular and multi-view human pose estimation. In *European Conference on Computer Vision*, pages 424–442. Springer, 2022.

[68] J. Malik, I. Abdelaziz, A. Elhayek, S. Shimada, S. A. Ali, V. Golyanik, C. Theobalt, and D. Stricker. Handvoxnet: Deep voxel-based network for 3d hand shape and pose estimation from a single depth map. *arXiv preprint arXiv:2004.01588*, 2020.

[69] G. Moon, J. Y. Chang, and K. M. Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5079–5088, 2018.

[70] G. Moon and K. M. Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 752–768. Springer, 2020.

[71] G. Moon, T. Shiratori, and K. M. Lee. Deephandmesh: A weakly-supervised deep encoder-decoder framework for high-fidelity hand mesh modeling. In *European Conference on Computer Vision*, pages 440–455. Springer, 2020.

[72] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee. Interhand2.6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision (ECCV)*, 2020.

[73] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–59, 2018.

[74] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1284–1293, 2017.

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

[76] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011.

[77] P. Panteleris and A. Argyros. Back to rgb: 3d tracking of hands and hand-object interactions based on short-baseline stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 575–584, 2017.

[78] P. Panteleris, I. Oikonomidis, and A. Argyros. Using a single RGB frame for real time 3D hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445. IEEE, 2018.

[79] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017.

[80] J. Park, Y. Oh, G. Moon, H. Choi, and K. M. Lee. Handoccnet: Occlusion-robust 3d hand mesh estimation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1496–1505, 2022.

[81] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[82] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele. DeepCut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4929–4937, 2016.

[83] T. Piumsomboon, A. Clark, M. Billinghurst, and A. Cockburn. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*, pages 282–299. Springer, 2013.

[84] Pytorch. Pytorch margin ranking loss, 2022.

[85] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1106–1113, 2014.

[86] N. Qian, J. Wang, F. Mueller, F. Bernard, V. Golyanik, and C. Theobalt. Html: A parametric hand texture model for 3d hand reconstruction and personalization. In *European Conference on Computer Vision*, pages 54–71. Springer, 2020.

[87] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[88] J. M. Rehg and T. Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European conference on computer vision*, pages 35–46. Springer, 1994.

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

[90] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):1–17, 2017.

[91] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[92] T. Schmidt, R. A. Newcombe, and D. Fox. Dart: Dense articulated real-time tracking. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[93] M. Scutari and K. Strimmer. Introduction to graphical modelling, 2010.

[94] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[95] L. Sigal, M. Isard, B. Sigelman, and M. Black. Attractive people: Assembling loose-limbed models using non-parametric belief propagation. *Advances in neural information processing systems*, 16, 2003.

[96] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[97] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[98] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[99] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

[100] A. Spurr, U. Iqbal, P. Molchanov, O. Hilliges, and J. Kautz. Weakly supervised 3d hand pose estimation via biomechanical constraints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 211–228. Springer, 2020.

[101] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.

[102] S. Sridhar, A. M. Feit, C. Theobalt, and A. Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3643–3652. ACM, 2015.

[103] E. B. Sudderth, M. I. Mandel, W. T. Freeman, and A. S. Willsky. Visual hand tracking using nonparametric belief propagation. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 189–189. IEEE, 2004.

[104] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019.

[105] D. J. Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5610–5619, 2016.

[106] A. Tkach, A. Tagliasacchi, E. Remelli, M. Pauly, and A. Fitzgibbon. Online generative model personalization for hand tracking. *ACM Transactions on Graphics (ToG)*, 36(6):1–11, 2017.

[107] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in neural information processing systems*, pages 1799–1807, 2014.

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

[109] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[110] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[111] C. Wan, T. Probst, L. V. Gool, and A. Yao. Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10853–10862, 2019.

[112] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3D regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2018.

[113] C. Wang, S. Pan, G. Long, X. Zhu, and J. Jiang. Mgae: Marginalized graph autoencoder for graph clustering. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 889–898, 2017.

[114] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[115] J. Wang, Z. Wei, T. Zhang, and W. Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016.

[116] Y. Wang, C. Peng, and Y. Liu. Mask-pose cascaded cnn for 2d hand pose estimation from single color image. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[117] Z. Wang, H. Chen, X. Li, C. Liu, Y. Xiong, J. Tighe, and C. Fowlkes. Sscap: Self-supervised co-occurrence action parsing for unsupervised temporal action segmentation. In *WACV*, 2022.

[118] Z. Wang, L. Chen, S. Rathore, D. Shin, and C. Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. In *arxiv*, 2019.

[119] Z. Wang, L. Chen, S. Rathore, D. Shin, and C. Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. *arXiv preprint arXiv:1905.07718*, 2019.

[120] Z. Wang, L. Chen, S. Rathore, D. Shin, and C. Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. In *Arxiv 1905.07718*, 2019.

[121] Z. Wang, D. Shin, and C. Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. In *ECCV 3DPW workshop*, 2020.

[122] Z. Wang, D. Shin, and C. C. Fowlkes. Predicting camera viewpoint improves cross-dataset generalization for 3d human pose estimation. *arXiv preprint arXiv:2004.03143*, 2020.

[123] Z. Wang, J. Yang, and C. Fowlkes. The best of both worlds: Combining model-based and nonparametric approaches for 3d human body estimation. In *CVPR ABAW workshop*, 2022.

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

[125] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[126] B. Xiao, H. Wu, and Y. Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 466–481, 2018.

[127] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, and S. Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, volume 41, pages 641–676. Wiley Online Library, 2022.

[128] F. Xiong, B. Zhang, Y. Xiao, Z. Cao, T. Yu, J. T. Zhou, and J. Yuan. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–802, 2019.

[129] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[130] X. Yan, H. Tang, S. Sun, H. Ma, D. Kong, and X. Xie. After-unet: Axial fusion transformer unet for medical image segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3971–3981, 2022.

[131] L. Yang, J. Li, W. Xu, Y. Diao, and C. Lu. Bihand: Recovering hand mesh with multi-stage bisected hourglass networks. *arXiv preprint arXiv:2008.05079*, 2020.

[132] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2016.

[133] Z. Yu, J. S. Yoon, I. K. Lee, P. Venkatesh, J. Park, J. Yu, and H. S. Park. Humbi: A large multiview dataset of human body expressions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2990–3000, 2020.

[134] S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Yong Chang, K. Mu Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, et al. Depth-based 3D hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2018.

[135] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng. End-to-end hand mesh recovery from a monocular rgb image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2354–2364, 2019.

[136] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[137] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

[138] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[139] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single RGB images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4903–4911, 2017.

[140] C. Zimmermann and T. Brox. Learning to estimate 3d hand pose from single rgb images. Technical report, arXiv:1705.01389, 2017. https://arxiv.org/abs/1705.01389.

[141] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 813–822, 2019.