A dissertation submitted in accordance with the requirements for the degree of

Doctor of Philosophy

3D Hand Pose Estimation: Methods, Datasets, and Challenges

SHANXIN YUAN

Imperial College London

Department of Electrical & Electronic Engineering

October 2018

© The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. The research detailed in this dissertation was conducted under the guidance of **Dr. Tae-Kyun Kim**.

The research presented in this dissertation resulted in several publications and submissions with joint authorship as follows:

S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T-K. Kim, Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals, **CVPR**, 2018, **Spotlight**.

S. Yuan, Q. Ye, B. Stenger, S. Jain, T-K. Kim, BigHand2.2M Benchmark: Hand Pose Data Set and State of the Art Analysis, **CVPR**, 2017.

S. Yuan, B. Stenger, T-K. Kim, 3D Hand Pose Estimation from RGB Images with Privileged Information from Depth Data, arXiv, 2018.

S. Yuan, Q. Ye, G. Garcia-Hernando, T-K. Kim, The 2017 Hands in the Million Challenge on 3D Hand Pose Estimation, arXiv, 2017.

Q. Ye, **S. Yuan**, T-K. Kim, Spatial Attention Deep Net with Partial PSO for Hierarchical Hybrid Hand Pose Estimation, **ECCV**, 2016.

G. Garcia-Hernando, **S. Yuan**, S. Baek, T-K. Kim, First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations, **CVPR**, 2018.

D. Tang, Q. Ye, **S. Yuan**, J. Taylor, P. Kohli, C. Keskin, T-K. Kim, J. Shotton, Opening the Black Box: Hierarchical Sampling Optimization for Estimating Human Hand Pose, **TPAMI** 2018

Shanxin Yuan, October 2018

Abstract

3D hand pose estimation is an important task in the computer vision community due to its vast applications, including but not limited to, human computer interaction, virtual reality and augmented reality, sign language recognition, medical image analysis. The challenges for this task lie in high degree of freedom of a human hand, self-occlusions, different hand shapes, ambiguities among different fingers. The obstacles faced by the research communities are lack of proper methods and limitation in the current datasets. In view of this, I investigate in this thesis in three aspects: methods, datasets, and challenges. More specifically, the contributions of this thesis are: (1) Proposed a large-scale hand pose dataset, collected using a novel capture method, the dataset is known as the BigHand2.2M dataset; (2) Hosted a depth-based 3D hand pose challenge that attracted the top research groups across the world to evaluate the current best state-of-the-art methods, to investigate into the best practices, and to show some promising research directions; (3) Proposed a method for 3D hand pose estimation from RGB images with privileged information from depth data.

Real datasets are limited in quantity and coverage, mainly due to the difficulty to annotate them. To deal with this issue, this thesis proposed a tracking system with six magnetic 6D sensors and inverse kinematics to automatically obtain 21-joints hand pose annotations of depth maps captured with minimal restriction on the range of motion. The automatic annotation method allows us to build the largest real dataset with higher joint annotations accuracy. To find out the current state of 3D hand pose estimation from depth and the next challenges, we hosted the *Hands In the Million Challenge (HIM2017)*, and investigated the state-of-the-art methods on three tasks: single frame 3D pose estimation, 3D hand tracking, and hand pose estimation during object interaction. This thesis analysed the performance of different CNN structures with regard to hand shape, joint visibility, view point and articulation distributions. The advancement in hand pose estimation from RGB images lagged behind that of depth images, this thesis proposed a method for hand pose estimation from RGB images that uses both

external large-scale depth image datasets and paired depth and RGB images as privileged information at training time. We show that providing depth information during training significantly improves performance of pose estimation from RGB images during testing.

Keywords

3D hand pose estimation, human hand, convolutional neural network, deep learning, random forest, particle swarm optimization, real dataset, synthetic dataset, RGB images, depth images, magnetic 6D sensors, BigHand2.2M, Hand in the Million challenge, inverse kinematic, hand model, discriminative model, generative model, privileged information, transfer learning, latent space, generative adversarial network.

Acknowledgments

I would like to express gratitude to my supervisor, Dr. Tae-Kyun Kim, for his guidance, support, encouragement, and scholarship award. I am also highly grateful to Dr. Björn Stenger, for his constructive suggestion which helps shape me as a researcher.

I am grateful to Imperial College London for providing me with financial security through the Department Fees Scholarship, without which this work would not have been possible.

I thank all the members, past and present, of the Imperial College Vision Lab, for their companionship. Discussions with them also inspired some of the findings in this thesis. I also thank the administrative staff of Imperial College London for their help during my study period.

Finally, I thank my family for their unconditional support.

CONTENTS

LIST OF FIGURES xvi

LIST OF TABLES xxvi

CHAPTER 1

INTRODUCTION 1

1.1 Hand Pose Representation 2

1.2 Evaluation Metrics 3

1.2.1 Standard error metrics 3

1.2.2 Proposed error metrics 4

1.3 Useful Algorithms 4

1.3.1 Particle Swarm Optimization 4

1.3.2 Random Forest 6

1.3.3 Deep Learning 7

1.4 Literature Review 8

1.4.1 Generative, Discriminative, and Hybrid Methods 8

1.4.2 Model-based Hand Tracking 12

1.4.3 Tracking a Hand Manipulating an Object 12

1.4.4 Tracking Hand in Egocentric RGB-D Images 13

1.4.5 3D Body Pose Estimation from Depth and RGB Images. 13

1.4.6 3D Hand Pose Estimation From RGB Images 14

1.4.7 Contact Force Estimation 15

1.5 Relationship between Chapters 15
1.6 Thesis Outlines and Contributions 16
1.6.1 BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis 17
1.6.2 Hand Pose from Depth: Current Achievements and Future Goals 17
1.6.3 Pose Estimation from RGB with Depth as Privileged Information 18
1.6.4 Conclusion and Future Work 19
1.6.5 Appendix: Hands In the Million Challenge 19

CHAPTER 2
BIGHAND2.2M BENCHMARK: HAND POSE DATASET AND STATE OF THE ART ANALYSIS **21**
2.1 Overview 21
2.2 Existing benchmarks 25
2.3 Full hand pose annotation 29
2.3.1 Annotation by inverse kinematics 29
2.3.2 Synchronization and calibration 32
2.4 BigHand2.2M benchmark 33
2.4.1 Hand view-point space 33
2.4.2 Hand articulation space 33
2.4.3 Hand shape space 34
2.5 Analysis of the state of the art 36
2.5.1 Cross-benchmark performance 37
2.5.2 State-of-the-art comparison 41
2.5.3 Baselines on BigHand2.2M 43
2.5.4 Egocentric dataset 45
2.6 Discussion and conclusions 46

CHAPTER 3
HAND POSE FROM DEPTH: CURRENT ACHIEVEMENTS AND FUTURE GOALS **47**

3.1 Overview 47
3.2 Related work. 50
3.3 Evaluation tasks 51
3.4 Evaluated methods 51
3.5 Results 57
3.5.1 Single frame pose estimation 58
3.5.2 Hand pose tracking 72
3.5.3 Hand object interaction 75
3.6 Discussion and conclusions 79

CHAPTER 4
POSE ESTIMATION FROM RGB WITH DEPTH AS PRIVILEGED INFORMATION **81**
4.1 Overview 81
4.1.1 Contributions 84
4.2 Related Work 85
4.3 Methods 88
4.3.1 Architecture 88
4.3.2 Training with privileged information 89
4.3.3 Foreground mask as privileged information 91
4.4 Experiments 91
4.4.1 3D hand pose estimation from RGB 92
4.4.2 2D hand pose estimation from RGB 99
4.5 Discussion and conclusions 103

CHAPTER 5
CONCLUSION AND FUTURE WORK **105**
5.1 Contributions and Relationship between Chapters 105
5.1.1 Chapter 2 105
5.1.2 Chapter 3 106
5.1.3 Chapter 4 107

5.1.4 Appendix 107
5.2 Future Work 108

CHAPTER A
HAND IN THE MILLION CHALLENGE **111**
A.1 Overview 111
A.2 Challenge tasks 114
A.3 Dataset details 114
A.3.1 Training data 115
A.3.2 Test data 116
A.4 Participation rules 117
A.4.1 Challenge rules 118
A.4.2 Annotation and results format 119
A.4.3 Evaluation 119
A.5 Discussion and conclusions 122

REFERENCES **131**

LIST OF FIGURES

2.1 **Example images from the *BigHand2.2M* dataset**. The dataset covers the range of hand poses that can be assumed without applying external forces to the hand. The accuracy of joint annotations is higher than in previous benchmark datasets. 23

2.2 **More example images from the *BigHand2.2M* dataset**. 24

2.3 **Hand model definition**. (a) Our hand model has 21 joints and moves with 31 degrees of freedom (dof). (b) Model fitted to hand shape. (c) and (d) show how hand shape is measured. 29

2.4 **Hand pose inference using six 6D magnetic sensors**. Global hand pose can be inferred from the location and orientation of sensor S6 on the back of the palm. Each sensor on the nail is used to infer the TIP and DIP joints of the corresponding finger. Each PIP joint can be calculated using bone lengths and physical constraints. 30

2.5 **Equipment** [NDI trakSTAR,]. (a) Mid-range Transmitters, (b) trakSTAR electronics unit, (c) 'Model 180' 6D magnetic sensor. 30

2.6 **Annotation settings**. The equipment used in our annotation system are: Two hardware synchronized electromagnetic tracking units, six 6D magnetic sensors, one mid-range transmitter, and an Intel RealSense SR300 camera. 32

2.7 **2D t-SNE embedding of the view point space**. *BigHand2.2M* is represented by blue, ICVL by red, and NYU by green dots. The figure shows global view point space coverage. Compared with existing datasets, the *BigHand2.2M* contains a more complete range of variation. 34

2.8 **2D t-SNE embedding of the hand articulation space**. *BigHand2.2M* is represented by blue, ICVL by red, and NYU by green dots. The figure shows articulation space (25D). Compared with existing datasets, the *BigHand2.2M* contains a more complete range of variation. 35

2.9 **2D t-SNE embedding of the hand pose space**. *BigHand2.2M* is represented by blue, ICVL by red, and NYU by green dots. The figure shows the combined of global orientation and articulation coverage. Compared with existing datasets, the *BigHand2.2M* contains a more complete range of variation. 36

2.10 **Hand shape variation**. Hand variation is visualized by applying PCA to the shape parameters. The *BigHand2.2M* dataset contains 10 hand shapes and an additional shape for testing. The ICVL dataset contains one hand shape due to its annotation method. The NYU dataset includes two hand shapes, the MSRC dataset includes one synthetic hand shape. 37

2.11 **Cross-benchmark performance**. CNN models are trained on the ICVL, NYU, MSRC, and the new *BigHand2.2M* dataset, respectively, and evaluated on NYU test data. A CNN trained on *BigHand2.2M* achieves state-of-the-art performance on NYU, while the CNNs trained on ICVL, NYU, and MSRC do not generalize well to other benchmarks. The networks CNN_MSRC, CNN_ICVL, CNN_NYU, and CNN_BigHand are trained on the training set of MSRC, ICVL, NYU, and *BigHand2.2M*, respectively. 38

2.12 **Cross-benchmark performance**. CNN models are trained on the ICVL, NYU, MSRC, and the new *BigHand2.2M* dataset, respectively, and evaluated on ICVL test data. A CNN trained on *BigHand2.2M* achieves state-of-the-art performance on ICVL, while the CNNs trained on ICVL, NYU, and MSRC do not generalize well to other benchmarks. The networks CNN_MSRC, CNN_ICVL, CNN_NYU, and CNN_BigHand are trained on the training set of MSRC, ICVL, NYU, and *BigHand2.2M*, respectively. 39

2.13 **Data size effect on cross benchmark evaluation.** When the CNN model is trained on $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and all of the benchmark data, the test results on ICVL, NYU, MSRC, and *BigHand2.2M* keep improving. 40

2.14 **Generalization of the CNN trained on *BigHand2.2M***. The CNN generalizes to the ICVL dataset with a lower error than the original annotated ground truth. (top) ICVL ground truth annotations, (bottom) our estimation results. 41

2.15 **MSRC benchmark examples**. Synthetic data lacks real hand shape and sensor noise, and tends to have kinematically implausible hand poses. The top row shows some depth images, the bottom row shows the corresponding ground truth annotation. 41

2.16 **Hand pose estimation performance**.Baseline performances on the new subject's 37K frames of hand images. The *Holi* CNN significantly outperforms the tracking-based methods FORTH [Oikonomidis et al., 2011a] and Intel [Intel SR300,]. 42

2.17 **Hand pose estimation performance**. The CNN trained on 90% of the *BigHand2.2M* data achieves high accuracy on the remaining 10% validation images. 43

2.18 **Hand pose estimation performance**. 10-fold cross-validation result when using the CNN for egocentric hand pose estimation. We achieved a similar-level accuracy to that of third-view hand pose estimation. 44

2.19 **Qualitative results on the egocentric-view dataset**. A CNN trained on *BigHand2.2M* achieves state-of-the-art performance in the egocentric-view pose estimation task. 45

3.1 **Evaluated tasks**. For each scenario the goal is to infer the 3D locations of the 21 hand joints from a depth image. In **Single frame pose estimation** (top) and the **Interaction task** (bottom), each frame is annotated with a bounding box. In the **Tracking task** (middle), only the first frame of each sequence is fully annotated. 48

3.2 **Method taxonomy.** We selection these design choices to group the evaluated methods. 54

3.3 **Method taxonomy.** Our evaluated methods are grouped into different categories. 54

3.4 **Estimation errors.** (Left) Proportion of frames within maximum error threshold [Taylor et al., 2012]. (Right) Proportion of joints within an error threshold [Sharp et al., 2015]. 57

3.5 **Estimating annotation errors.** Three examples of *Easy Poses* overlayed with estimates by the top five methods (shown in different colors). Poses vary slightly, but are close to the ground truth (black). 57

3.6 **Success rates for different methods.** (Left) all evaluated methods using all test data. (Right) the average of the top five methods for four cases. 58

3.7 **Easy pose examples.** From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 60

3.8 (Left) Direct comparison with 2D CNN and 3D CNN, *mmadadi* is a 2D CNN, *NAIST_RV* has the same structure but replaced 2D CNN with 3D CNN. (Right) The average of top-two detection-based and regression-based methods in four cases. 61

3.9 **Generalization to new hand shapes.** Each column represents one unseen hand shape. From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 62

3.10 (Left) Comparison among structured methods. (Right) Comparison of a cascaded multistage method (*THU_VCLab*) and a one-off method (*rvhand*). Both of them use *REN* [Guo et al., 2017] as backbone. 64

3.11 **Joint visibility. Top:** Joint visibility distributions for training set and testing sets. **Bottom:** Average error (mm) for different numbers of visible joints and different methods. 65

3.12 **Occlusions.** From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 66

3.13 **View point distributions.** The error is significantly higher for small angles between hand and camera orientations. 67

3.14 **Articulation distribution.** Errors increase for larger finger articulation angles. 68

3.15 **Viewpoints in range [0, 10] degrees.** From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 69

3.16 **Viewpoints in range [170, 180] degrees.** From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 70

3.17 **Articulation angle larger than 70 degrees.** From top to bottom, rows show results of *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU_VCLab* [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Wan, 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017], respectively. The ground truth annotation is shown in black. 71

3.18 **Average error of the top five methods for each joint in the *Single frame pose estimation* task.** Finger tips have larger errors than other joints. For non-tip joints, joints on ring finger and middle finger have lower average errors than other fingers. 'T', 'I', 'M', 'R', 'P' denotes 'Thumb', 'Index', 'Middle', 'Ring', and 'Pinky' finger, respectively. 72

3.19 **Error curves for hand tracking.** (Left) number of frames with IoU below threshold. (Right) success rate for the detection part of each method in two cases. 73

3.20 **Error curves for hand tracking.** Success rate for three methods in four cases. 74

3.21 **More results on tracking.** Left is for the proportion of frames within maximum error threshold [Taylor et al., 2012], right plot is the results for the proportion of joints within error threshold [Sharp et al., 2015] 76

3.22 **Error curves for hand-object interaction.** (Left): success rate for each method using average error per-frame. (Right) success rate for visible and occluded joints. 76

3.23 **Error curves for hand-object interaction.** Average error for each joint. 77

3.24 **More results on hand object interaction.** Left is for the proportion of frames within maximum error threshold [Taylor et al., 2012], right plot is the results for the proportion of joints within error threshold [Sharp et al., 2015] 78

4.1 **Proposed framework for 3D hand pose estimation from an RGB image using privileged depth data.** *Training proceeds in two stages, a pre-training stage and privileged information (PI)-training stage. In the first stage, a depth-based network (top) and an RGB-based (bottom) network are trained independently to minimize 3D pose loss Loss_D and Loss_C. In the second stage, we freeze the parameters of the depth-based network and continue training with paired RGB and depth images, by minimizing a joint loss, which includes Loss_C and a mid-level feature regression loss Loss_Inter.* 82

4.2 **Treating hand mask as privileged information**. Hand mask are used as privileged information to suppress the responses from the background area in the middle layers. 90

4.3 **Results on RHD dataset for 3D hand pose accuracy**.*This plot shows the self-comparison on RHD dataset.* 93

4.4 **Results on Stereo dataset for 3D hand pose accuracy**.*The plot shows the self-comparison on Stereo dataset.* 94

4.5 **Qualitative 3D pose estimation results**. Comparing the outputs of the RGB network (blue, second row), the RGB network with PI training (red, third row), and the Depth network (magenta, bottom row) with the ground truth 3D pose (green) on the Stereo dataset. Top row are the original images. 95

4.6 **More qualitative results for 3D pose estimation**. Comparing the outputs of the RGB network (blue, second row), the RGB network with PI training (red, third row), and the Depth network (magenta, bottom row) with the ground truth 3D pose (green) on the Stereo dataset. Top row are the original images. 96

4.7 **Results one Stereo and RHD dataset for 3D hand pose accuracy**. *The plot is comparison with state-of-the-art on Stereo dataset.* 97

4.8 **Results one Stereo and RHD dataset for 2D pose accuracy**.*The plot is self comparison on RHD dataset.* 97

4.9 **Results one Stereo and RHD dataset for 2D pose accuracy**.*The plot is self comparison on Stereo dataset.* 98

4.10 **Results one Stereo and RHD dataset for 2D pose accuracy**.*The plot is comparison with state-of-the-art on the Dexter-Object dataset.* 98

4.11 **Feature activation maps**. (top row) input images, (row 2) activations of the RGB network trained on RGB only, (row 3) activations of the RGB network trained with additional depth data, (row 4) activations of the depth network, and (row 5) depth images. During training, depth data helps the RGB network focus on the region of interest, reducing the influence of background regions. 100

4.12 **More feature activation maps**. (top row) input images, (row 2) activations of the RGB network trained on RGB only, (row 3) activations of the RGB network trained with additional depth data, (row 4) activations of the depth network, and (row 5) depth images. During training, depth data helps the RGB network focus on the region of interest, reducing the influence of background regions. 101

4.13 **Loss function evolution on the stereo dataset [Zhang et al., 2016].** The loss for 3D hand pose (left plot) of the RGB network on the test data converges at iteration 15,000, we continue training for another 5,000 iterations. From iteration 20,000, we fix the depth network parameters and connect mid-level features between the RGB and depth networks, and continue training by minimizing the joint loss (right plot) using RGB-D image pairs. The intermediate loss (middle plot) is used to suppress the difference between the mid-level feature between the RGB and depth networks. Loss for 3D hand pose of the RGB network, and the joint loss stop decreasing at around iteration 30k. 102

4.14 **Qualitative 2D pose estimation results**. Comparing the outputs of (middle) the RGB network and (bottom) the RGB network with PI training on the RHD dataset. Top row are the original images. 103

4.15 **Qualitative 2D pose estimation results on Stereo dataset**. Comparing the outputs of (top) the RGB network and (bottom) the RGB network with PI training. 104

A.1 **Example images of the challenge. Top row:** third-person viewpoint hand poses. **Middle row:** first-person viewpoint hand poses in object-free scenario. **Bottom row:** first-person viewpoint hand poses involving manipulated objects. 113

A.2 **Challenge tasks**. **Top row: 3D hand pose tracking**, the first frames of sequences are fully annotated. **Bottom row: 3D hand pose estimation**, each frame is annotated with a bounding box and frames are shuffled. The objective of both task is to infer the 3D position of the 21 joints at each given depth image. 113

A.3 **Test data**. The test images consists of two part, (1) these images with random poses for all the ten subjects, (2) these images with egocentric poses for five unseen subjects and the second halves of egocentric poses for five seen subjects. 116

A.4 **Seen vs Unseen subjects**. Seen subjects are chosen to build the training data. The plot shows the first two principal components of the hand shape parameters. 117

A.5 **Training data**. To build the training data, five out of ten subjects in *BigHand2.2M* are selected. The training data consists of uniformly sampled schemed poses and the half of the egocentric poses. 118

A.6 **Hand annotation format**. The left figure shows the hand skeleton, the right figure is a real depth image with annotation. 'T', 'I', 'M', 'R', 'P' denotes 'Thumb', 'Index', 'Middle', 'Ring', and 'Pinky' finger, respectively. 'MCP', 'PIP', 'DIP', 'TIP', denotes metacarpophalangeal, proximal interphalangeal, and distal interphalangeal, and tip joints, respectively. 120

A.7 **Joint visibility illustration**. Top-left: all joints are visible. Top-right: 'TMCP', 'TPIP' joints are invisible, they are occluded by other fingers. Bottom-left: 'RDIP', 'RTIP', 'MDIP', 'MTIP' are occluded by the object. Bottom-right: 'PPIP', 'RPIP','MPIP', 'IDIP' are occluded by the thumb in this egocentric viewed hand pose. 121

A.8 **Example sequences for third-person viewpoint**. The sequences are for all subjects in BigHand2.2M [Yuan et al., 2017], each row shows one sequence of a subject. 124

A.9 **More example sequences for third-person viewpoint**. 125

A.10 **Example sequences for egocentric view**. The sequences are from the egocentric part of the BigHand2.2M [Yuan et al., 2017] dataset. The test sequences covers all the 32 *extremal poses* where each finger assumes a maximally bent or extended position. 126

A.11 **More example sequences for egocentric view**. 127

A.12 **Examples of sequences of hand actions [Garcia-Hernando et al.,]:** From top to bottom: 'open peanut butter', 'put sugar', 'pour milk' and 'wash with sponge' (all in kitchen). Figure courtesy of [Garcia-Hernando et al.,]. 128

A.13 **More examples of sequences of hand actions [Garcia-Hernando et al.,]:** From top to bottom: 'charge cell phone' and 'tear paper' (office); 'handshake' and 'toast with wine glass' (social). Figure courtesy of [Garcia-Hernando et al.,]. 129

List of Tables

2.1 **Benchmark comparison**. Existing datasets are limited in the number of frames, due to their annotation methods, either manual or semi-automatic. Our automatic annotation method allows collecting fully annotated depth images at frame rate. 25

2.2 **Benchmark comparison**. Our dataset is collected with the latest Intel RealSense SR300 camera [Intel SR300,], which captures depth images at 640 × 480-pixel resolution. 25

2.3 **Cross-benchmark comparison**. Mean errors of CNNs trained on ICVL, NYU, MSRC and *BigHand2.2M* when cross-tested. The model trained on *BigHand2.2M* performs well on ICVL and NYU, less so on the synthetic MSRC data. Training on ICVL, NYU, or MSRC does not generalize well to other datasets. 40

2.4 Egocentric Benchmark size comparison. The egocentric subset of *BigHand2.2M* dataset is 130 time larger than the next largest available dataset. 44

3.1 **Data set sizes and number of subjects.** 50

3.2 **Methods evaluated in the hand pose estimation challenge.** Methods are ordered by average error on the leader-board. * in both methods, hand segmentation is performed considering different hand arm lengths. 52

3.3 **Methods evaluated in the hand pose estimation challenge.** Methods are ordered by average error on the leaderboard. For hyper-parameters, 'MT' is momentum, 'LR' is learning rate, 'WD' is weight decay, 'MB' is Mimi-batch size, 'EP' is total epoch number, SGD is 'Stochastic Gradient Descent'. 53

3.4 **Mean errors (in mm) for single frame pose estimation, divided by cases.** 'Seen' and 'Unseen' refers to whether or not the hand shape was in the training set, and 'Occ' denotes 'Occluded joints'. 58

3.5 **Methods evaluated on 3D hand pose tracking.** The last column is the average error in mm for all frames. 73

3.6 **Methods evaluated on hand pose estimation during hand-object interaction.** The last column is the average error (mm) for all frames. 77

4.1 **Public datasets used in our experiments.** 91

4.2 **2D Hand Pose Accuracy**. Results when training on the RHD and Stereo datasets. 103

A.1 **Size of the challenge data splits:** number of subjects, sequences, and frames. 115

A.2 **Test scenarios.** Detailed evaluation will be performed in different scenarios. 117

A.3 **Evaluation metrics.** To evaluate a method, we take into account joints visibility as well as pose happening frequency. 121

A.4 Average errors (in mm) for different viewpoint and different subjects. 122

CHAPTER 1

INTRODUCTION

Contents

1.1	Hand Pose Representation	2
1.2	Evaluation Metrics	3
1.3	Useful Algorithms	4
1.4	Literature Review	8
1.5	Relationship between Chapters	15
1.6	Thesis Outlines and Contributions	16

With the advancements made in visual data storage, computing power, and state-of-the-art algorithms, we are enjoying the convenience brought by artificial intelligence more than ever before. From low-level image processing, *e.g.* capturing visually appealing pictures with our smart phones to high-level applications, *e.g.* autonomous driving cars, computer vision is playing a key role. In the computer vision society, one fundamental component is to do hand pose estimation from images. Hand pose estimation can be used in a large number of applications, including object manipulation in virtual reality, human computer interaction, and medical image analysis. The following two examples are to illustrate the importance of hand

pose estimation: (1) Gaming in virtual reality with two hands controlling virtual objects is more user-friendly than using a mouse and a keyboard. (2) With accurate hand pose estimation, joint-related diseases, *e.g*. parkinsons, can be diagnosed quicker than traditional methods. Human hand plays an important role in our daily life, performing all kinds of poses, e.g. grasping, waving, pointing, without explicit consciousness. Human hand as one of our most important tool, for which we have taken for granted, is less studied in computer vision and machine learning than other areas, e.g. human face and human body pose.

1.1 Hand Pose Representation

In the computer vision society, hand pose estimation is the task of inferencing hand pose from images (weather color images, or depth images). The hand pose is usually defined as the locations of all the key points (or joints). In the literature, there are several explicit ways to define the hand pose. Depends on the number of key points, there are two main schemes to define the key points: 1) 16 key points [Tang et al., 2014] and 2) 21 key points [Yuan et al., 2017, Zhang et al., 2016]. The 16-key-point scheme consists of one key point on the palm center, three key points for each of the five fingers. The three key points on each finger are located at the centers of three connected bones, which are called distal phalange, middle phalange, and proximal phalange, in accordance with biological anatomy. The 21-key-point scheme consists of one key point on the wrist, and four key points on each finger. The four key points on each finger are three joints on the connected bones, and one finger tip. Sometimes the 21-key-point have variances, e.g., some authors choose the palm center instead of the wrist as one key point [Zhang et al., 2016]. Even the same 21-key-point scheme has some minor variance among different datasets, e.g. some authors choose the MetaCarpoPhalangeal (MCP) joints to be closer to the palm center [Yuan et al., 2017], other are a bit further away from the palm center [Sun et al., 2015]. A simple geometric model can facilitate computation. Oikonomidis *et al* [Oikonomidis et al., 2011a] proposed a polygonal model consisting of a palm and five fingers using spheres, ellipsoids, and cylinders. This polygonal model has the merit of enabling a high degree of computational parallelism, and thus make it possible for

GPU parallel computing. Qian *et al*presented a much simpler sphere model using 48 spheres with flexible locations and fixed empirically set radii. Each of the five fingers has 4 degree of freedoms (DoFs) (2 are for the finger's base to represent the flexion and abduction of the finger, two for flexion for the remaining joints), while the whole hand has the global 6 DoFs (3 DoFs are for the global location and the remaining 3 DoFs are for the global orientation of the hand). In order to avoid implausible articulated hand pose, kinematic constraints are usually added to the hand model to keep the joint angles within valid range.

1.2 Evaluation Metrics

Estimated hand poses are usually evaluated using different error metrics. We use both standard error metrics and a new metric that we believe will provide further insights into the performance of a hand pose estimation system.

1.2.1 Standard error metrics

Following the literature [Oikonomidis et al., 2011a, Taylor et al., 2012, Sharp et al., 2015], we use the following error metrics:

1. The mean error for all joints for each frame and average across all test frames [Oikonomidis et al., 2011a]. This metric can faithfully evaluate the performance a hand pose estimation system.

2. The ratio of joints within a certain error bound [Sharp et al., 2015] defined as:

$$r_j = \frac{N_j}{N * n}, \tag{1.1}$$

where N is total number of frame, n is the number of joints of a hand (21 in our case), $N_j = f(\epsilon)$ is the total number of joints within a euclidean distance of ϵ to the ground

truth. Accuracy curve will be drawn by varying the value of ϵ. This metric takes into account each joint's performance individually across all the frames.

3. The most challenging one, the ratio of frames r_f that have all joints within a certain distance to ground truth annotation [Taylor et al., 2012] defined as:

$$r_f = \frac{N_f}{N}, \tag{1.2}$$

where N is total number of frame, $N_f = g(\epsilon)$ is the number of frames whose joints are all with in euclidean distance of ϵ to the ground truth. This metric take only into account the worst estimated joint of a hand, and ignore the rest better estimated joints.

1.2.2 Proposed error metrics

We also propose new evaluation metrics, by taking into account the joint visibility. Hand pose often present occlusions, *e.g.*, self occlusion and occlusion from objects. When occlusion happens, especially in the settings of egocentric view and hand-object interaction, measuring only the quality of the visible joints can be of interest.

1.3 Useful Algorithms

In this section, I present the most-related machine learning algorithms for 3D hand pose estimation: particle swarm optimization, random forest, and deep learning. State-of-the-art methods using these machine learning algorithms are reviewed.

1.3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic optimization algorithm introduce by Kennedy and Eberhart [Kennedy and Eberhart, 1995] in 1995, originated in the social behaviors' studies of synchronous bird flocking and fish schooling. The original PSO algorithm

has been modified by several researchers to improve its convergence properties and search capabilities. PSO variants abound in the literature, including variants using inertia weight ω [Shi and Eberhart,] and constriction factor [Clerc and Kennedy, 2002], fully informed particle swarm [Mendes et al., 2002] and PSO using different population topology [Kennedy and Mendes, 2002] (static topologies and dynamic topologies). Particle Swarm Optimization has been used in many generative methods [Oikonomidis et al., 2011a, Oikonomidis et al., 2011b, Oikonomidis et al., 2012, Qian et al., a], where the search in the hand model configuration space is executed to find the configuration that minimizes the discrepancy between the feature and the hand hypothesis. Oikonomidis *et al* [Oikonomidis et al., 2011a] successfully tracked hand poses using PSO. Qian *et al* [Qian et al., a] proposed a hand pose tracking method by exploiting the merits of the both Iterated Closest Point (ICP) and PSO. One main disadvantage of these methods is that they do optimization using non-convex objective functions defined in the high dimensional hand configuration space, which is time consuming and easily stuck into local minimas. One way to alleviate this problem is to divide the full hand pose space into several subspaces. Partial PSO, dividing the large dimensional parameter space into smaller subspaces and performing PSO within each to find partial poses, has been widely used for markerless human body pose tracking [John et al., , Sharifi et al.,], and recently for hand pose estimation [Park et al., ,Poier et al.,]. Vijay *et al* [John et al.,] divided the full 31 dimensional human body pose space into 12 subspaces to estimation partial poses and improved the accuracy and speed of markerless human articulated tracking. Sharifi *et al* [Sharifi et al.,] partition the 45-D parameters into 6 subspaces to tackle the high dimensional problem for marker-based human pose etimation. Park *et al* [Park et al.,] decomposed the 26-D hand configurations space into six lower dimensional spaces for hand tracking, showing the advantages of this divide-and-conquer approach (Partial PSO) over existing full hand pose PSO. Poier *et al* [Poier et al.,] achieved a "significant speed-up" [Poier et al.,] by estimating the hand pose in two stages for their generative part: in the first stage the global hand location and orientation are optimized using PSO, while in the second stage each finger's pose is estimated by one PSO, and five PSOs run simultaneously.

1.3.2 Random Forest

Random forests [Tang et al., 2014, Tang et al., 2015, Wan et al., 2016] trained on public depth image datasets [Tang et al., 2014, Tompson et al., 2014a, Sun et al., 2015] have shown good performance for 3D hand pose estimation. Keskin *et al* [Keskin et al.,] proposed a multi-layered random decision forest to do hand pose estimation and hand shape classification using depth images. They first group the training data into K clusters, each of which represents one shape. A shape classification random forest is trained on the whole training set, with the label as cluster index. For each cluster, a part classification forest is trained to localize the hand parts. After obtaining the hand part labels for all the pixels, a mean-shift is used to find the hand joints. Keskin *et al* [Keskin et al., 2013] directly borrowed the idea of using random forest to do human pose estimation from [Shotton et al., 2013] into hand pose estimation. Zhu *et al* [Zhu et al., 2014] proposed a pixel-level hand detection and hand part labeling approach that exploits the contextual information from structured hand labeling in a random forest framework. Rogez *et al* [Rogez et al., 2014] was built in a tracking-by-detection framework, without manual initialization. Both hand detection and pose classification are treated as a K-way classification problem, where the classifier are used to classify K different discrete hand poses and the background. A hierarchical coarse-to-fine tree classifier [Rogez et al., 2012] is built during the training. The tree has K leaves, each corresponding to a certain pose class. Each splitting node has a part classifier used to make a coarse classification, the lower the node is in the tree, the more pose-specific the part classifier is. Tang *et al* [Tang et al., 2014] presented the Latent Regression Forest (LRF) for real-time 3D hand pose estimation from a single depth image. The input image is recursively divided into cohesive sub-regions, until each sub-region contains one joint. Li *et al* [Li et al.,] improved the Latent Regression Forest by proposing Segmentation Index Points (SIPs), using which they can generate different hand topology for different groups of hand poses. HSO [Tang et al., 2015] estimates partial poses separately in the kinematic hierarchy, where in each level a random forest is used to predict a partial hand pose. Sun *et al* [Sun et al., 2015] proposed a two-layers of a hierarchy by calculating rotation invariant pixel difference features from the whole image using random forests. Want *et al* [Wan

et al., 2016] explored a new normal difference feature, rather than pixel difference feature, in the Frame Conditioned Regression Forest to do hand pose estimation in a hierarchical manner.

1.3.3 Deep Learning

Deep Learning based methods [Ye et al., 2016, Ge et al., 2017, Wan et al., 2017, Tompson et al., 2014a] trained on large-scale public depth image datasets [Tang et al., 2014, Tompson et al., 2014a, Sun et al., 2015, Yuan et al., 2017, Garcia-Hernando et al., 2018] have shown good performance. A recent benchmark evaluation [Yuan et al., 2018] showed that modern methods achieve mean 3D joint position errors of less than 10mm, making them applicable in certain real world scenarios. Moon *et al.* [Moon et al.,] propose a 3D CNN to estimate per-voxel likelihoods for each hand joint, the network has a encoder and decoder. Ge *et al.* [Ge et al., 2016a] project the depth image onto three orthogonal planes and train a 2D CNN for each projection, then fusing the results. In [Ge et al., 2017] they propose a 3D CNN by replacing 2D projections with a 3D volumetric representation (projective D-TSDF volumes [Song and Xiao,]). The method of *RCN-3D* [Molchanov et al., 2017] is an RCN+ network [Honari et al., a], based on Recombinator Networks (RCN) [Honari et al., b] with 17 layers and 64 output feature maps for all layers except the last one, which outputs a probability density map for each of the 21 joints. *V2V-PoseNet* [Moon et al.,] uses a 3D CNN to estimate per-voxel likelihood of each joint, and a CNN to estimate the center of mass from the cropped depth map. For training, 3D likelihood volumes are generated by placing normal distributions at the locations of hand joints. Zhou *et al* [Zhou et al.,] estimates all joint angles, which are fed into a forward kinematic layer to estimate the hand joints. *mmadadi* [Madadi et al., 2017] exploited a hierarchical tree-like structured CNN, the Convolution+Relu+Pooling blocks were hierarchically branched into six branches, each of which was then followed with fully connect layers, the last layers of each branch are concatenated into one layer to predict all joints. One of the six blocks was to estimate the global orientation of the hand, which was intended to improve the palm joints' accuracy. For the rest five blocks, each was to estimate the palm with one finger to focus on a specific finger.

1.4 Literature Review

In this section, I discuss the state-of-the-art methods in the literature. Detailed comparisons are made for generative method, discriminative method, and hybrid methods. I also review the model-based 3D hand tracking, hand tracking while manipulating an object, 3D body pose estimation and contact force estimation.

1.4.1 Generative, Discriminative, and Hybrid Methods

In the computer vision society, hand tracking is to extract the hand pose for every frame in an image sequence. A hand pose usually can be expressed as a collection of the locations of the finger joints and the palm. Hand tracking is gaining momentum in both academia and industry because of its wide applications, such as human computer interaction, robotic design, avatar animation, gesture understanding, and augmented reality [Wang and Popović, 2009, Gustus et al., 2012, Sueda et al.,]. Although a lot of efforts have been put in this domain, hand tracking is still very challenging because of these problems: 1) Large dimensional configuration space of the hand pose, 2) Homogeneous color distribuion of the hand skin, 3) Frequent self occlusion and occlusion by other objects, 4) Fast motion of the hand. Figure 1 shows some typical challenging hand poses, which are difficult to track.

The most effective and accurate tools to capture hand poses are magnetic sensing devices (data gloves) [Sturman and Zeltzer, 1994, Hale and Stanney, 2002], which can be worn on the hand to capture the joint locations in real time. But it is very expensive, it needs complex calibrations, and thus hinders the free motion of the hand. Computer vision, in contrast, can provide a more natural way to do hand pose tracking. For vision-based hand tracking methods, color image [Wang and Popović, 2009], or depth image [Qian et al., a], or both [Oikonomidis et al., 2011a] are used to do the task. In order to capture the articulated 3D hand motion, multiple cameras systems [Sridhar et al., 2013] are usually used, while using only depth images can reduce the complexity [Qian et al., a].

Vision-based hand pose estimation or tracking methods in the literature can be divided into three categories: generative methods, discriminative methods, and those hybrid methods combining discriminative and generative.

1.4.1.1 Generative Method

Generative methods [Oikonomidis et al., 2011a, Oikonomidis et al., 2011b, Oikonomidis et al., 2012, Qian et al., a, Stenger et al., 2006, de La Gorce et al., , Ballan et al., , Makris et al., 2015], also called model-based methods, usually generate hand configuration hypotheses and evaluate them on available visual observations. For each frame of an image sequence, a search in the hand model configuration space is executed to find the best configuration that minimize the discrepancy between the actual observation and the hand hypothesis. Generative methods differ from each other by different models, features, kinematic constraints, cost function, and optimization techniques. Different hand models have been proposed, Oikonomidis *et al* [Oikonomidis et al., 2011a] proposed polygonal mesh model, where the palm is modeled as an elliptic cylinder and two ellipsoids, each finger is modeled by cones for the bones and spheres for joints. Qian *et al* [Qian et al., a] modeled the hand by simply using 48 spheres to accelerate the calculation of the discrepancy. Gorce *et al* [de La Gorce et al.,] used 1000 facets to build a surface mesh model. The frequently used features are edges, shading, color, optical flow, and depth [Lu et al., , de La Gorce et al., , Oikonomidis et al., 2011a]. In order to find the best hand configuration, several optimization techniques have been used. Rehg *et al* [Rehg and Kanade,] proposed one of the first approaches to estimate 3D hand pose using local optimization. Other methods rely on stochastic optimization algorithms, such as particle filter [MacCormick and Isard,] and Kalman filter [Stenger et al.,]. One of the drawbacks for them is that they can only address very limited number of hand poses. Sudderth *et al* [Sudderth et al.,] used belief propagation in a redundant hand model. Oikonomidis *et al* [Oikonomidis et al., 2011a] successfully reached the optimization through PSO. Qian *et al* [Qian et al., a] proposed a hybrid optimization approach to combine the merits and overcome the drawbacks of the both Iterated Closest Point (ICP) and PSO. Generative methods are suitable for continuous tracking through

consecutive frames with small and predictable movements, but they are computationally costly, some times they have to be implemented in GPU to reach real time performance [Oikonomidis et al., 2011a].

1.4.1.2 Discriminative Method

Discriminative methods aim at extracting the hand pose from a single image by using classification or regression techniques [Tang et al., 2013, Wang and Popović, 2009, Rogez et al., 2014, Tang et al., 2014, Ong and Bowden, , Keskin et al.,]. Learning techniques include randomized decision forests [Keskin et al.,], regression forests [Tang et al., 2013], and boosted classifier trees [Ong and Bowden,]. In order to do hand tracking, discriminative methods tend to choose the tracking-by-detection strategy, or in a image retrieval manner. Rogez *et al* [Rogez et al., 2014] proposed a egocentric hand pose detection algorithm, which was set in a tracking-by-detection framework without manual initialization. Both hand detection and pose classification are treated as a K-way classification problem, where the classifier are used to classify K different discrete hand poses and the background. Wang *et al* [Wang and Popović, 2009] treated the hand pose estimation problem as image retrieval algorithm. They obtained the pose of the hand by nearest-neighbor search in the database, and inferred the global hand position using Inverse Kinematics. Tang *et al* [Tang et al., 2014] presented the Latent Regression Forest (LRF) for real-time 3D hand pose estimation from a single depth image. Hand pose estimation is done by estimating all the joints locations of the hand. The input image is recursively divided into cohesive sub-regions, until each sub-region contains one joint. There are three main drawbacks for discriminative methods: 1) Classification-based methods can only classify a certain number of known discrete hand poses and are less suitable for continuous and accurate pose estimation of a freely performing hand, where the hand pose space is exponentially high with around 26 degrees of freedom. 2) A large amount of training data, synthetic, realistic, or combined [Tang et al., 2013], is needed to train the classifier, which is difficult due to the lack of public annotated data sets. 3) It is difficult to extend single hand pose estimation or tracking to multiple hands scenarios.

1.4.1.3 Hybrid Method

Discriminative methods can only deal with a certain number of discrete hand poses, which should be annotated in the training data sets, it is difficult to do hand tracking. Generative methods are most suitable for continuous tracking through consecutive frames with small and predictable movements, while in reality the hand pose can change very fast even between two consecutive frames. Sridhar *et al.* [Sridhar et al., 2013] proposed a hybrid approach combining a discriminative, part-based pose retrieval method with a generative pose estimation method to do hand tracking, but their method need a complex multiview RGB camera system. There is still a gap between discriminative methods and generative methods. We will address this problem and come up with a novel method that can reach the state-of-the-art performance, and hopefully outperform the state-of-the-art methods regarding to efficiency and accuracy.

Hybrid methods [Sharp et al., 2015, Sridhar et al., 2013, Krejov et al.,] combine discriminative and generative to make the hand tracking method more robust. Sharp *et al* [Sharp et al., 2015] improve the robustness of the model fitting process with per-frame reinitialization, which are obtained by a discriminative two-layer predictor. Sridhar *et al* [Sridhar et al., 2013] proposed a hybrid approach combining a discriminative part-based pose retrieval method with a generative pose estimation method to do hand tracking, but theire method need a complex multiview RGB camera system. There is still a gap between discriminative methods and generative methods. We will address this problem and come up with a novel method that can reach the state-of-the-art performance, and hopefully outperform the state-of-the-art methods regarding to efficiency and accuracy. Krejov *et al* [Krejov et al.,] presented a hand pose estimation method by combining discriminative and model based methods to overcome the limitations of each technique in isolation. They used a randomized decision forests to find an initial estimation of the hand region, which is used in a local optimization strategy using a Rigid Body Dynamics.

1.4.2 Model-based Hand Tracking

Particle Swarm Optimization is widely used for hand pose tracking from depth image. By using Particle Swarm Optimization [Clerc and Kennedy, 2002], a main obstacle is that it is difficult to get diverse initial particle to address the fast motion problems. One possible way is to use diverse hand models to generate diverse particles, for example, some particle can be extracted using hand models with strict kinematic constraints, some particles can be extracted using relatively looser kinematics constraints, other particles can be obtained by using finger tip detection methods. In order to get better particles, certain simple and fast methods can be pursued to filter out bad candidates, so we can find better particles without gaining more calculation burden. Since the hand tracking is gaining momentum recently, there still exist potential research areas. Current state-of-the-art object tracking algorithms can be borrowed into hand tracking.

1.4.3 Tracking a Hand Manipulating an Object

Manipulating an object introduces some occlusion information, i.e. certain part of the depth image is corrupted, to the hand pose estimation problem. This forces us to deal with the occlusion explicitly. Discriminative methods may have difficulty in dealing with this kind of occlusion, since they need a training stage where un-corrupted depth information is needed. Generative methods can deal with this problem more easily, since when calculating the discrepancy between the depth image and the synthesized surface we can, for example, explicitly introduce a certain threshold to indicate whether certain part is occluded or not. Hamer *et al* [Hamer et al., 2009] proposed tracking a hand manipulating an object. They divide the hand into 3 $times$ 5 + 1 = 16 segments, three segments for each finger plus one segment for the palm. Each segment has its own local tracker to sample candidates in the 6D space (position and rotation). With anatomic constraints placed between adjacent segments, they build a hand graph as a tree, whose nodes correspond to the segments. The constraints between adjacent segments obey the first order Markov property. The best hand configuration is obtained with belief propagation.

They explicitly deal with occlusion by placing a distance threshold.

1.4.4 Tracking Hand in Egocentric RGB-D Images

In a typical egocentric hand/object manipulation scenario, a camera is mounted on the chest of a viewer, who interact with an object using his/her hands [Rogez et al., 2014]. This first-person view activity is more difficult then third-person-view activity due to two facts: 1) The hands move outside the camera view frequently, leading to difficulty in using previous estimates. 2) Occlusion happens a lot more often in egocentric viewpoint than third-person viewpoint. Rogez *et al.* [Rogez et al., 2014] proposed a 3D hand pose detection algorithms using egocentric RGB-D images. But their method is built in a tracking-by-detection framework and thus limited to certain different discrete hand poses. A potential research direction is to do hand tracking in egocentric RGB-D images using a generative model.

1.4.5 3D Body Pose Estimation from Depth and RGB Images.

A large number of work focused on recovering 3D pose of people given perfect 2D pose as input [Lee and Chen, 1985, Taylor, 2000, Parameswaran and Chellappa, 2004]. They made use of the structural information of the human skeleton to recover the 3D human pose from a single image. More recent works tried to learn a prior model for the human body from 3D mocap data [Akhter and Black, 2015,Fan et al., 2014,Ramakrishna et al., 2012]. Methods to estimation 3D human body pose from images fall into two categories: (1) methods directly learn the 3D pose from images, and (2) pipeline approaches that first to estimate 2D pose from images and then lift the 2D pose to 3D pose. Earlier works learn a mapping from image features, e.g., silhouettes, HOG or SIFT, to 3D human poses [Agarwal and Triggs, 2006, Ek et al., 2007, Mori and Malik, 2006]. More recent works tried to train CNNs to predict 3D human pose directly from the image [Li and Chan, 2014, Zhou et al., 2016a, Toshev and Szegedy, 2014]. With the CNNs becoming more prevalent, 2D pose estimation has become more reliable and many recent works [Tome et al., 2017, Tompson et al., 2014b, Simo-Serra et al., 2012, Zhou et al.,

2016b] focus on the pipeline approach: first to estimate the 2D pose [Wei et al., 2016] and then to lift the 2D pose into 3D pose with the help of structured learning or graphical models.

1.4.6 3D Hand Pose Estimation From RGB Images

In the past few years, hand pose estimation from rgb images has emerged as a new trend [Simon et al., 2017, Zimmermann and Brox, 2017, Mueller et al., 2017b, Panteleris and Argyros, 2017]. Due to the lack of proper RGB datasets. Current state-of-the-art methods opt for synthetic datasets [Zimmermann and Brox, 2017], or using GAN [Goodfellow et al., 2014] to generate training data [Shrivastava et al., 2017, Mueller et al., 2017b]. Simon et.al [Simon et al., 2017] (*OpenPose*) proposed a multiview bootstrapping methods to estimation 2D hand pose by iteratively fine-tuning the estimation results. They proposed an method to generated an annotated RGB dataset using a panoptic studio camera setup. Mueller et.al [Mueller et al., 2017b] proposed to use RGB-D input to estimation 3D hand pose in an egocentric setting, where the hand is interacting with objects. Zimmermann et.al [Zimmermann and Brox, 2017] proposed an pipeline to estimate 3D hand pose from an RGB image. The pipeline has three steps: hand segmentation, 2D hand pose estimation, 3D hand pose estimation. But estimated 3D pose is relative 3D pose to an canonical pose rather than absolute 3D pose. They also proposed a synthetic dataset, even though it has perfect annotation quality, it lacks realism. Paschalis et.al [Panteleris and Argyros, 2017] tried to estimation the absolution 3D hand pose using a two step pipeline, first to estimation the 2D hand pose with a CNN, then to estimate 3D hand pose by optimizing the hand model in 3D space with inverse kinematics. Mueller et.al [Mueller et al., 2017a] do 3D hand tracking from rgb images by combining a CNN with a kinematic 3D hand model. To train the model, they used an image-to-image translation network to create large amount of semi-real (GANerated) images, which have perfect annotation and also have the same statistical distributions as real images. But their method requires a predefined hand model for each subject (person) and the hand model is obtained by a *per-user skeleton adaptation* process.

1.4.7 Contact Force Estimation

Pham et.al. [Pham et al., 2015] estimated contact forces during hand-object interactions, but do so in a "in-the-lab" scenario where objects of known geometry are used. Rogez et.al. [Rogez et al., 2015b] explored the contact the force prediction based on grasp type. Different hand poses (kinematics) tend have different forces. Another challenging issues is that, given a hand pose, it is still challenging to estimate the forces, where near-identical kinematic hand pose, but very different functional manipulations, including a wide-object grasp, a precision grasp, and a finger extension. Contact regions and force appear to define such manipulations. Estimating forces from visual signals typically requires knowledge of object mass and velocity, which is difficult to reliably infer from a single image or even a video sequence. Isometric forces are even more difficult to estimate because no motion may be observed. Finally, even traditional tasks such as kinematic hand pose estimation are now difficult because manipulated objects tend to generate significant occlusions. Direct kinetics, such as interaction forces, are generally measured with transducers and other physical objects. However, recent advances in visual tracking [Pham et al., 2017] have enabled the measurement of forces from vision, starting from mathematical models designed to predict applied normal forces from kinematic data in static multi-finger [Niu et al., 2012] grasping and two finger [Slota et al., 2011] motion.

1.5 Relationship between Chapters

This thesis makes contributions in three important aspects for 3D hand pose estimation: methods, datasets, and challenges. Chapter 1 discusses the problem definition and important useful tools, it also covers a general literature review in the field. Main contributions are covered in Chapter 2, Chapter 3, Chapter 4. Chapter 5 provides discussions, conclusions, and future works. In Chapter 2, I describe the motivation for a large scale dataset, and come up with the BigHand2.2M [Yuan et al., 2017] dataset, and also demonstrate the importance of such a dataset with cross-dataset validation. The subset including a large number of fully annotated egocentric sequences allows me to train a CNN model, and achieved significant advancement

on egocentric hand pose estimation, and reaches similar estimation accuracy as third-view hand pose estimation problem. With the BigHand2.2M dataset as well as First-Person Hand Action [Garcia-Hernando et al.,] datasets, I propose the 2017 Hands in the Million Challenge (HIM2017) (detailed in the appendix), a public competition designed for the evaluation of the task of 3D hand pose estimation. The goal of this challenge is to assess how far is the state of the art in terms of solving the problem of 3D hand pose estimation as well as detect major failure and strength modes. Detailed descriptions are provided in the appendix, *Hands In the Million Challenge*. Thanks to the *Hands In the Million Challenge*, I can analysis the state-of-the-art methods in the literature. In Chapter 3, I present detailed analysis of the top 10 among 17 participating methods from the *HIM2017* challenge [Yuan et al.,], in order to answer two questions: What is the current state of 3D hand pose estimation from depth images? And, what are the next challenges that need to be tackled? I discuss the performance of different CNN structures with regard to hand shape, joint visibility, view point and articulation distributions. Motivated by the ubiquitous presence of RGB cameras in our daily life, the lack of proper-sized RGB dataset for RGB-based hand pose estimation, Chapter 4 proposes a method for 3D hand pose estimation from RGB images that uses both external large-scale depth image datasets and paired depth and RGB images as privileged information at training time. Chapter 5 concludes the thesis and provides future research directions.

1.6 Thesis Outlines and Contributions

This thesis consists of fix chapters and an appendix section, the content of the remaining chapters are listed are summarized as follows.

1.6.1 BigHand2.2M Benchmark: Hand Pose Dataset and State of the Art Analysis

Chapter 2 proposes the BigHand2.2M [Yuan et al., 2017] dataset. We captured a new million scale standard real dataset for 3D hand pose estimation, consisting of 2.2 million fully annotated images. Depth maps are accurately annotated with 3D joint locations using a magnetic tracking system. We show that training a CNN on this data achieves accurate results. The data was used in the Hands in the Million Challenge (Chapter 3). We discuss the importance of a large scale accurately annotated real benchmark, through cross benchmark evaluation, and through achieving significant advancement on egocentric hand pose estimation. With the proposed dataset, we are able to, for the very first time in the community, achieve a similar estimation accuracy for egocentric viewpoint as that of third-view hand pose estimation problem.

Related Publication

S. Yuan, Q. Ye, B. Stenger, S. Jain, T-K. Kim, BigHand2.2M Benchmark: Hand Pose Data Set and State of the Art Analysis. **CVPR**, 2017 [Yuan et al., 2017].

1.6.2 Hand Pose from Depth: Current Achievements and Future Goals

Chapter 3 presents detailed analysis of the top 10 among 17 participating methods from the *HIM2017* challenge [Yuan et al.,], providing insights into the current state of 3D hand pose estimation and future research directions. The *Hands In the Million (HIM2017)* challenge [Yuan et al.,] serves as a test-bed for this analysis. This benchmark dataset includes data from *BigHand2.2M* [Yuan et al., 2017] and the *First-Person Hand Action dataset (FHAD)* [Garcia-Hernando et al.,], allowing the comparison of different algorithms in a variety of settings. The challenge considers three different tasks: single-frame pose estimation, tracking and hand-object interaction. We aim to answer the question of where we are, as a field, in terms of accuracy. We consider network architectures, preprocessing strategies and data representations.

Our findings include: (1) isolated 3D hand pose estimation achieves low mean errors (10 mm) in the view point range of [70, 120] degrees; (2) 3D volumetric representations outperform 2D CNNs; (3) While joint occlusions pose a challenge for most methods, explicit modeling of structure constraints can significantly narrow the gap between errors on visible and occluded joints.

Related Publication

S. Yuan, G. Garcia-Hernando, B. Stenger, G. Moon, J. Y. Chang, K. M. Lee, P. Molchanov, J. Kautz, S. Honari, L. Ge, J. Yuan, X. Chen, G. Wang, F. Yang, K. Akiyama, Y. Wu, Q. Wan, M. Madadi, S. Escalera, S. Li, D. Lee, I. Oikonomidis, A. Argyros, and T-K. Kim, Depth-Based 3D Hand Pose Estimation: From Current Achievements to Future Goals, **CVPR**, 2018, **Spotlight** [Yuan et al., 2018].

1.6.3 Pose Estimation from RGB with Depth as Privileged Information

In Chapter 4, I proposed an privileged training scheme to do 3D hand pose estimation from RGB image with privileged information from depth data. Given the fact that there is large amount of fully annotated depth data for hand pose estimation, and that there is limited available RGB training data, we investigate the usefulness of depth image to improve the performance of 3D hand pose estimation from an RGB image. We aims to do hand pose estimation from an RGB image, with the training stage helped with privileged information, the depth data. We propose three ways to use the privileged information: as an external training data for a depth-based network branch, as paired depth data to transfer supervision from the depth-based network to the RGB-based network, and as hand mask to suppress the background activations in the depth-based network. The privileged training strategy can be easily embedded into existing pose estimation methods. As an illustration, we do 2D hand pose estimation from an RGB image using a different CNN model. Results on 2D hand pose estimation, using our training strategy, is improved over state-of-the-art methods for 2D hand pose estimation from a RGB

input.

Related Publication

S. Yuan, B. Stenger, T-K. Kim, 3D Hand Pose Estimation from RGB Images with Privileged Information from Depth Data, 2018.

1.6.4 Conclusion and Future Work

In the last Chapter (Chapter 5), I draw a conclusion for this thesis. Contents of all previous chapters are summarized and future research directions are provided.

1.6.5 Appendix: Hands In the Million Challenge

The appendix presents the *Hands In the Million Challenge (HIM2017)*, where I played I leading role to organize, with the aim of assessing how far is the state of the art in terms of solving the problem of 3D hand pose estimation, and detecting major failure and strength modes of both systems and evaluation metrics that can help to identify future research directions. The major part of the used dataset is from Chapter 2, we also propose an evaluation protocol and a public competition. Through out the competition period, we have attracted 17 methods from top groups, working on hand pose estimation, across the world. Over the 6-week course of the challenge the best average 3D estimation error could be reduced from 20 to less than 10 millimeter. The competition result was presented at the 3rd International Workshop on Observing and Understanding Hands in Action, HANDS 2017, with ICCV 2017.

Related Publication

S. Yuan, Q. Ye, G. Garcia-Hernando, T-K. Kim, The 2017 Hands in the Million Challenge on 3D Hand Pose Estimation, arXiv, 2017 [Yuan et al.,].

CHAPTER 2

BIGHAND2.2M BENCHMARK: HAND POSE DATASET AND STATE OF THE ART ANALYSIS

Contents

2.1	Overview	21
2.2	Existing benchmarks	25
2.3	Full hand pose annotation	29
2.4	BigHand2.2M benchmark	33
2.5	Analysis of the state of the art	36
2.6	Discussion and conclusions	46

2.1 Overview

In this chapter, we introduce a large-scale hand pose dataset, collected using a novel capture method. Existing datasets are either generated synthetically or captured using depth sensors:

synthetic datasets exhibit a certain level of appearance difference from real depth images, and real datasets are limited in quantity and coverage, mainly due to the difficulty to annotate them. We propose a tracking system with six magnetic 6D sensors and inverse kinematics to automatically obtain 21-joints hand pose annotations of depth maps captured with minimal restriction on the range of motion. The capture protocol aims to fully cover the natural hand pose space. As shown in embedding plots, the new dataset exhibits a significantly wider and denser range of hand poses compared to existing benchmarks. Current state-of-the-art methods are evaluated on the dataset, and we demonstrate significant improvements in cross-benchmark performance. We also show significant improvements in egocentric hand pose estimation with a CNN trained on the new dataset.

There has been significant progress in the area of hand pose estimation in the recent past and a number of systems have been proposed [Keskin et al., , Liang et al., , Neverova et al., , Oberweger et al., 2015b, Oikonomidis et al., 2011a, Qian et al., b, Sharp et al., 2015, Tang et al., 2013, Li et al., , Choi et al., 2015, Jang et al., 2015, Wan et al., 2016, Khamis et al.,]. However, as noted in [Oberweger et al.,], existing benchmarks [Qian et al., b, Sun et al., 2015, Tang et al., 2014, Tompson et al., 2014a, Wetzler et al.,] are restricted in terms of number of annotated images, annotation accuracy, articulation coverage, and variation in hand shape and viewpoint.

The current state of the art for hand pose estimation employs deep neural networks to estimate hand pose from input data [Tompson et al., 2014a, Oberweger et al., 2015a, Oberweger et al., 2015b, Zhou et al., , Ge et al., 2016b, Ye et al., 2016]. It has been shown that these methods scale well with the size of the training dataset. The availability of a large-scale, accurately annotated dataset is therefore a key factor for advancing the field. Manual annotation has been the bottleneck for creating large-scale benchmarks [Qian et al., b, Sridhar et al.,]. This method is not only labor-intensive, but can also result in inaccurate position labels. Semi-automatic capture methods have been devised where 3D joint locations are inferred from manually annotated 2D joint locations [Oberweger et al.,]. Alternatives, which are still time-consuming, combine tracking a hand model and manual refinement, if necessary iterating these steps [Sun et al., 2015, Tang et al., 2014, Tompson et al., 2014a]. Additional sensors can aid automatic cap-

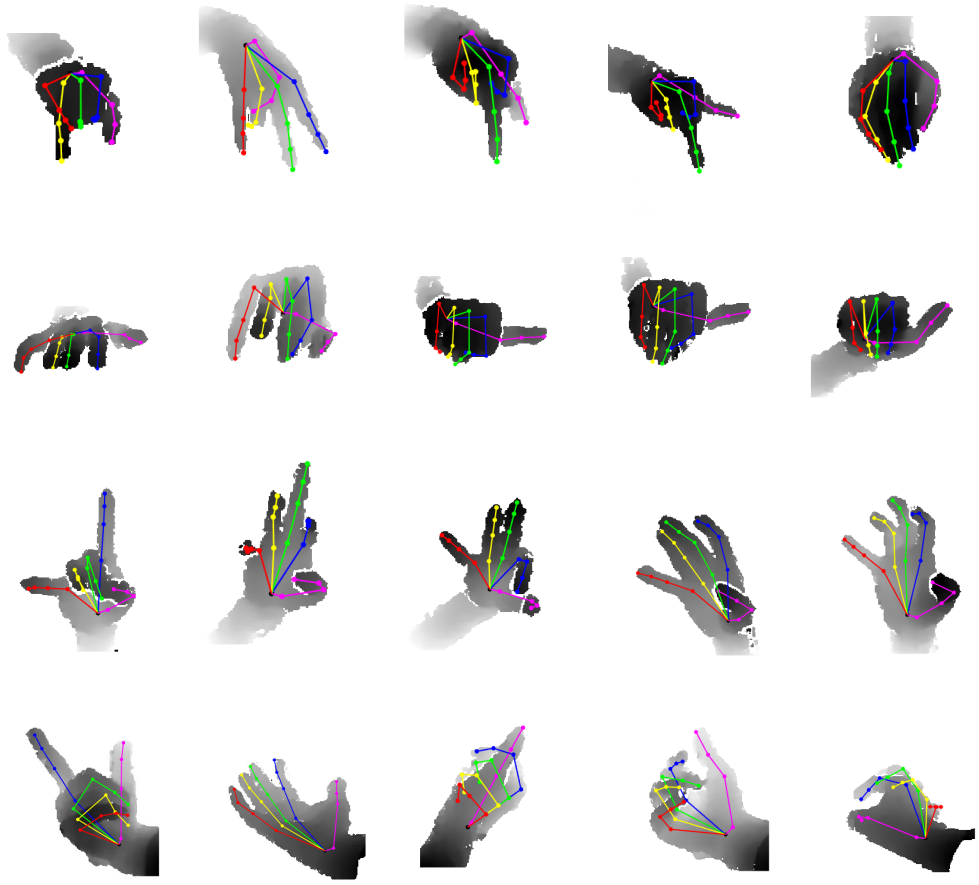


Figure 2.1: ***Example images from the BigHand2.2M dataset****. The dataset covers the range of hand poses that can be assumed without applying external forces to the hand. The accuracy of joint annotations is higher than in previous benchmark datasets.*

ture significantly, but care must be taken not to restrict the range of motion, and to minimise the depth appearance difference from bare hands, for example when using a data-glove [Xu et al., 2015]. More recently, less intrusive magnetic sensors have been employed for finger tip annotation in the *HandNet* dataset [Wetzler et al.,] .

In this chapter, we introduce our million-scale *BigHand2.2M* dataset that makes a significant advancement in terms of completeness of hand data variation and annotation quality, see Figure 2.1, Figure 2.2, Table 2.1, and Table 2.2. We detail the capture set-up and methodology that enables efficient hand pose annotation with high accuracy. This enables us to capture the range of hand motions that can be adopted without external forces. Our dataset contains 2.2

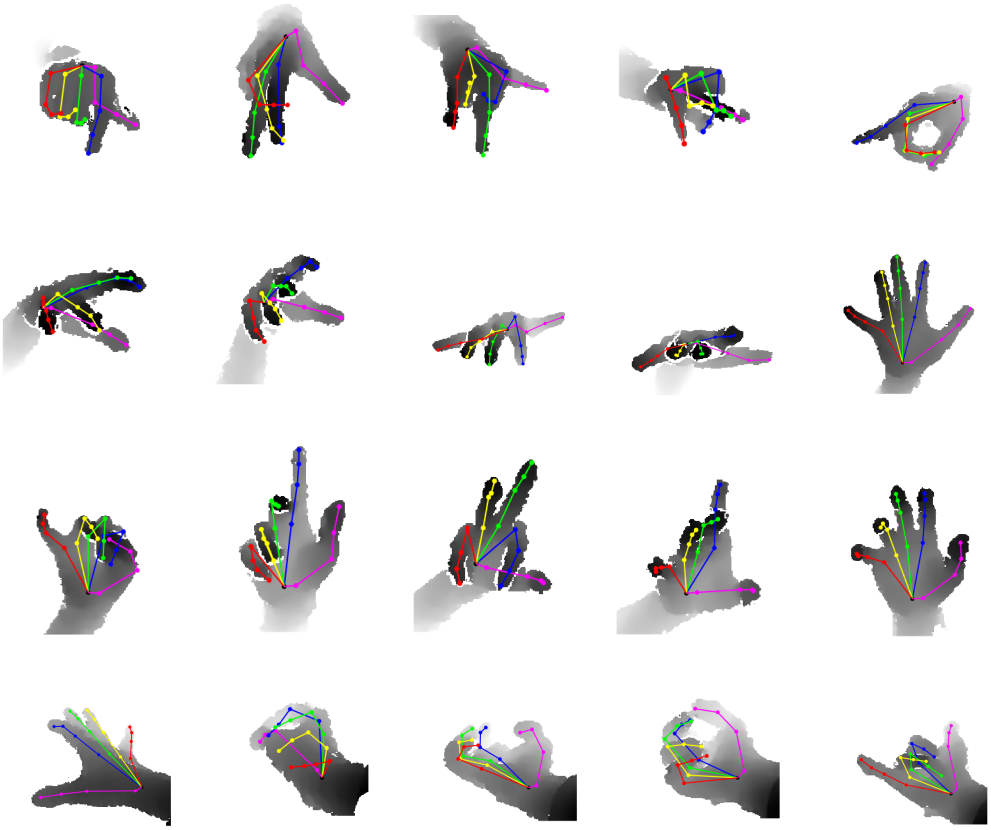


Figure 2.2: ***More example images from the BigHand2.2M dataset.***

million depth maps with accurately annotated joint locations. The data is captured by attaching six magnetic sensors on the hand, five on each finger nail and one on the back of the palm, where each sensor provides accurate 6D measurements. Locations of all joints are obtained by applying inverse kinematics on a hand model with 31 degrees of freedom (dof) with kinematic constraints. The *BigHand2.2M* dataset also contains 290K frames of egocentric hand poses, which is 130 times more than previous egocentric hand pose datasets (Table 2.4). Training a Convolutional Neural Network (CNN) on the data shows significantly improved results. The recent study by Supancic *et al.* [Supancic et al., 2015] on cross-benchmark testing showed that approximately 40% of poses are estimated with an error larger than 50mm. This is due to a different capture set-up, hand shape variation, and annotation schemes. Training a CNN using the *BigHand2.2M* dataset, we demonstrate state-of-the-art performance on existing benchmarks,

Dataset	Annotation	No. frames	No. joints
Dexter 1 [Sridhar et al.,]	manual	2,137	5
MSRA14 [Qian et al., b]	manual	2,400	21
ICVL [Tang et al., 2014]	track + refine	17,604	16
NYU [Tompson et al., 2014a]	track + refine	81,009	36
MSRA15 [Sun et al., 2015]	track + refine	76,375	21
UCI-EGO [Rogez et al., 2015a]	semi-automatic	400	26
Graz16 [Oberweger et al.,]	semi-automatic	2,166	21
ASTAR [Xu et al., 2015]	automatic	870	20
HandNet [Wetzler et al.,]	automatic	212,928	6
MSRC [Sharp et al., 2015]	synthetic	102,000	22
BigHand2.2M	automatic	2.2M	21

Table 2.1: ***Benchmark comparison***. *Existing datasets are limited in the number of frames, due to their annotation methods, either manual or semi-automatic. Our automatic annotation method allows collecting fully annotated depth images at frame rate.*

Dataset	No. subjects	View point	Depth map resolution
Dexter 1 [Sridhar et al.,]	1	3rd	320×240
MSRA14 [Qian et al., b]	6	3rd	320×240
ICVL [Tang et al., 2014]	10	3rd	320×240
NYU [Tompson et al., 2014a]	2	3rd	640×480
MSRA15 [Sun et al., 2015]	9	3rd	320×240
UCI-EGO [Rogez et al., 2015a]	2	ego	320×240
Graz16 [Oberweger et al.,]	6	ego	320×240
ASTAR [Xu et al., 2015]	30	3rd	320×240
HandNet [Wetzler et al.,]	10	3rd	320×240
MSRC [Sharp et al., 2015]	1	3rd	512×424
BigHand2.2M	10	full	640×480

Table 2.2: ***Benchmark comparison***. *Our dataset is collected with the latest Intel RealSense SR300 camera [Intel SR300,], which captures depth images at* 640×480*-pixel resolution.*

15-20mm average errors.

2.2 Existing benchmarks

Existing benchmarks for evaluation and comparison are significantly limited in scale (from a few hundred to tens of thousands), annotation accuracy, articulation, view point, and hand shape [Qian et al., b, Sharp et al., 2015, Sun et al., 2015, Tang et al., 2014, Tompson et al., 2014a, Wetzler et al., , Oberweger et al.,]. The bottleneck for building a large-scale benchmark

using captured data is the lack of a rapid and accurate annotation method. Creating datasets by manual annotation [Qian et al., b, Sridhar et al.,] is labor-intensive and can result in inaccurate labels. These benchmarks are small in size, *e.g*MSRA14 [Qian et al., b] and Dexter 1 [Sridhar et al.,] have only 2,400 and 2,137 frames, respectively, making them unsuitable for large-scale training. Alternative annotation methods, which are still labor-intensive and time-consuming, track a hand model and manually refine the results, if necessary they iterating these two steps [Sun et al., 2015, Tang et al., 2014, Tompson et al., 2014a]. ICVL [Tang et al., 2014] and NYU [Tompson et al., 2014a] benchmarks are among the first few and popularly used benchmarks. ICVL benchmark [Tang et al., 2014] has 16,008 training images, and 1,506 testing images. This benchmark is firstly annotated using 3D skeletal tracking method [Melax et al.,] and then manually refined. Even though a lot of human effort has been put into the annotation, the benchmark is still not well annotated. NYU benchmark [Tompson et al., 2014a] is a much larger and has larger range of view point, it contains 72,757 training frames and 8,252 testing frames of RGB and depth images. In order to do annotation, NYU benchmark adopted model based hand tracking using depth images from three cameras. Particle Swarm Optimization is used to find out the final annotation, but this method often drifts to wrong pose, where manual annotation is needed to restart the tracking process. The ICVL dataset [Tang et al., 2014] is one of the first benchmarks and it was annotated using 3D skeletal tracking [Melax et al.,] followed by manual refinement. However, its scale is small and limitations of the annotation accuracy have been noted in the literature [Sun et al., 2015, Oberweger et al.,]. The NYU dataset [Tompson et al., 2014a] is larger with a wider range of view points. Its annotations were obtained by model-based hand tracking on depth images from three cameras. Particle Swarm Optimization was used to obtain the final annotation. This method often drifts to incorrect poses, where manual correction is needed to re-initialize the tracking process. The MSRA15 dataset [Sun et al., 2015] is currently the most complex in the area [Oberweger et al.,]. It is annotated in an iterative way, where an optimization method [Qian et al., b] and manual re-adjustment alternate until convergence. The annotation also contains errors, such as occasionally missing finger and thumb annotations. This benchmark has a large view point coverage, but it has only small variation in articulation, capturing 17 base articulations and

varying each of them within a 500-frame sequence.

MSRA14 benchmark [Qian et al., b] is a mall benchmark aiming for model-based hand tracking method, it has 2,400 frames, containing six subjects, each of which has 400 frames. MSRA14 benchmark is manually annotated with full annotation. Due to its small size, it is only suitable for model based hand tracking. MSRA15 benchmark [Sun et al., 2015] contains 76,500 depth images of 9 subjects, 8 of which are used for training and the remaining is for testing. MSRA15 benchmark is annotated in an iterative way, where an optimization method [Qian et al., b] based annotator and manual readjustment are used iteratively until a satisfactory annotation is found. Even though this method paid a lot of attention to annotation quality, it is time and human labour consuming. If we look further into the annotation, it still contains a lot of error, such as missing annotation on a single finger, especially for the thumb. This benchmark has large view point coverage, but it has very small variations in articulation since use only 17 base articulations and each of them is almost fixed during a 500-frame sequence.

Two small datasets were captured using semi-automatic annotation methods [Oberweger et al., , Rogez et al., 2015a]. The UCI-EGO dataset [Rogez et al., 2015a] was annotated by iteratively searching for the closest example in a synthetic set and subsequent manual refinement. The Graz16 dataset [Oberweger et al.,] was annotated by iteratively annotating visible joints in a number of key frames and automatically inferring the complete sequence using an optimization method, where the appearance as well as temporal, and distance constraints are exploited. However, it remains challenging to annotate rapidly moving hands. It also requires manual correction when optimization fails. This semi-automatic method resulted in a 2,166-frame annotated egocentric dataset, which is also insufficient for large-scale training. Graz16 benchmark [Oberweger et al.,] is annotated using a semi-automatic 3D annotation method, where a human annotator is need to annotate the 2D reprojections of visible joints in some reference frames, which are automatically selected. Since this annotation method exploits the appearance, temporal, and distances constraints, it is difficult to annotate fast moving hand. It also require additional manual annotation when the inference fails. Even though this semi-automatic annotation largely reduced the human input and successfully annotated a 2,000 frames egocentric benchmark, it is still difficult to create a large scale benchmark due

to its constraints from the hand tracking methods and dependence on human input. UCI-EGO benchmark [Rogez et al., 2015a] contains 4 Sequences of 1000 frames out of which every 10th or so frame is annotated with the mixture of automated and manual annotation total annotated. This benchmark use a semi-automatic annotation technique, where a few 2D joints are labelled in the image and then a full hand pose is created by iteratively choosing the closest synthetic example from the training set and manually refining. This benchmark is only suitable for testing because of its small size.

Additional sensors can aid automatic capture significantly [Pons-Moll et al., , von Marcard et al., ,Wetzler et al., ,Xu et al., 2015], but care must be taken not to restrict the range of motion. The ASTAR dataset [Xu et al., 2015] used a *ShapeHand* data-glove [ShapeHand, 2009], but wearing the glove influences the captured hand images, and to some extent hinders free hand articulation. This benchmark has 870 fully annotated frontal view images, which are collected from 30 subjects with varying hand shapes. Each subject produces 29 unique static gestures. In their training and testing protocol, half of the images are for training, the remaining half are for testing. This benchmark is limited in scale (only 870 depth images) and accuracy, where ShapeHand glove significantly effect the real hand images, and hinder the free movement of the hand.

In the works of [Pons-Moll et al., , von Marcard et al.,], full human body pose esimtation was treated as a state estimation problem given magnetic sensor and depth data. More recently, less intrusive magnetic sensors have been used for finger tip annotation in the HandNet dataset [Wetzler et al.,], which exploits a similar annotation setting as our benchmark with trakSTAR magnetic sensors [NDI trakSTAR,]. However, this dataset only provides fingertip locations, not the full hand annotations. HandNet benchmark [Wetzler et al.,] uses a similar annotation settings as our benchmark with trakSTAR magnetic sensors [NDI trakSTAR,]. This benchmark contains depth images of 10 participants' hands non-rigidly deforming in front of a RealSense RGB-D camera while wearing the magnetic sensors. The annotations are generated by a magnetic annotation technique. The total number of frames is 212,928. For HandNet training, they randomly select 202,928 images and use the remaining 10,000 images for testing.

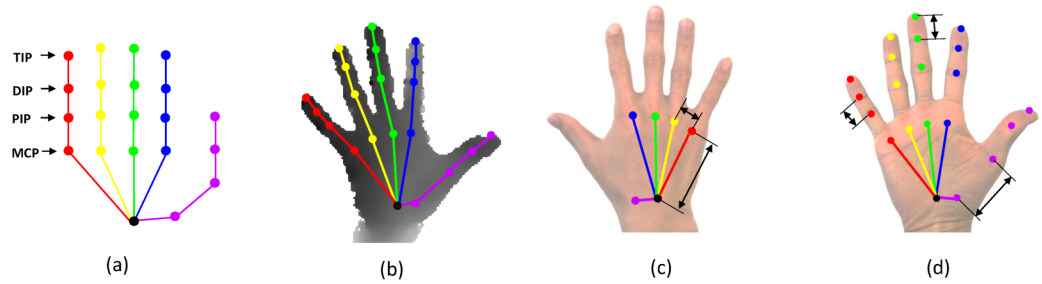


Figure 2.3: ***Hand model definition***. *(a) Our hand model has 21 joints and moves with 31 degrees of freedom (dof). (b) Model fitted to hand shape. (c) and (d) show how hand shape is measured.*

Synthetic data has been exploited for generating training data [Riegler et al., ,Rogez et al., 2015b, Xu and Cheng, 2013], as well as evaluation [Sharp et al., 2015]. Even though one can generate unlimited synthetic data, there currently remains a gap between synthetic and real data. Apart from differences in hand characteristics and the lack of sensor noise, synthetically generated images sometime produce kinematically implausible and unnatural hand poses, see Figure 2.15. The MSRC benchmark dataset [Sharp et al., 2015] is a synthetic benchmark, where data is uniformly distributed in the 3D view point space. However, the data is limited in the articulation space, where poses are generated by random sampling from six articulations.

2.3 Full hand pose annotation

In this section we present our method to do accurate full hand pose annotations using the trakSTAR tracking system with 6D magnetic sensors, see Figure 2.5.

2.3.1 Annotation by inverse kinematics

Our hand model has 21 joints and can move with 31 degrees of freedom (dof), as shown in Figure A.6. We capture 31 dimensions, including 6 dimensions for global pose and 25 joint angles. Each finger's pose is represented by five angles, including the twist angle, flexion

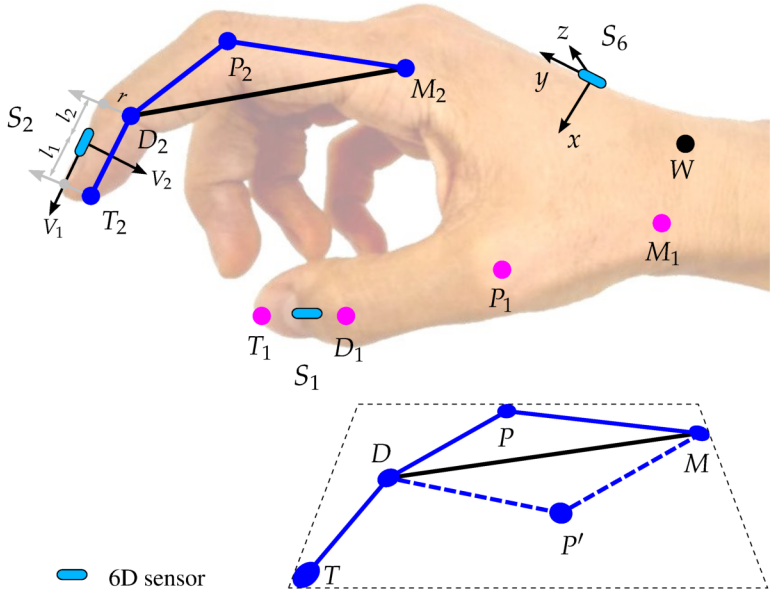


Figure 2.4: ***Hand pose inference using six 6D magnetic sensors***. *Global hand pose can be inferred from the location and orientation of sensor S6 on the back of the palm. Each sensor on the nail is used to infer the TIP and DIP joints of the corresponding finger. Each PIP joint can be calculated using bone lengths and physical constraints.*



(a) (b) (c)

Figure 2.5: ***Equipment*** *[NDI trakSTAR,]. (a) Mid-range Transmitters, (b) trakSTAR electronics unit, (c) 'Model 180' 6D magnetic sensor.*

angle, abduction angle for the MCP joint and flexion angles for the DIP and PIP joints. For each subject, we manually measure the bone lengths, see Figure A.6 (c) and (d).

Given the six magnetic sensors, each with 6D dof (location and orientation), along with a hand model, we use inverse kinematics to infer the full hand pose defined by the locations of 21 joints, as shown in Figure A.6. The physical constraints per subject are (1) the wrist and 5 MCP joints are fixed relative to each other, (2) bone lengths are constant, and (3) MCP, PIP, DIP, and TIP joints for each finger lie on the same plane.

Similar to [Schaffelhofer and Scherberger, 2012] five magnetic sensors (from thumb to little finger, the sensors are S1, S2, S3, S4, S5) are attached on the five fingers' tips. The sixth sensor (S6) is attached to the back of the palm, see Figure 2.4. Given the location and orientation of S6, as well as the hand model shape, the wrist (W) and five MCP joints (M1, M2, M3, M4, M5) are inferred. For each finger, given the sensor's location and orientation, the TIP and DIP are calculated in the following way (as shown in Figure 2.4, take the index finger as an example): the sensor's orientation is used to find the three orthogonal axes, V_1 is along the finger, V_2 is pointing forward from the finger tip. The TIP (T) and DIP (D) joint locations are calculated as:

$$T = L(S) + l_1 V_1 + r V_2, \tag{2.1}$$

$$D = L(S) - l_2 V_1 + r V_2, \tag{2.2}$$

where $L(S)$ denotes the sensor location, r is half the finger thickness, and l_1+l_2 = b, where b is the bone length connecting the DIP and TIP joints. The final joint to infer is the PIP, shown at location P in Figure 2.4, is calculated using the following conditions: (1) T, M, D are given, (2) $\|P - D\|$ and $\|P - M\|$ are fixed, (3) T, D, P, M are on the same plane, and (4) T and P should be on different sides of the line connecting M and D. These constraints are sufficient to uniquely determine P.

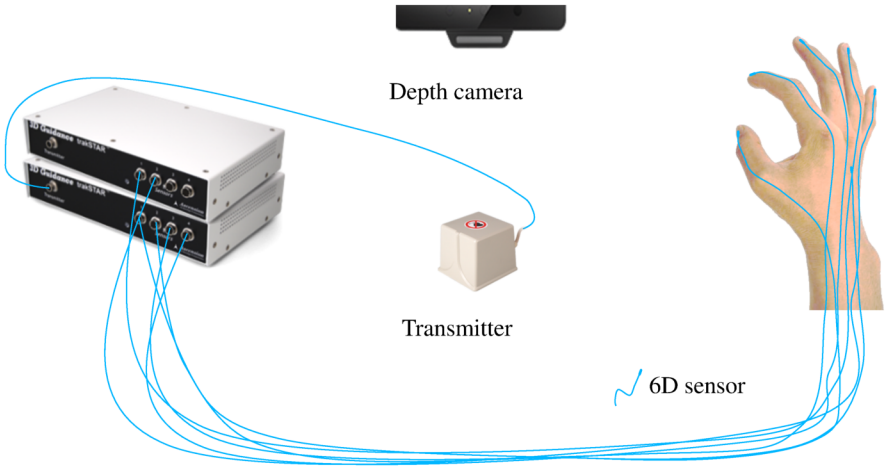


Figure 2.6: ***Annotation settings***. *The equipment used in our annotation system are: Two hardware synchronized electromagnetic tracking units, six 6D magnetic sensors, one mid-range transmitter, and an Intel RealSense SR300 camera.*

2.3.2 Synchronization and calibration

To build and annotate our dataset, we use a trakSTAR tracking system [NDI trakSTAR,] combined with the latest generation Intel RealSense SR300 depth sensor [Intel SR300,], see Figure 2.6. The trakSTAR system consists of two hardware synchronized electromagnetic tracking units, each of which can track up to four 6D magnetic sensors. The 6D sensor ("Model 180") is 2mm wide and is attached to a flexible 1.2mm wide and 3.3m long cable. When the cable is attached to the hand using tight elastic loops the depth images and hand movements are minimally affected. We use the mid-range transmitter with a maximum tracking distance of 660mm, which is suitable for hand tracking. The tracking system captures the locations and orientations of the six sensors at 720fps and is stable and without drift in continuous operation. The depth camera captures images with a resolution of 640×480 and runs at a maximum speed of 60fps. The measurements are synchronized by finding the nearest neighboring time stamps. The time gap between the depth image and the magnetic sensors in this way is 0.7 millisecond at most.

The trakStar system and the depth sensor have their own coordinate systems, and we use a

solution to the perspective-N-point problem to calibrate the coordinates as in [Wetzler et al.,]. Given a set of 3D magnetic sensor locations and the corresponding 2D locations in the depth map as well as intrinsic camera parameters, the ASPnP algorithm [Zheng et al.,] estimates the transformation between these two coordinate systems.

2.4 BigHand2.2M benchmark

We collected the *BigHand2.2M* dataset containing 2.2 million depth images of single hands with annotated joints (see Section 2.3). Ten subjects (7 male, 3 female) were captured for two hours each.

2.4.1 Hand view-point space

In order to cover diverse view points, we vary the sensor height, the subject's position and arm orientation. The view point space (a hemisphere for the 3rd person view point) is divided into 16 regions (4 regions uniformly along each of two 3D rotation axes), and subjects are instructed to carry out random view point changes within each region. In addition, our dataset collects random changes in the egocentric view point. As the t-SNE visualization in Figure 2.7 shows, our benchmark data covers a significantly larger region of the global view-point space than the ICVL and NYU dataset.

2.4.2 Hand articulation space

Similar to [Wu et al.,], we define 32 *extremal poses* as hand poses where each finger assumes a maximally bent or extended position. For maximum coverage of the articulation space, we enumerate all $\binom{32}{2} = 496$ possible pairs of these *extremal poses*, and capture the natural motion when transitioning between the two poses of each pair.

In total the *BigHand2.2M* dataset consists of three parts: (1) Schemed poses: to cover all the articulations that a human hand can freely adopt, this contains has 1.534 million frames,

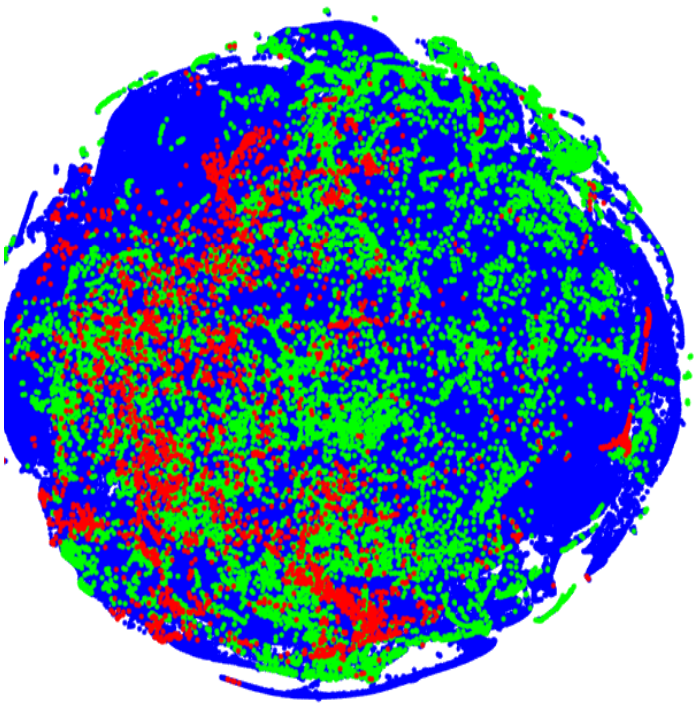


Figure 2.7: ***2D t-SNE embedding of the view point space****. BigHand2.2M is represented by blue, ICVL by red, and NYU by green dots. The figure shows global view point space coverage. Compared with existing datasets, the BigHand2.2M contains a more complete range of variation.*

captured as described above. (2) Random poses: 375K frames are captured with participants being encouraged to fully explore the pose space. (3) Egocentric poses: 290K frames of egocentric poses are captured with subjects carrying out the 32 *extremal poses* combined with random movements.

As Figure 2.8 and Figure 2.9 show, our benchmark spans a wider and denser area in the articulation and the combined of articulation and view-point space, compared to the ICVL and NYU.

2.4.3 Hand shape space

We select ten participants with different hand shapes (7 male, 3 female, age range: 25-35 years). Existing benchmarks also use different participants, but are limited in annotated hand

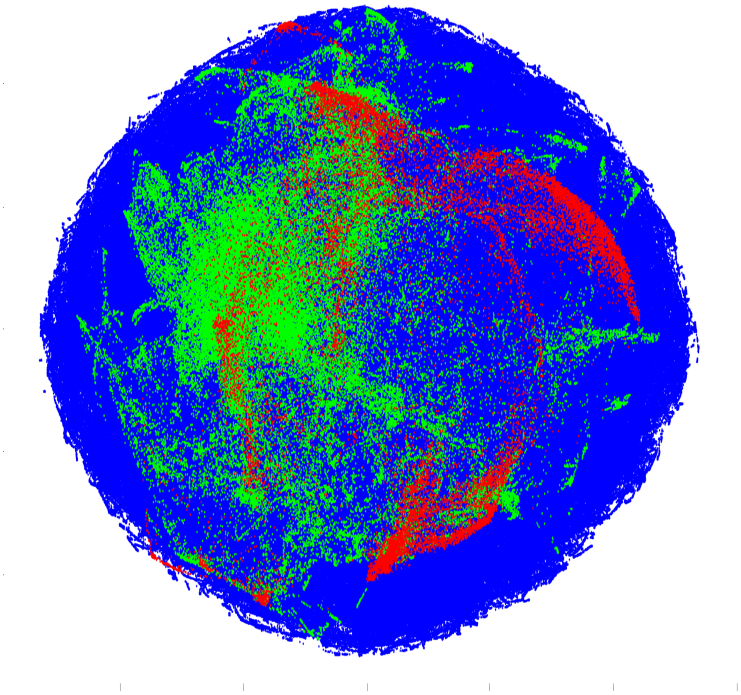


Figure 2.8: ***2D t-SNE embedding of the hand articulation space***. *BigHand2.2M is represented by blue, ICVL by red, and NYU by green dots. The figure shows articulation space (25D). Compared with existing datasets, the BigHand2.2M contains a more complete range of variation.*

shapes due to annotation methods. Figure 2.10 visualizes shapes in different datasets using the first two principal components of the hand shape parameters. The ICVL dataset [Tang et al., 2014] includes ten participants with similar hand size, and all are annotated with a single hand shape model. The NYU [Tompson et al., 2014a] training data uses one hand shape, while its test data uses two hand shapes, one of which is from the training set. The MSRA15 dataset includes nine participants, but in the annotated ground truth data, only three hand shapes are used. The MSRC [Sharp et al., 2015] synthetic benchmark includes a single shape.

In the experiments, we use the dataset of 10 subjects for training, and testify how well the learnt model generalises to different shapes in existing benchmarks (cross-benchmark) and an unseen new shape (“new person" in Figure 2.10). See section 2.5 for more explanations.

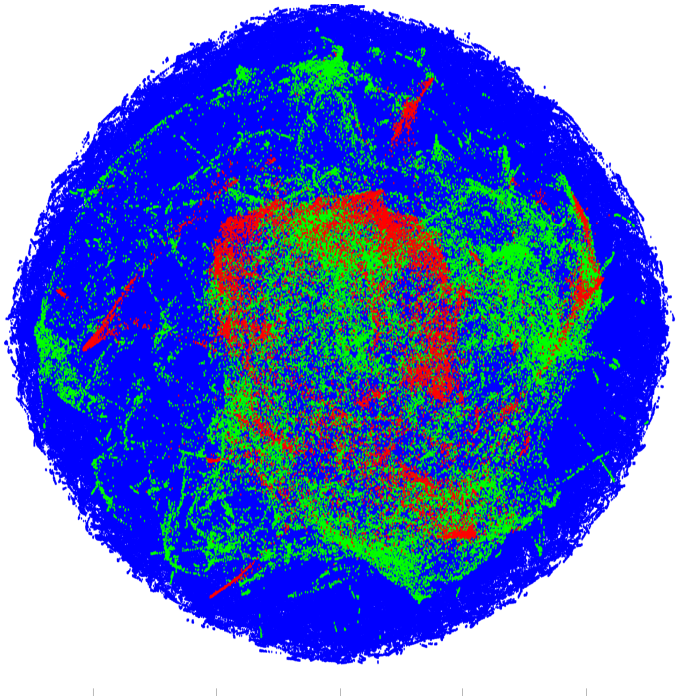


Figure 2.9: ***2D t-SNE embedding of the hand pose space****. BigHand2.2M is represented by blue, ICVL by red, and NYU by green dots. The figure shows the combined of global orientation and articulation coverage. Compared with existing datasets, the BigHand2.2M contains a more complete range of variation.*

2.5 Analysis of the state of the art

In this section we use the *Holi* CNN architecture [Ye et al., 2016] as the current state of the art. The detailed structure is shown in the supplementary material. The input for the CNN model is the cropped hand area using the ground truth joint locations. This region is normalized to 96×96 pixels and is fed into the CNN, together with two copies downsampled to 48×48 and 24×24. The cost function is the mean squared distance between the location estimates and the ground truth locations. The CNN is implemented using Theano and is trained on a desktop with an Nvidia GeForce GTX TITAN Black and a 32-core Intel processor. The model is trained using Adam, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\alpha = 0.0003$. We stop the training process when the cost of the validation set reaches the minimum, where each training epoch takes approximately 40 minutes. When training the CNN model on *BigHand2.2M*, ICVL,

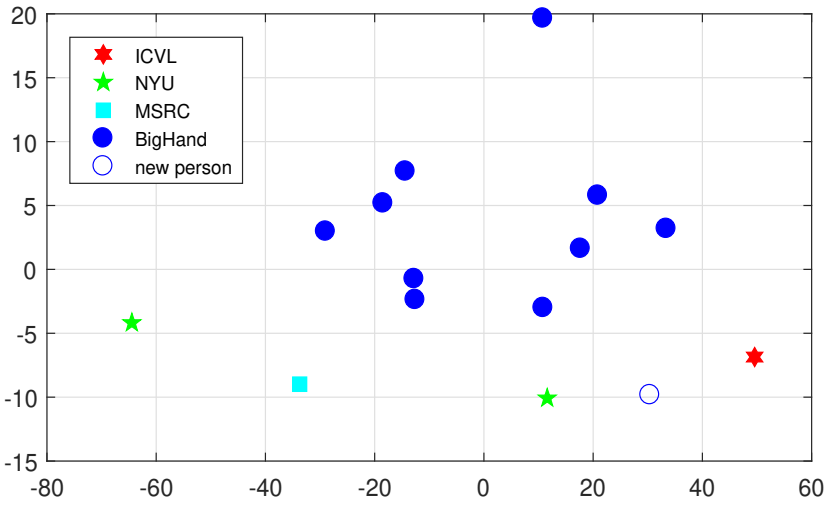


Figure 2.10: ***Hand shape variation***. *Hand variation is visualized by applying PCA to the shape parameters. The BigHand2.2M dataset contains 10 hand shapes and an additional shape for testing. The ICVL dataset contains one hand shape due to its annotation method. The NYU dataset includes two hand shapes, the MSRC dataset includes one synthetic hand shape.*

NYU and MSRC, we keep the CNN structure and β_1, β_2, α of Adam unchanged.

All frames of 10 subjects are uniformly split into a training set and a validation set with a 9-to-1 ratio, which is similar to ICVL, NYU, and HandNet [Wetzler et al.,]. In addition to the 10 subjects, a challenging test sequence of 37K frames of a new subject is recorded and automatically annotated, as shown “new person" in Figure 2.10. For a quantitative comparison, we measure the ratio of joints within a certain error bound ϵ [Ye et al., 2016, Tang et al., 2015, Sharp et al., 2015].

2.5.1 Cross-benchmark performance

Cross-benchmark evaluation is a challenging and less-studied problem in many fields, like face recognition [Parkhi et al.,] and hand pose estimation [Supancic et al., 2015]. Due to the small number of training datasets, existing hand pose estimation systems perform poorly when tested on unseen hand poses. As pointed out in [Supancic et al., 2015], in existing datasets, “test

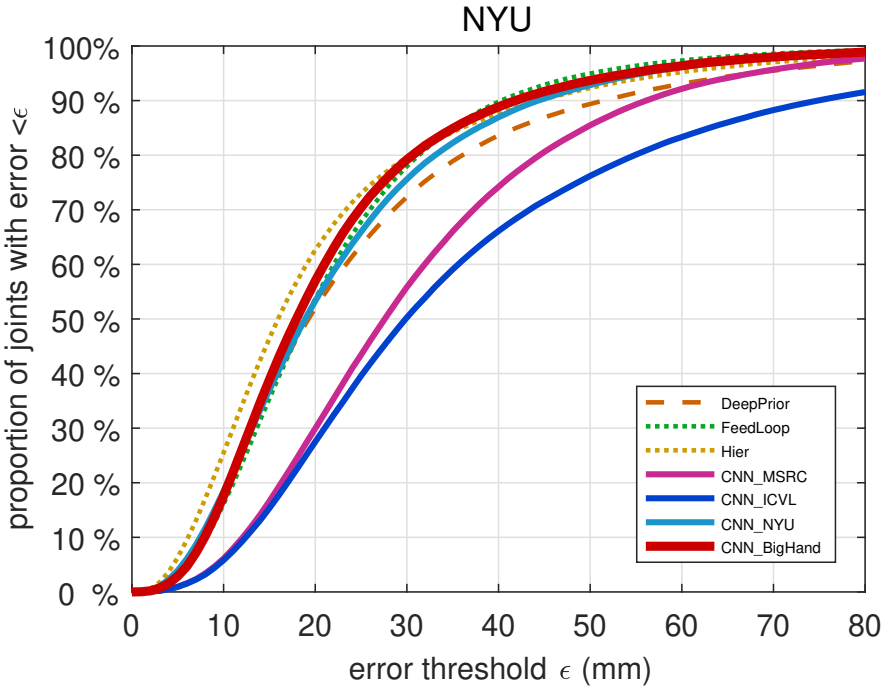


Figure 2.11: ***Cross-benchmark performance***. *CNN models are trained on the ICVL, NYU, MSRC, and the new BigHand2.2M dataset, respectively, and evaluated on NYU test data. A CNN trained on BigHand2.2M achieves state-of-the-art performance on NYU, while the CNNs trained on ICVL, NYU, and MSRC do not generalize well to other benchmarks. The networks CNN_MSRC, CNN_ICVL, CNN_NYU, and CNN_BigHand are trained on the training set of MSRC, ICVL, NYU, and BigHand2.2M, respectively.*

poses remarkably resemble the training poses", and they proposed “a simple nearest-neighbor base line that outperforms most existing systems".

Table 2.3, Figure 2.12, and Figure 2.11 show the estimation errors of the CNNs trained on ICVL, NYU, MSRC and *BigHand2.2M* when cross-tested. The performance when the CNN is trained on the*BigHand2.2M* training set is still high when evaluated on other datasets. On real test datasets (ICVL and NYU), it achieves comparable or even better performance than models trained on the corresponding training set. This confirms that with high annotation accuracy and with sufficient variation in shape, articulation and viewpoint parameters, a CNN trained on a large-scale dataset is able to generalize to new hand shapes and viewpoints, while the nearest neighbor method showed poor cross-testing performance [Supancic et al., 2015].

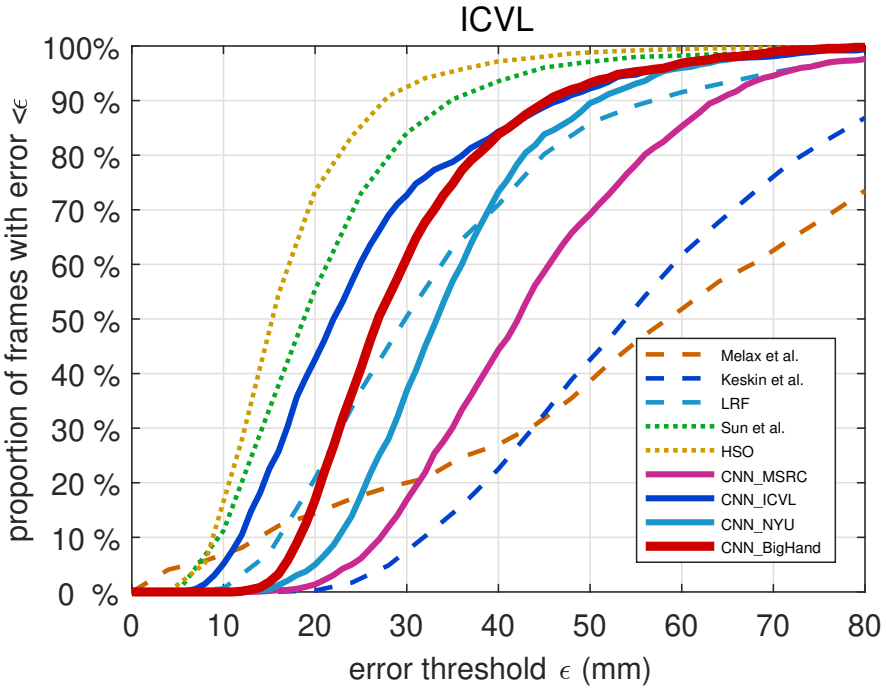


Figure 2.12: ***Cross-benchmark performance***. *CNN models are trained on the ICVL, NYU, MSRC, and the new BigHand2.2M dataset, respectively, and evaluated on ICVL test data. A CNN trained on BigHand2.2M achieves state-of-the-art performance on ICVL, while the CNNs trained on ICVL, NYU, and MSRC do not generalize well to other benchmarks. The networks CNN_MSRC, CNN_ICVL, CNN_NYU, and CNN_BigHand are trained on the training set of MSRC, ICVL, NYU, and BigHand2.2M, respectively.*

The MSRC dataset is a synthetic dataset with accurate annotations and evenly distributed viewpoints. When training the CNN on MSRC and testing on all real testing sets, the performance is worse than the CNN trained on NYU, and significantly worse than when trained on *BigHand2.2M*. Performance is to that of a CNN trained on ICVL which is only one-sixth in size compared to the MSRC training set. On the other hand, the model trained on *BigHand2.2M* shows consistently high performance across all real datasets, but poorer performance on the MSRC test set due to the differences between real and synthetic data. Figure 2.15 shows examples from the MSRC dataset. Synthetically generated images tend to produce kinematically implausible hand poses, which are difficult to assume without applying external force. There are also differences in hand shape, *e.g*the thumb appears large compared to the rest of the hand.

Increasing the amount of training data improves the performance on cross benchmark eval-

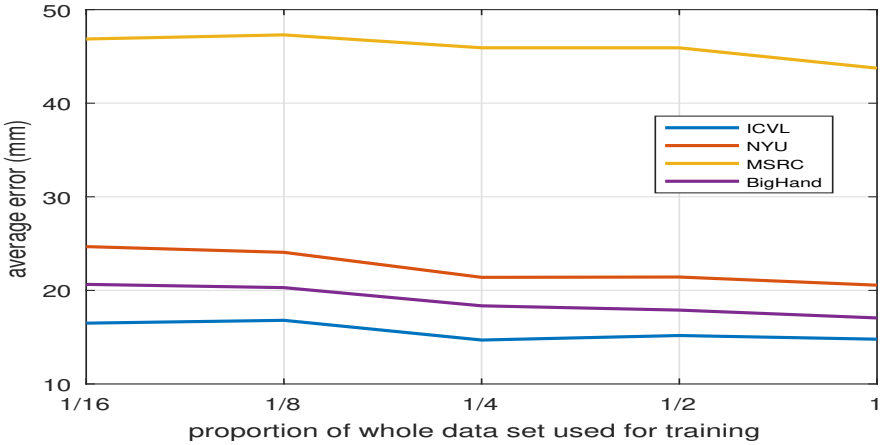


Figure 2.13: ***Data size effect on cross benchmark evaluation.*** *When the CNN model is trained on* $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, *and all of the benchmark data, the test results on ICVL, NYU, MSRC, and BigHand2.2M keep improving.*

train \ test	ICVL	NYU	MSRC	*BigHand2.2M*
ICVL	**12.3**	35.1	65.8	46.3
NYU	20.1	**21.4**	64.1	49.6
MSRC	25.3	30.8	**21.3**	49.7
BigHand2.2M	**14.9**	**20.6**	**43.7**	**17.1**

Table 2.3: ***Cross-benchmark comparison***. *Mean errors of CNNs trained on ICVL, NYU, MSRC and BigHand2.2M when cross-tested. The model trained on BigHand2.2M performs well on ICVL and NYU, less so on the synthetic MSRC data. Training on ICVL, NYU, or MSRC does not generalize well to other datasets.*

uation, see Figure 2.13. In this experiment, we uniformly subsample fractions of $\frac{1}{16}$, $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, and 1 from the training and validation data, respectively. When we train CNNs with the increasing portions of *BigHand2.2M* and test them on ICVL, NYU, MSRC, and *BigHand2.2M*'s test sequences, the performance is fairly improved. These observations support that the larger amount of training data enables CNNs to better generalize to new unseen data. Also, note our dataset is dense such that the random small fractions of the training data still delivers the good accuracies.

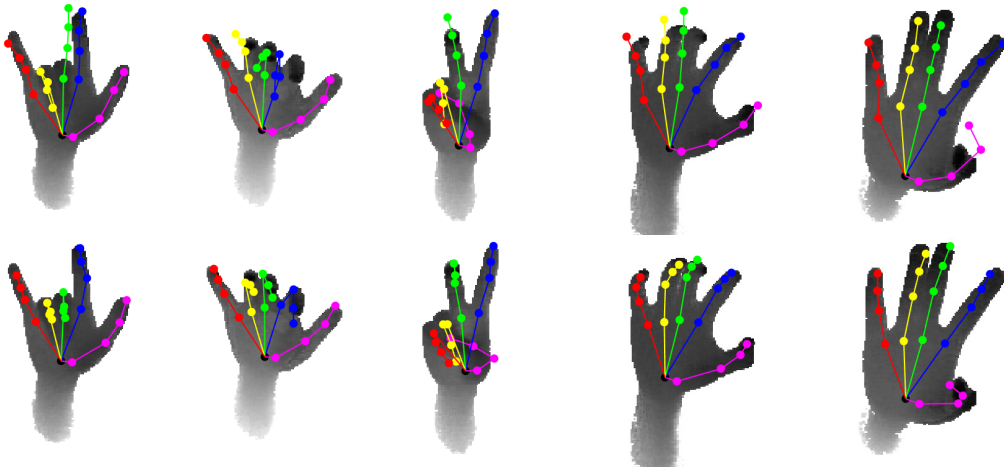


Figure 2.14: ***Generalization of the CNN trained on BigHand2.2M****. The CNN generalizes to the ICVL dataset with a lower error than the original annotated ground truth. (top) ICVL ground truth annotations, (bottom) our estimation results.*

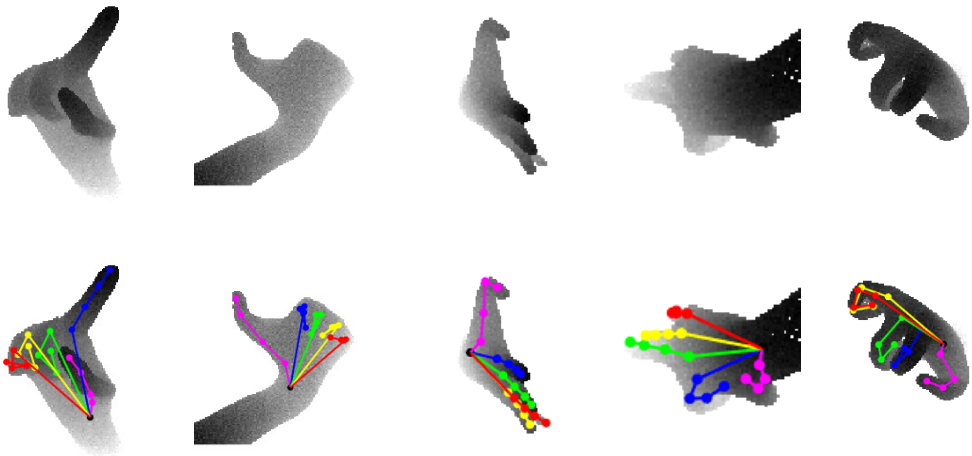


Figure 2.15: ***MSRC benchmark examples****. Synthetic data lacks real hand shape and sensor noise, and tends to have kinematically implausible hand poses. The top row shows some depth images, the bottom row shows the corresponding ground truth annotation.*

2.5.2 State-of-the-art comparison

In this section we compare our CNN model trained on *BigHand2.2M* with 8 state-of-the-art methods including HSO [Tang et al., 2015], Sun *et al.* [Sun et al., 2015], Latent Regression

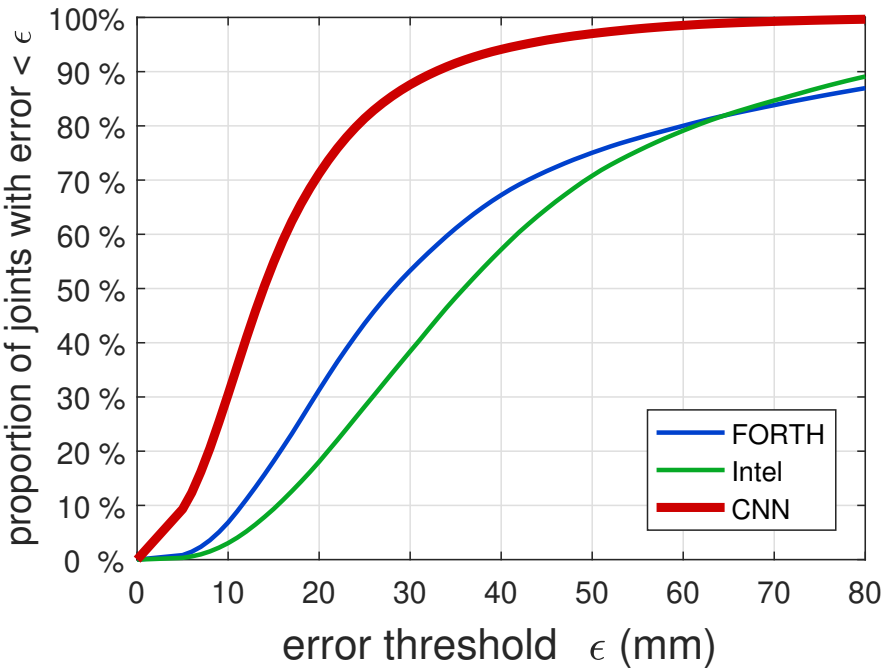


Figure 2.16: ***Hand pose estimation performance****.Baseline performances on the new subject's 37K frames of hand images. The Holi CNN significantly outperforms the tracking-based methods FORTH [Oikonomidis et al., 2011a] and Intel [Intel SR300,].*

Forest (LRF) [Tang et al., 2014], Keskin *et al.* [Keskin et al.,], Melax *et al.* [Melax et al.,], DeepPrior [Oberweger et al., 2015a], FeedLoop [Oberweger et al., 2015b], and Hier [Ye et al., 2016].

When the CNN model trained on *BigHand2.2M* is used for testing on NYU, it outperforms two recent methods, DeepPrior [Oberweger et al., 2015a] and FeedLoop [Oberweger et al., 2015b], and achieves comparable accuracy with Hier [Ye et al., 2016], even though the model has never seen any data from NYU benchmark, demonstrated in the right of Figure 2.11. Since the annotation scheme of NYU is different from ours, we choose a common (still deviated to a certain degree) subset of 11 joint locations for this comparison. We expect better results for consistent annotation schemes.

The ICVL test error curve of the CNN model trained on *BigHand2.2M* is shown in Figure 2.12. We choose the maximum allowed error [Tang et al., 2015] metric. Although it does not appear as good as that trained on ICVL itself, HSO and Sun et al., it outperforms the other

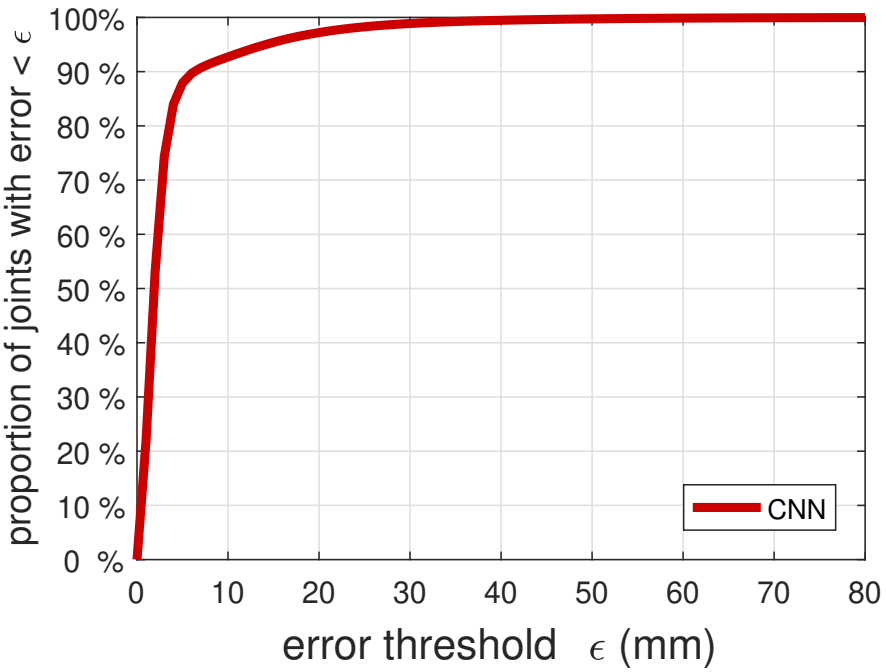


Figure 2.17: ***Hand pose estimation performance***. *The CNN trained on 90% of the BigHand2.2M data achieves high accuracy on the remaining 10% validation images.*

methods. Note that the mean estimation error for our CNN model is already as low as 14mm, which means that a small annotation discrepancy between training and test data will have a large influence on the result. As noted in [Oberweger et al.,], the annotation of ICVL is not as accurate as that of NYU. Many frames of our estimation results look plausible, but result in larger estimation errors because of inaccurate annotations, see Figure 2.14 for qualitative comparisons. Another reason is that the hand measurement scheme is different from ours. In our dataset, each subject's hand shape is determined by manually measuring joint distances. In ICVL, the same synthetic model is used for all subjects and the MCP joints tend to slide towards the fingers rather than remaining on the physical joints.

2.5.3 Baselines on BigHand2.2M

Three baselines are evaluated on our 37K-frame testing sequence, the CNN trained on *BigHand2.2M*, the Particle Swarm Optimization method (FORTH) [Oikonomidis et al., 2011a]

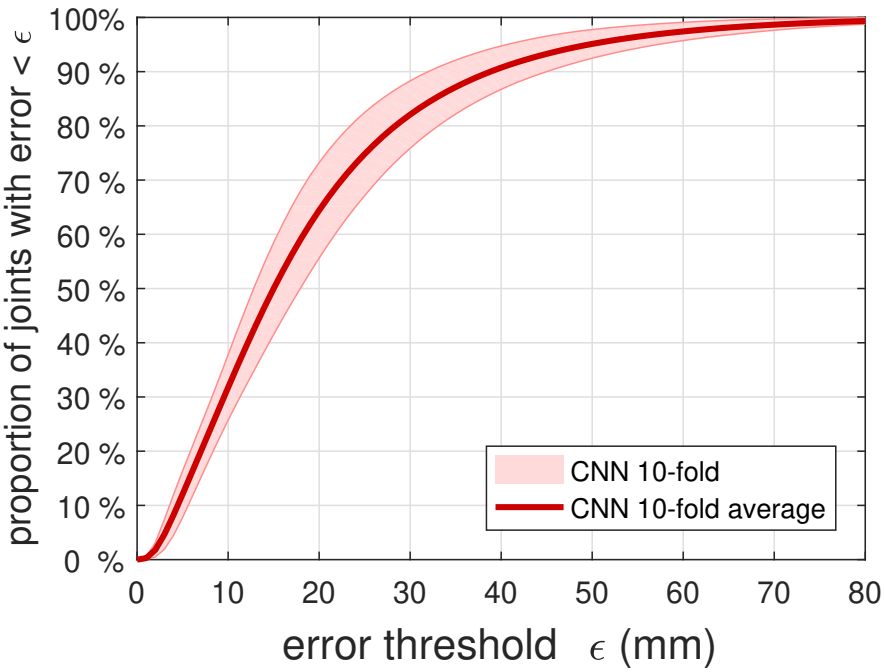


Figure 2.18: ***Hand pose estimation performance***. *10-fold cross-validation result when using the CNN for egocentric hand pose estimation. We achieved a similar-level accuracy to that of third-view hand pose estimation.*

Benchmarks	Rogez [Rogez et al., 2015a]	Oberweger [Oberweger et al.,]	BigHand2.2M Egocentric
No. Frames	400	2166	290K

Table 2.4: *Egocentric Benchmark size comparison. The egocentric subset of BigHand2.2M dataset is 130 time larger than the next largest available dataset.*

and the method by Intel [Intel SR300,]. The latter two are generative tracking methods. The CNN model outperforms the two generative methods, see the left plot of Figure 2.16. As described in the above, we chose a size ratio between training and validation sets of 9:1. Figure 2.17 shows the result on the validation set, where 90% of the joints can be estimated within a 5mm error bound.

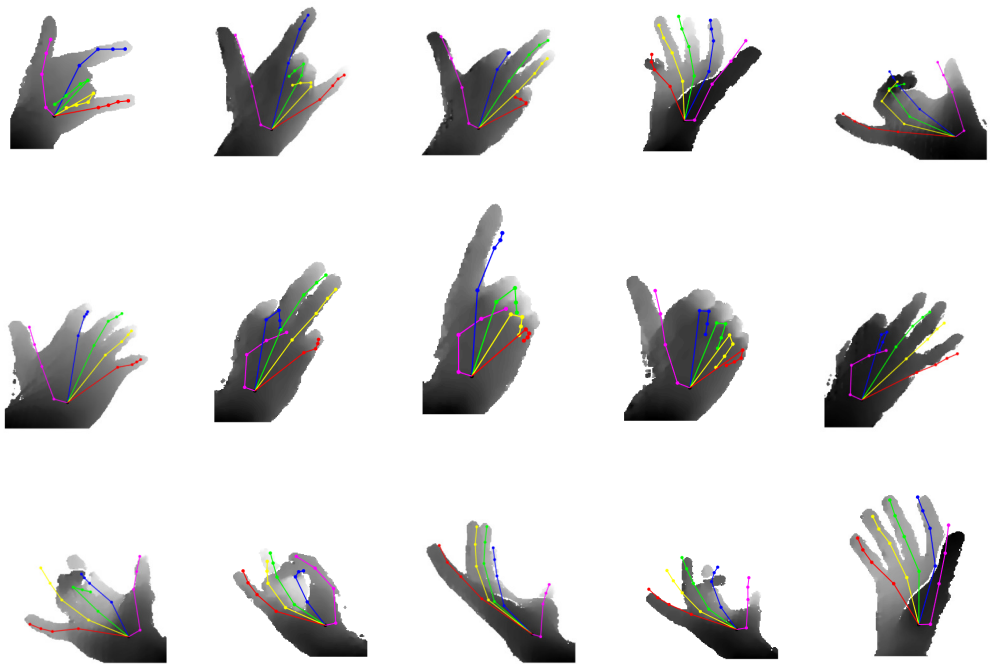


Figure 2.19: ***Qualitative results on the egocentric-view dataset.*** *A CNN trained on BigHand2.2M achieves state-of-the-art performance in the egocentric-view pose estimation task.*

2.5.4 Egocentric dataset

The lack of a large-scale annotated dataset has been a limiting factor for egocentric hand pose estimation. Existing egocentric benchmarks [Rogez et al., 2015a, Oberweger et al.,] are small, see Table 2.4. Rogez *et al* [Rogez et al., 2015a] provide 400 frames and Oberwerger *et al* [Oberweger et al.,] provide 2,166 annotated frames. The *BigHand2.2M* egocentric subset contains 290K annotated frames of ten subjects (29K frames each). This dataset enabled us to train a CNN model resulting in performance competitive with that of third view hand pose estimation. In the experiment we train the CNN on nine subjects and test it on the remaining one. This process is done with 10-fold cross validation. We report mean and standard deviation of the ten folds, see Figure 2.18. Figure 2.19 shows qualitative results.

2.6 Discussion and conclusions

Hand pose estimation has attracted a lot of attention and some high-quality systems have been demonstrated, but the development in datasets still lagged behind the algorithm advancement. To close this gap we captured a million-scale benchmark dataset of real hand depth images. For automatic annotation we proposed using a magnetic tracking system with six magnetic 6D sensors and inverse kinematics. To build a thorough yet concise benchmark, we systematically designed a hand movement protocol to capture the natural hand poses. The *BigHand2.2M* dataset includes approximately 290K frames captured from an egocentric view to facilitate the advancement in the area of egocentric hand pose estimation. Current state-of-the-art methods were evaluated using the new benchmark, and we demonstrated significant improvements in cross-benchmark evaluations. It is our aim that the dataset will help to further advance the research field, allowing the exploration of new approaches.

CHAPTER 3

HAND POSE FROM DEPTH: CURRENT ACHIEVEMENTS AND FUTURE GOALS

Contents

3.1	Overview	47
3.2	Related work.	50
3.3	Evaluation tasks	51
3.4	Evaluated methods	51
3.5	Results	57
3.6	Discussion and conclusions	79

3.1 Overview

In this chapter, we strive to answer two questions: What is the current state of 3D hand pose estimation from depth images? And, what are the next challenges that need to be tackled? Following the successful *Hands In the Million Challenge (HIM2017)*, we investigate the top 10



Figure 3.1: ***Evaluated tasks***. *For each scenario the goal is to infer the 3D locations of the 21 hand joints from a depth image. In* ***Single frame pose estimation*** *(top) and the* ***Interaction task*** *(bottom), each frame is annotated with a bounding box. In the* ***Tracking task*** *(middle), only the first frame of each sequence is fully annotated.*

state-of-the-art methods on three tasks: single frame 3D pose estimation, 3D hand tracking, and hand pose estimation during object interaction. We analyze the performance of different CNN structures with regard to hand shape, joint visibility, view point and articulation distributions. Our findings include: (1) isolated 3D hand pose estimation achieves low mean errors (10 mm) in the view point range of [70, 120] degrees, but it is far from being solved for extreme view points; (2) 3D volumetric representations outperform 2D CNNs, better capturing the spatial structure of the depth data; (3) Discriminative methods still generalize poorly to unseen hand shapes; (4) While joint occlusions pose a challenge for most methods, explicit modeling of structure constraints can significantly narrow the gap between errors on visible and occluded joints.

The field of 3D hand pose estimation has advanced rapidly, both in terms of accuracy [Baek et al., , Choi et al., a, Choi et al., b, Cihan Camgoz et al., , Deng et al., 2017, Mueller et al., ,Oberweger et al., ,Remelli et al., ,Simon et al., ,Tagliasacchi et al., 2015,Tang et al., 2015,Tang et al., 2013, Wan et al., , Wan et al., 2016, Ye et al., 2016, Zhang et al., 2016, Zimmermann and Brox, 2017] and dataset quality [Garcia-Hernando et al., ,Madadi et al., , Sun et al., 2015, Tang et al., 2017, Tompson et al., 2014a, Yuan et al., 2017]. Most successful methods treat the estimation task as a learning problem, using random forests or convolutional neural networks (CNNs). However, a review from 2015 [Supancic et al., 2015] surprisingly concluded that *a simple nearest-neighbor baseline outperforms most existing systems*. It concluded that *most systems do not generalize beyond their training sets* [Supancic et al., 2015], highlighting the need for more and better data. Manually labeled datasets such as [Qian et al., b, Sridhar et al.,] contain just a few thousand examples, making them unsuitable for large-scale training. Semi-automatic annotation methods, which combine manual annotation with tracking, help scaling the dataset size [Sun et al., 2015, Tang et al., 2014, Tompson et al., 2014a], but in the case of [Tang et al., 2014] the annotation errors are close to the lowest estimation errors. Synthetic data generation solves the scaling issue, but has not yet closed the realism gap, leading to some kinematically implausible poses [Sharp et al., 2015].

A recent study confirmed that cross-benchmark testing is poor due to different capture set-ups and annotation methods [Yuan et al., 2017]. It showed that training a standard CNN on a million-scale dataset achieves state-of-the-art results. However, the estimation accuracy is not uniform, highlighting the well-known challenges of the task: variations in view point and hand shape, self-occlusion, and occlusion caused by objects being handled.

In this chapter we analyze the top methods of the *HIM2017* challenge [Yuan et al.,]. The benchmark dataset includes data from *BigHand2.2M* [Yuan et al., 2017] and the *First-Person Hand Action dataset (FHAD)* [Garcia-Hernando et al.,], allowing the comparison of different algorithms in a variety of settings. The challenge considers three different tasks: single-frame pose estimation, tracking, and hand-object interaction. In the evaluation we consider different network architectures, preprocessing strategies, and data representations. Over the course of the challenge the lowest mean 3D estimation error could be reduced from 20 millimeter to less

Number of	Train	Test single	Test track	Test interact
frames	957K	295K	294K	2,965
subjects (unseen)	5	10 (5)	10 (5)	2 (0)

Table 3.1: ***Data set sizes and number of subjects.***

than 10 millimeter. This chapter analyzes the errors with regard to seen and unseen subjects, joint visibility, and view point distribution. We conclude by providing insights for designing the next generation of methods.

3.2 Related work.

Public benchmarks and challenges in other areas such as ImageNet [Russakovsky et al., 2015] for scene classification and object detection, PASCAL [Everingham et al., 2015] for semantic and object segmentation, and the VOT challenge [Kristan et al., 2015] for object tracking, have been instrumental in driving progress in their respective field. In the area of hand tracking, the review from 2007 by Erol *et al* [Erol et al., 2007] proposed a taxonomy of approaches. Learning-based approaches have been found effective for solving single-frame pose estimation, optionally in combination with hand model fitting for higher precision, *e.g.*, [Taylor et al., 2016]. The review by Supancic *et al* [Supancic et al., 2015] compared 13 methods on a new dataset and concluded that deep models are well-suited to pose estimation [Supancic et al., 2015]. It also highlighted the need for large-scale training sets in order to train models that generalize well. In this chapter we extend the scope of previous analyses by comparing deep learning methods on a large-scale dataset, carrying out a fine-grained analysis of error sources and different design choices.

3.3 Evaluation tasks

We evaluate three different tasks on a dataset containing over a million annotated images using standardized evaluation protocols. Benchmark images are sampled from two datasets: *BigHand2.2M* [Yuan et al., 2017] and *First-Person Hand Action dataset (FHAD)* [Garcia-Hernando et al.,]. Images from *BigHand2.2M* cover a large range of hand view points (including third-person and first-person views), articulated poses, and hand shapes. Sequences from the *FHAD* dataset are used to evaluate pose estimation during hand-object interaction. Both datasets contain 640×480-pixel depth maps with 21 joint annotations, obtained from magnetic sensors and inverse kinematics. The 2D bounding boxes have an average diagonal length of 162.4 pixels with a standard deviation of 40.7 pixels. The evaluation tasks are 3D single hand pose estimation, *i.e*, estimating the 3D locations of 21 joints, from (1) individual frames, (2) video sequences, given the pose in the first frame, and (3) frames with object interaction, *e.g*, with a juice bottle, a salt shaker, or a milk carton. See Figure A.2 for an overview. Bounding boxes are provided as input for tasks (1) and (3). The training data is sampled from the *BigHand2.2M* dataset and only the interaction task uses test data from the *FHAD* dataset. See Table A.1 for dataset sizes and the number of total and unseen subjects for each task. The training data contains 957K frames captured from 5 subjects. Single frame pose estimation and tracking are evaluated on 5 seen and 5 unseen subjects.

3.4 Evaluated methods

We evaluate the top 10 among 17 participating methods [Yuan et al.,]. The ten methods are all Deep Learning based, they are: *V2V-PoseNet* [Moon et al.,], *RCN-3D* [Molchanov et al., 2017], *oasis* [Ge et al., 2018], *THU VCLab* (Pose-REN) [Chen et al., 2017a], *NAIST_RV* [Yang et al., 2017], *Vanora* [Ge and Yuan, 2017], *strawberryfg* [Sun et al., 2017], *rvhand* [Akiyama et al., 2017], *mmadadi* [Madadi et al., 2017], and *LSL* [Li and Lee, 2017]. We also indirectly evaluated *DeepPrior* [Oberweger et al., 2015a], *DeepModel* [Zhou et al.,], and *REN* [Guo

Method	Model	Input	Aug. range (s,θ,t)
V2V-PoseNet [Moon et al.,]	3D CNN, per-voxel likelihood of each joint	88×88×88 voxels	[0.8, 1.2] [-40,40] [-8,8]
RCN-3D [Molchanov et al., 2017]	RCN+ network [Honari et al., a] with 17 convolutional layers	80×80	[0.7, 1.1] [0, 360] [-8,8]
oasis [Ge et al., 2018]	Hierarchical PointNet.	1024 3D points	random scaling*
THU_VCLab [Chen et al., 2017a]	Pose-REN [Chen et al., 2017a]: *REN* [Guo et al., 2017] + cascaded + hierarchical.	96×96	[0.9,1.1] [-45,45] [-5,5]
NAIST_RV [Yang et al., 2017]	3D CNN with 5 branches, one for each finger	50×50×50 3D grid	[0.9,1.1] [-90, 90] [-15,15]
Vanora [Ge and Yuan, 2017]	shallow CNN trained end-to-end	resized 2D	random scaling*
strawberryfg [Wan, 2017]	ResNet-152 + [Sun et al., 2017]	224×224	None
rvhand [Akiyama et al., 2017]	ResNet [He et al., 2016] + *REN* [Guo et al., 2017] + Deep Prior [Oberweger et al., 2015a]	192×192	[0.9,1.1] [-90, 90] [-15,15]
mmadadi [Madadi et al., 2017]	Hierarchical tree-like structured CNN [Madadi et al., 2017]	192×192	[random] [-30, 30] [-10,10]
LSL [Li and Lee, 2017]	ScaleNet to estimate hand scale + *DeepModel* [Zhou et al.,]	128×128	[0.85,1.15] [0,360] [-20,20]

Table 3.2: ***Methods evaluated in the hand pose estimation challenge.*** *Methods are ordered by average error on the leader-board. * in both methods, hand segmentation is performed considering different hand arm lengths.*

et al., 2017], since *DeepPrior* and *REN* are essential parts of *rvhand* [Akiyama et al., 2017], *DeepModel* [Zhou et al.,] is the backbone of *LSL* [Li and Lee, 2017]. We also indirectly evaluate *DeepPrior* [Oberweger et al., 2015a] and *REN* [Guo et al., 2017], which are components of *rvhand* [Akiyama et al., 2017], as well as *DeepModel* [Zhou et al.,], which is the backbone of *LSL* [Li and Lee, 2017]. Table 3.2 and Table 3.3 list the methods with some of their key properties. We group methods based on different design choices, see Figure 3.2 and Figure 3.3. We use both standard error metrics [Oikonomidis et al., 2011a, Taylor et al., 2012] and new proposed metrics to provide further insights.

2D CNN *vs*3D CNN. 2D CNNs have been popular for 3D hand pose estimation [Akiyama et al., 2017, Chen et al., 2017a, Ge and Yuan, 2017, Guo et al., 2017, Li and Lee, 2017, Madadi et al., 2017, Molchanov et al., 2017, Oberweger et al., 2015a, Ye et al., 2016, Zhou et al.,]. Common pre-processing steps include cropping and resizing the hand volume by normalizing

Method	**Hyper-parameters, Optimization method**	**GPU, Framework, Time (train, test)**
V2V-PoseNet [Moon et al.,]	MT 0.99, LR 2.5e-4, WD 1e-8, MB 8, EP 6, random init, RMS prop	Titan X×2, Torch7 150 hours, 50fps
RCN-3D [Molchanov et al., 2017]	LR 2.0e-4 to 1.0e-5, EP 31, random init, MSE loss, Adam	Titan X, Theano 80 hours, 55 fps
oasis [Ge et al., 2018]	MT 0.9, LR 2.0e-2 decreasing, WD 5.0e-4, MB 64, EP 20, random init, SGD	1080×2, PyTorch 70 hours, 47 fps
THU_VCLab [Chen et al., 2017a]	MT 0.9, LR 1.0e-3 decreasing, WD 5.0e-4, MB 128, L1 loss, SGD	Titan X, Caffe 84.5 hours, 92 fps
NAIST_RV [Yang et al., 2017]	MB 32, EP 20, L1 loss, Adam	Titan X, Tensorflow 160 hours, 16 fps
Vanora [Ge and Yuan, 2017]	MT 0.9, LR 1.0e-2 decreasing, WD 5.0e-4, MB 64, EP 60, random init, SGD	1080×2, PyTorch 145 hours, 54 fps
strawberryfg [Wan, 2017]	MT 0.9, LR 1.0e-3 decreasing, WD 2.0e-4, MB 20, SGD	1080Ti×2, Caffe 8.5 hours, 77 fps
rvhand [Akiyama et al., 2017]	MB 192, WD 5.0e-6, EP 90, Adam	1080Ti, Tensorflow
mmadadi [Madadi et al., 2017]	MT 0.9, LR 5.0e-5, WD 5.0e-4, MB 50, random init, SGD	Titan X, MatConvNet 40hours, 50fps
LSL [Li and Lee, 2017]	LR 1e-5, MB 32, Xavier init, L2 loss, Adam	GTX 660, 50 hours, 23 fps

Table 3.3: ***Methods evaluated in the hand pose estimation challenge.*** *Methods are ordered by average error on the leaderboard. For hyper-parameters, 'MT' is momentum, 'LR' is learning rate, 'WD' is weight decay, 'MB' is Mimi-batch size, 'EP' is total epoch number, SGD is 'Stochastic Gradient Descent'.*

the depth values to [-1, 1]. For example, the input for *RCN-3D* [Molchanov et al., 2017] is the cropped depth image resized to 80×80 pixels (see Table 3.2, third column), while the input for *THU_VCLab* [Chen et al., 2017a] is a cropped depth image of size 96×96. Recently, several methods have used a 3D CNN [Ge et al., 2017, Moon et al., , Yang et al., 2017], where the input can be a 3D voxel grid [Moon et al., , Yang et al., 2017], or a projective D-TSDF volume [Ge et al., 2017]. Ge *et al.* [Ge et al., 2016a] project the depth image onto three orthogonal planes and train a 2D CNN for each projection, then fusing the results. In [Ge et al., 2017] they propose a 3D CNN by replacing 2D projections with a 3D volumetric representation (projective D-TSDF volumes [Song and Xiao,]). In the *HIM2017* challenge [Yuan et al.,], they apply a 3D deep learning method [Ge et al., 2018], where the inputs are 3D points and surface normals. Moon *et al.* [Moon et al.,] propose a 3D CNN to estimate per-voxel likelihoods for each hand

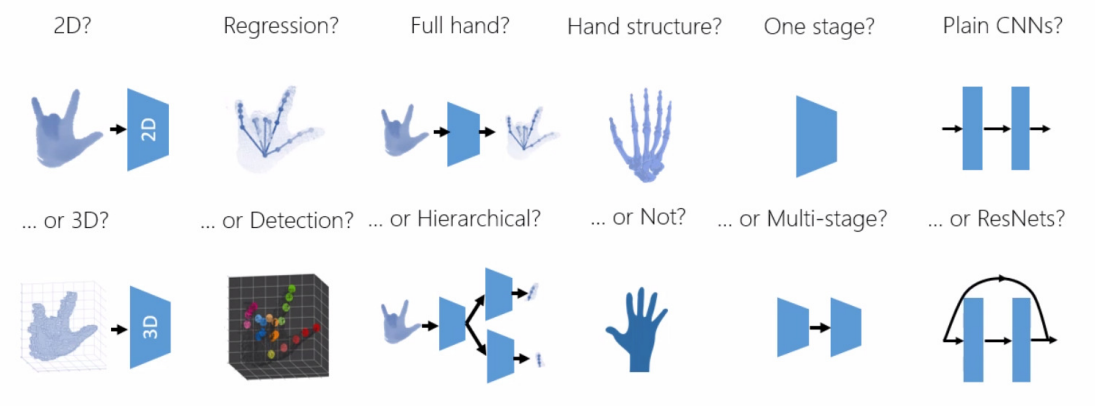
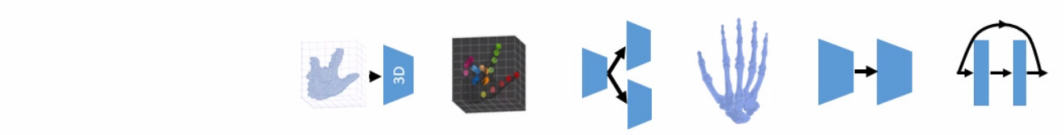


Figure 3.2: ***Method taxonomy.*** *We selection these design choices to group the evaluated methods.*



1	V2V-PoseNet	✓	✓				✓
2	RCN-3D		✓			✓	✓
3	oasis				✓		
4	THU_VCLab			✓		✓	✓
5	NAIST_RV	✓		✓			
6	Vanora						
7	strawberryfg				✓		✓
8	rvhand			✓	✓		
9	mmadadi						
10	LSL				✓		

Figure 3.3: ***Method taxonomy.*** *Our evaluated methods are grouped into different categories.*

joint, the network has a encoder and decoder. *NAIST_RV* [Yang et al., 2017] proposes a 3D CNN with a hierarchical branch structure, where the input is a 50^3-voxel grid.

Detection-based *vs*Regression-based. Detection-based methods [Molchanov et al., 2017, Moon et al.,] produce a probability density map for each joint. The method of *RCN-3D* [Molchanov et al., 2017] is an RCN+ network [Honari et al., a], based on Recombinator Networks (RCN) [Honari et al., b] with 17 layers and 64 output feature maps for all layers

except the last one, which outputs a probability density map for each of the 21 joints. *V2V-PoseNet* [Moon et al.,] uses a 3D CNN to estimate per-voxel likelihood of each joint, and a CNN to estimate the center of mass from the cropped depth map. For training, 3D likelihood volumes are generated by placing normal distributions at the locations of hand joints. Regression-based methods [Akiyama et al., 2017, Chen et al., 2017a, Ge et al., 2018, Ge and Yuan, 2017, Li and Lee, 2017, Madadi et al., 2017, Oberweger et al., 2015a, Yang et al., 2017] directly map the depth image to the joint locations or the joint angles of a hand model [Sinha et al., , Zhou et al.,]. *rvhand* [Akiyama et al., 2017] combines ResNet [He et al., 2016], Region Ensemble Network (*REN*) [Guo et al., 2017], and *DeepPrior* [Oberweger et al., 2015a] to directly estimate the joint locations. *LSL* [Li and Lee, 2017] uses one network to estimate a global scale factor and a second network [Zhou et al.,] to estimate all joint angles, which are fed into a forward kinematic layer to estimate the hand joints.

Hierarchical models divide the pose estimation problem into sub-tasks [Akiyama et al., 2017, Chen et al., 2017a, Guo et al., 2017, Madadi et al., 2017, Yang et al., 2017]. The evaluated methods divide the hand joints either by finger [Madadi et al., 2017, Yang et al., 2017], or by joint type [Akiyama et al., 2017, Chen et al., 2017a, Guo et al., 2017]. *mmadadi* [Madadi et al., 2017] designs a hierarchically structured CNN, dividing the convolution+ReLU+pooling blocks into six branches (one per finger with palm and one for palm orientation), each of which is then followed by a fully connected layer. The final layers of all branches are concatenated into one layer to predict all joints. *NAIST_RV* [Yang et al., 2017] chooses a similar hierarchical structure of a 3D CNN, but uses five branches, each to predict one finger and the palm. *THU_VCLab* [Chen et al., 2017a], *rvhand* [Akiyama et al., 2017], and *REN* [Guo et al., 2017] apply constraints per finger and joint-type (across fingers) in their multiple regions extraction step, each region containing a subset of joints. All regions are concatenated in the last fully connected layers to estimate the hand pose.

Structured methods embed physical hand motion constraints into the model [Ge et al., 2018, Li and Lee, 2017, Madadi et al., 2017, Oberweger et al., 2015a, Wan, 2017, Zhou et al.,]. Structural constraints are included in the CNN model [Li and Lee, 2017, Oberweger and Lepetit, , Oberweger et al., 2015a] or in the loss function [Madadi et al., 2017, Wan, 2017].

DeepPrior [Oberweger et al., 2015a] learns a prior model and integrates it into the network by introducing a *bottleneck* in the last CNN layer. *LSL* [Li and Lee, 2017] uses prior knowledge in *DeepModel* [Zhou et al.,] by embedding a kinematic model layer into the CNN and using a fixed hand model. *mmadadi* [Madadi et al., 2017] includes the structure constraints in the loss function, which incorporates physical constraints about natural hand motion and deformation. *strawberryfg* [Wan, 2017] applies a structure-aware regression approach, Compositional Pose Regression [Sun et al., 2017], and replaces the original ResNet-50 with ResNet-152. It uses phalanges instead of joints for representing pose, and defines a loss function that encodes long-range interaction between the phalanges.

Multi-stage methods propagate results from each stage to enhance the training of the subsequent stages [Chen et al., 2017a, Molchanov et al., 2017]. *THU_VCLab* [Chen et al., 2017a] uses *REN* [Guo et al., 2017] to predict an initial hand pose. In the following stages, feature maps are computed with the guidance of the hand pose estimate in the previous stage. *RCN-3D* [Molchanov et al., 2017] has five stages: (1) 2D landmark estimation using an RCN+ network [Honari et al., a], (2) estimation of corresponding depth values by multiplying probability density maps with the input depth image, (3) inverse perspective projection of the depth map to 3D, (4) error compensation for occlusions and depth errors (a 3-layer network of residual blocks) and, (5) error compensation for noise (another 3-layer network of residual blocks).

Residual networks. ResNet [He et al., 2016] is adopted by several methods [Akiyama et al., 2017, Chen et al., 2017a, Guo et al., 2017, Molchanov et al., 2017, Moon et al., , Sun et al., 2017]. *V2V-PoseNet* [Moon et al.,] uses residual blocks as main building blocks. *strawberryfg* [Wan, 2017] implements the Compositional Pose Regression method [Sun et al., 2017] by using ResNet-152 as basic network. *RCN-3D* [Molchanov et al., 2017] uses two small residual blocks in its fourth and fifth stage.

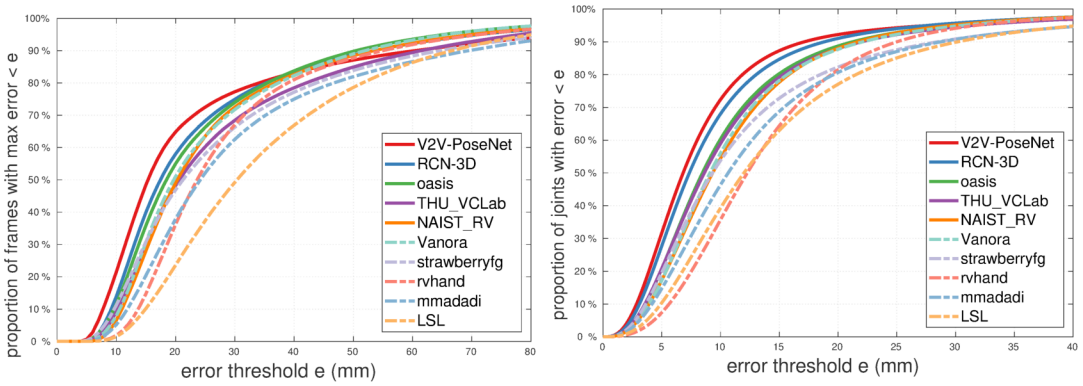


Figure 3.4: ***Estimation errors.*** *(Left) Proportion of frames within maximum error threshold [Taylor et al., 2012]. (Right) Proportion of joints within an error threshold [Sharp et al., 2015].*

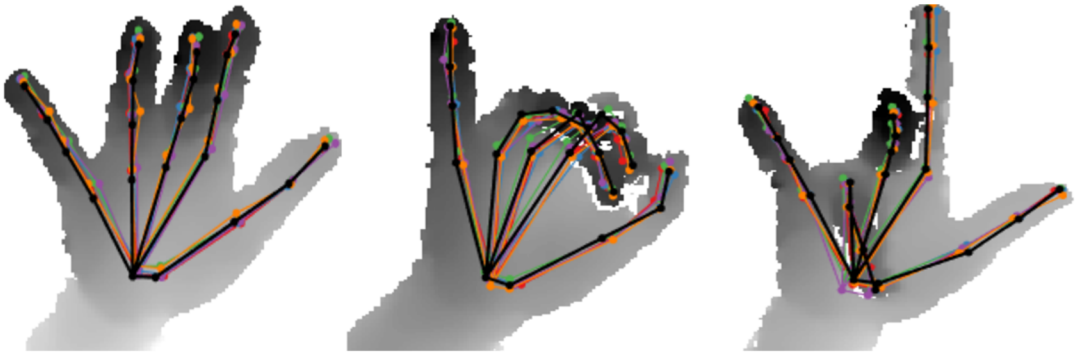


Figure 3.5: ***Estimating annotation errors.*** *Three examples of Easy Poses overlayed with estimates by the top five methods (shown in different colors). Poses vary slightly, but are close to the ground truth (black).*

3.5 Results

The aim of this evaluation is to identify success cases and failure modes. We use both standard error metrics [Oikonomidis et al., 2011a, Sharp et al., 2015, Taylor et al., 2012] and new proposed metrics to provide further insights. We consider joint visibility, seen *vs*unseen subjects, hand view point distribution, articulation distribution, and per-joint accuracy.

Case / Method	Seen Visible	Seen Occ	Unseen Visible	Unseen Occ
V2V-PoseNet	**6.2**	**8.0**	11.1	**14.6**
RCN-3D	6.9	9.0	**10.6**	14.8
oasis	8.2	9.8	12.4	14.9
THU_VCLab	8.4	10.2	12.5	16.1
NAIST_RV	8.8	10.1	13.1	15.6
Vanora	8.8	10.5	12.9	15.5
strawberryfg	9.3	10.7	16.4	18.8
rvhand	12.2	11.9	16.1	17.6
mmadadi	10.6	13.6	15.6	19.7
LSL	11.8	13.1	18.1	19.2
Top5	7.7	9.4	11.9	15.2
All	9.1	10.7	13.9	16.7

Table 3.4: ***Mean errors (in mm) for single frame pose estimation, divided by cases.*** *'Seen' and 'Unseen' refers to whether or not the hand shape was in the training set, and 'Occ' denotes 'Occluded joints'.*

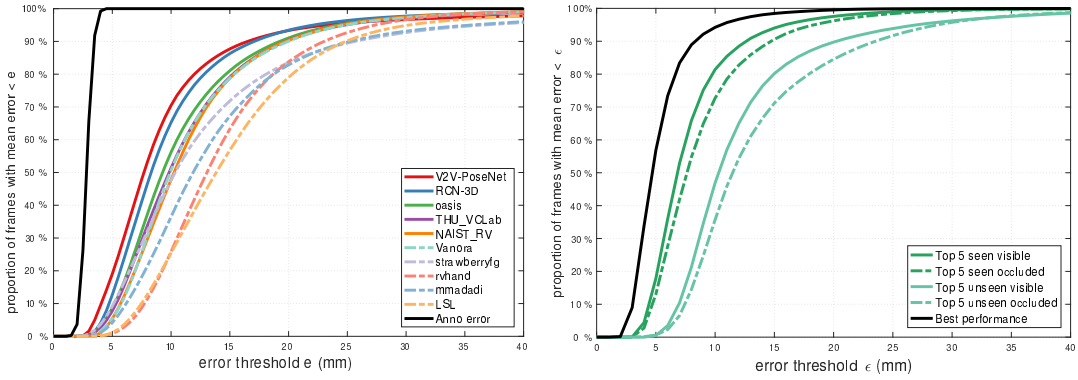


Figure 3.6: ***Success rates for different methods.*** *(Left) all evaluated methods using all test data. (Right) the average of the top five methods for four cases.*

3.5.1 Single frame pose estimation

Over the 6-week period of the challenge the lowest mean error could be reduced from 19.7 millimeter to 10.0 millimeter by exploring new model types and improving data augmentation,

optimization and initialization. For hand shapes seen during training, the mean error was reduced from 14.6 millimeter to 7.0 millimeter, and for unseen hand shapes from 24.0 millimeter to 12.2 millimeter. Considering typical finger widths of 10-20 millimeter, these methods are becoming applicable to scenarios like pointing or motion capture, but may still lack sufficient accuracy for fine manipulation that is critical in some UI interactions.

We evaluate ten state-of-the-art methods (Table 3.2) directly and three methods indirectly, which were used as components of others, *DeepPrior* [Oberweger et al., 2015a], *REN* [Guo et al., 2017], and *DeepModel* [Zhou et al.,]. Figure 3.4 shows the results in terms of two metrics: (left) the proportion of frames in which all joint errors are below a threshold [Taylor et al., 2012] and (right) the total proportion of joints below an error threshold [Sharp et al., 2015].

Figure 3.6 (left) shows the success rates based on per-frame average joint errors [Oikonomidis et al., 2011a] for a varying threshold. The top performer, *V2V-PoseNet*, estimates 70% of frames with a mean error of less than 10 millimeter, and 20% of frames with mean errors under 5 millimeter. All evaluated methods achieve a success rate greater than 80% with an average error of less than 20 mm.

As has been noted by [Guo et al., 2017, Oberweger and Lepetit,], data augmentation is beneficial, especially for small datasets. The top performers all chose to use data augmentation, see Table 3.2, the common practice is to do (1) rotation, e.g., in-plane rotation, (2) scaling the hand size in 3D by multiplying the original size with a scale factor, e.g., between 0.85 and 1.15, (3) translation by adding random 3D offset to hand location. However, note that even though the top performing methods employ data augmentation, it is still difficult to generalize to hands from unseen subjects, see Table 3.4, with an error gap of around 6 millimeter between seen and unseen subjects. Some methods generalize better than others, in particular *RCN-3D* is the top performer on unseen subjects, even though it is not the best on seen subjects.



Figure 3.7: ***Easy pose examples.*** *From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rv-hand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*

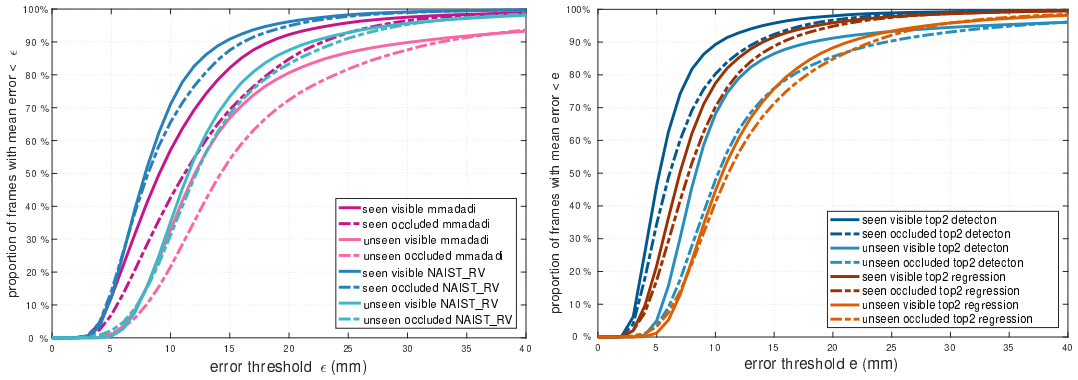


Figure 3.8: *(Left) Direct comparison with 2D CNN and 3D CNN, mmadadi is a 2D CNN, NAIST_RV has the same structure but replaced 2D CNN with 3D CNN. (Right) The average of top-two detection-based and regression-based methods in four cases.*

3.5.1.1 Annotation error

The annotation error takes into account inaccuracies due to small differences of 6D sensor placement for different subjects during the annotation, and uncertainty in the wrist joint location. To quantify this error we selected poses for which all methods achieved a maximum error [Taylor et al., 2012] of less than 10 millimeter. We denote these as *Easy Poses*. The pose estimation task for these can be considered solved, as shown in Figure 3.5. The outputs of the top five methods are visually accurate and close to the ground truth. We estimate the *Annotation Error* as the error on these poses, which has a mean value of 2.8 millimeter and a standard deviation of 0.5 millimeter. Figure 3.7 shows qualitative results on *easy pose* examples.

3.5.1.2 Analysis by occlusion and unknown subject

Average error for four cases: To analyze the results with respect to joint visibility and hand shape, we partition the joints into four groups, based on whether or not they are visible, and whether or not the subject was seen at training time: visible joints of seen hands (*i.e*subjects seen during training), occluded joints of seen hands, visible joints of unseen hands, and occluded joints of unseen hands. Different hand shapes and joint occlusions are responsible for



Figure 3.9: ***Generalization to new hand shapes.*** *Each column represents one unseen hand shape. From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rvhand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*

a large proportion of errors, see Table 3.4. The error for unseen subjects is significantly larger than for seen subjects. Moreover, the error for visible joints is smaller than for occluded joints. Based on the first group (visible, seen), we carry out a best-case performance estimate for the current state-of-the-art. For each frame of seen subjects, we first choose the best result from all methods, and calculate the success rate based on the average error for each frame, see the black curve in Figure 3.6 (Right). Figure 3.9 shows qualitative results on examples with hand shapes not seen during training.

2D *vs*3D CNNs: We compare two hierarchical methods with similar structure but different representation. Figure 3.8 (left) shows *mmadadi* [Madadi et al., 2017], which employs a 2D CNN, and *NAIST_RV* [Yang et al., 2017], using a 3D CNN. *mmadadi* and *NAIST_RV* have almost the same structure, but *NAIST_RV* [Yang et al., 2017] uses a 3D CNN, while *mmadadi* [Madadi et al., 2017] uses a 2D CNN. *NAIST_RV* [Yang et al., 2017] outperforms *mmadadi* [Madadi et al., 2017] in all four cases. *mmadadi* [Madadi et al., 2017] exploited a hierarchical tree-like structured CNN, the Convolution+Relu+Pooling blocks were hierarchically branched into six branches, each of which was then followed with fully connect layers, the last layers of each branch are concatenated into one layer to predict all joints. One of the six blocks was to estimate the global orientation of the hand, which was intended to improve the palm joints' accuracy. For the rest five blocks, each was to estimate the palm with one finger to focus on a specific finger. *NAIST_RV* [Yang et al., 2017] choose a similar hierarchical branch structure, but have five branches, each to predict one finger and the palm. The main improvements are two-folds: (1) *NAIST_RV* [Yang et al., 2017] used 3D CNN rather than 2D CNN, (2) *NAIST_RV* [Yang et al., 2017] adopted a hand detection which is developed from the U-net [Ronneberger et al., 2015].

Detection-based *vs*regression-based methods: We compare the average of the top two detection-based methods with the average of the top two regression-based methods. In all four cases, detection-based methods outperform regression-based ones, see Figure 3.8 (Right). In the challenge, the top two methods are detection-based methods, see Table 3.2. Note that a similar trend can be seen in the field of full human pose estimation, where only one method in a recent challenge was regression-based [MPII Leader Board,].

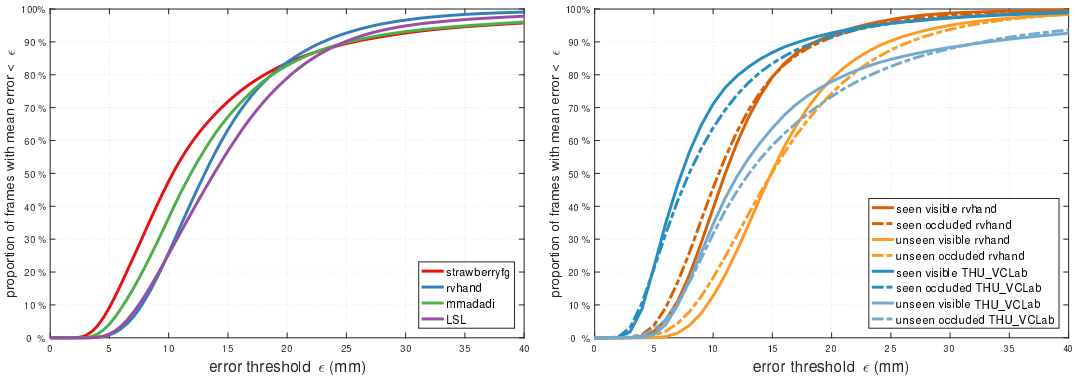


Figure 3.10: *(Left) Comparison among structured methods. (Right) Comparison of a cascaded multi-stage method (THU_VCLab) and a one-off method (rvhand). Both of them use REN [Guo et al., 2017] as backbone.*

Hierarchical methods: Hierarchical constraints can help in the case of occlusion. The hierarchical model in *rvhand* [Akiyama et al., 2017] shows similar performance on visible and occluded joints. *rvhand* [Akiyama et al., 2017] has better performance on occluded joints when the error threshold is smaller than 15 millimeter, see Figure 3.10 (right). The underlying *REN* module [Guo et al., 2017], which includes finger and joint-type constraints seems to be critical. Methods using only per-finger constraints, *e.g*, *mmadadi* [Madadi et al., 2017] and *NAIST_RV* [Yang et al., 2017], generalize less well to occluded joints, see Figure 3.8.

Structural methods: We compare four structured methods *LSL* [Li and Lee, 2017], *mmadadi* [Madadi et al., 2017], *rvhand* [Akiyama et al., 2017], and *strawberryfg* [Wan, 2017], see Figure 3.10 (left). *strawberryfg* [Wan, 2017] and *mmadadi* [Madadi et al., 2017] have higher success rates when the error threshold is below 15 millimeter, while *LSL* [Li and Lee, 2017] and *rvhand* [Akiyama et al., 2017] perform better for thresholds larger than 25 millimeter. Embedding structural constraints in the loss function has been more successful than including them within the CNN layers. *strawberryfg* [Wan, 2017] performs the best, using constraints on phalanges rather than on joints.

Single- *vs*multi-stage methods: Cascaded methods work better than single-stage methods, see Figure 3.10 (right). Compared to other methods, *rvhand* [Akiyama et al., 2017] and

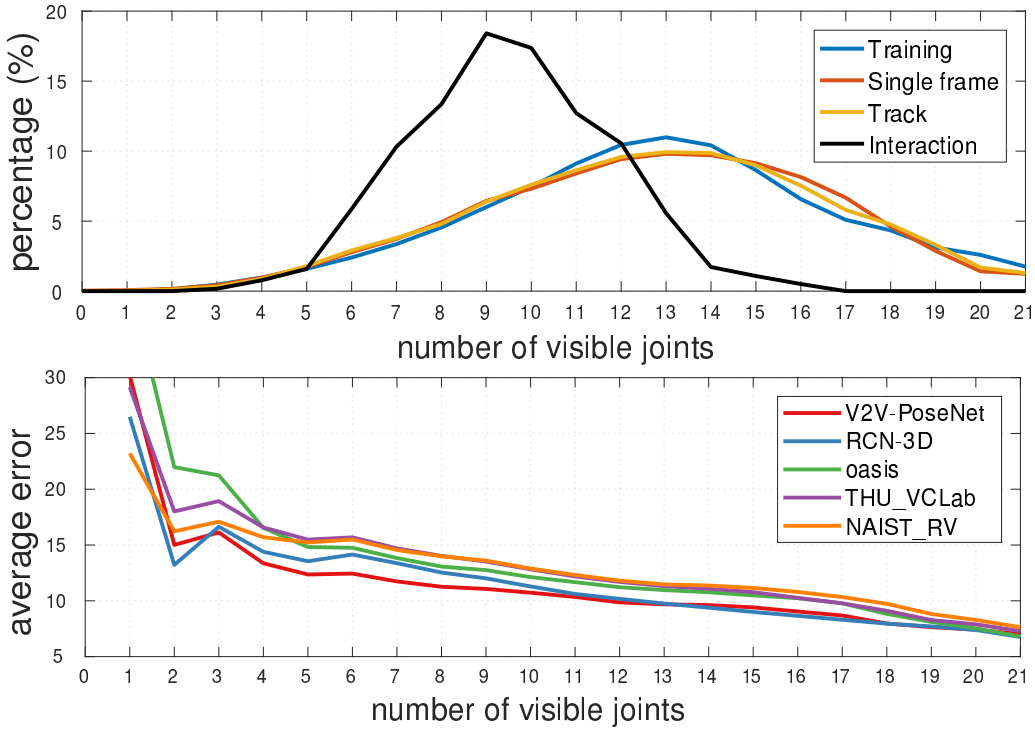


Figure 3.11: ***Joint visibility. Top:*** *Joint visibility distributions for training set and testing sets.* ***Bottom:*** *Average error (mm) for different numbers of visible joints and different methods.*

THU_VCLab [Chen et al., 2017a] both embed structural constraints, employing *REN* as their basic structure. *THU_VCLab* [Chen et al., 2017a] takes a cascaded approach to iteratively update results from previous stages, outperforming *rvhand* [Akiyama et al., 2017].

3.5.1.3 Analysis by number of occluded joints

Most frames contain joint occlusions, see Figure 3.11 (top). We assume that a visible joint lies within a small range of the 3D point cloud. We therefore detect joint occlusion by thresholding the distance between the joint's depth annotation value and its re-projected depth value. As shown in Figure 3.11 (bottom), the average error decreases nearly monotonously for increasing numbers of visible joints. Figure 3.12 shows qualitative results on examples with significant self-occlusion.



Figure 3.12: ***Occlusions.*** *From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rv-hand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*

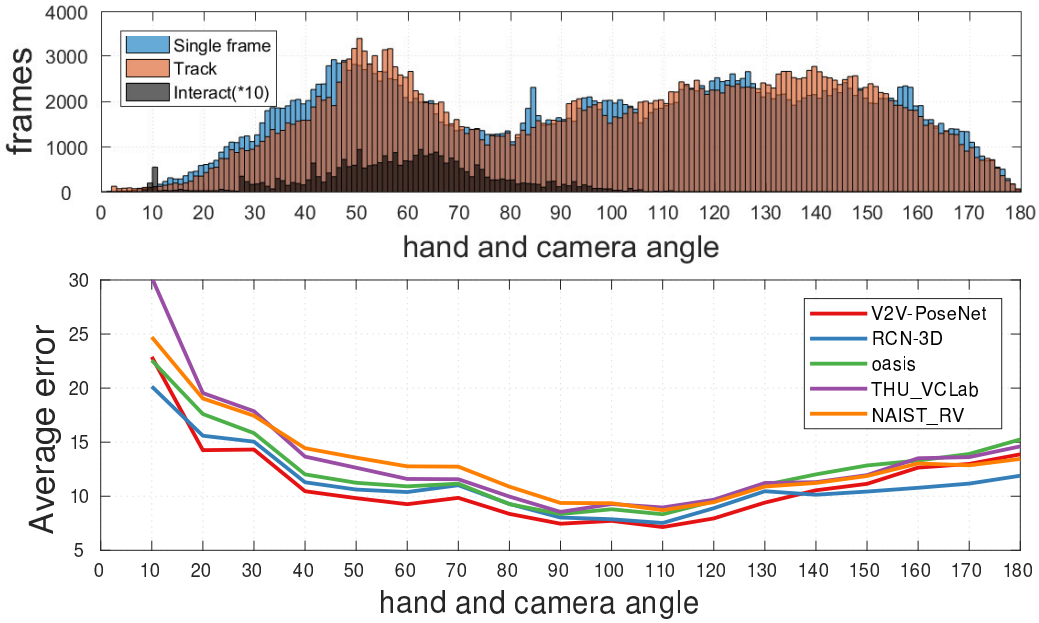


Figure 3.13: ***View point distributions.*** *The error is significantly higher for small angles between hand and camera orientations.*

3.5.1.4 Analysis based on view point

The view point is defined as the angle between the palm and camera directions. The test data covers a wide range of view points for the *Single frame pose estimation* task, see Figure 3.13 (top). View points in the [70, 120] range have a low mean error of below 10 millimeter. View points in the [0, 10] range have a significantly larger error due to the amount of self occlusion. View points in the [10, 30] range have an average error of 15-20 millimeter. View point ranges of [30,70] and [120, 180] show errors of 10-15 millimeter. Third-person and egocentric views are typically defined by the hand facing toward or away from the camera, respectively. However, as shown in Figure 3.13, there is no clear separation by view point, suggesting a uniform treatment of both cases is sensible. Note that *RCN-3D* [Molchanov et al., 2017] outperforms others with a margin of 2-3 millimeter on extreme view points in the range of [150,180] degrees due to their depth prediction stage. Figure 3.15 and Figure 3.16 show qualitative results on examples of extreme viewpoints.

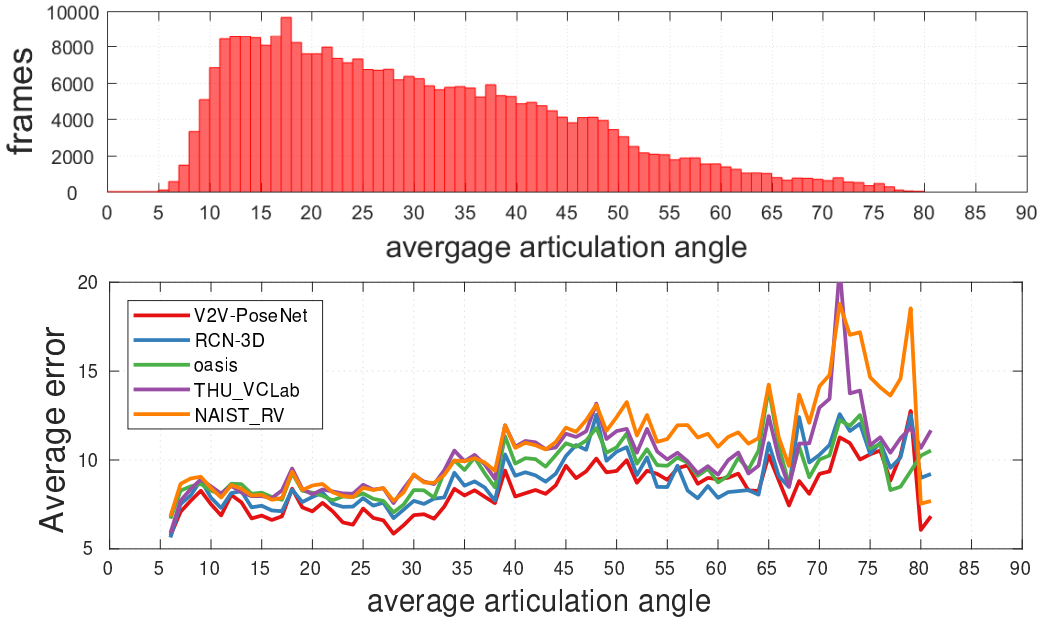


Figure 3.14: ***Articulation distribution.*** *Errors increase for larger finger articulation angles.*

3.5.1.5 Analysis based on articulation

We evaluate the effect of hand articulation on estimation accuracy, measured as the average of 15 finger flexion angles, see Figure 3.14. To reduce the influence from other factors such as view point, we select frames with view point angles within the range of [70, 120]. We evaluate the top five performers, see Figure 3.14 (bottom). For articulation angles smaller than 30 degrees, the mean error is 7 millimeter, when the average articulation angle increases to the range of [35, 70], errors increase to 9-10 mm. When the articulation angle is larger than 70 degrees, close to a fist pose, the mean error increases to over 12 millimeter. Figure 3.17 shows qualitative results on examples with large articulation angles.

3.5.1.6 Analysis by joint type

As before we group joints according to their visibility and the presence of the subject in the training set. We report the top five performers, see Figure 3.18. For the easiest case (visible

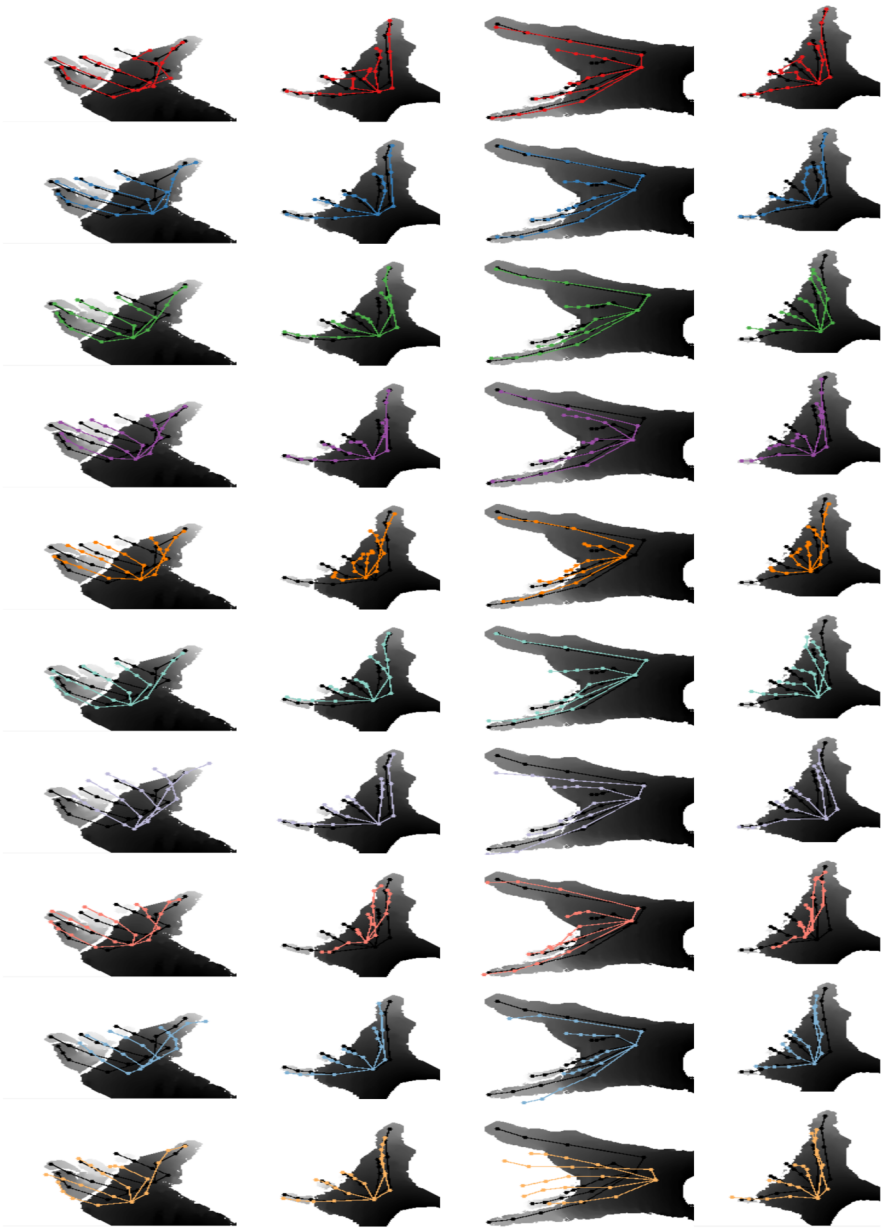


Figure 3.15: ***Viewpoints in range [0, 10] degrees.*** *From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rvhand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*



Figure 3.16: ***Viewpoints in range [170, 180] degrees.*** *From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rvhand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*



Figure 3.17: ***Articulation angle larger than 70 degrees.*** *From top to bottom, rows show results of V2V-PoseNet [Moon et al.,], RCN-3D [Molchanov et al., 2017], oasis [Ge et al., 2018], THU_VCLab [Chen et al., 2017a], NAIST_RV [Yang et al., 2017], Vanora [Ge and Yuan, 2017], strawberryfg [Wan, 2017], rvhand [Akiyama et al., 2017], mmadadi [Madadi et al., 2017], and LSL [Li and Lee, 2017], respectively. The ground truth annotation is shown in black.*

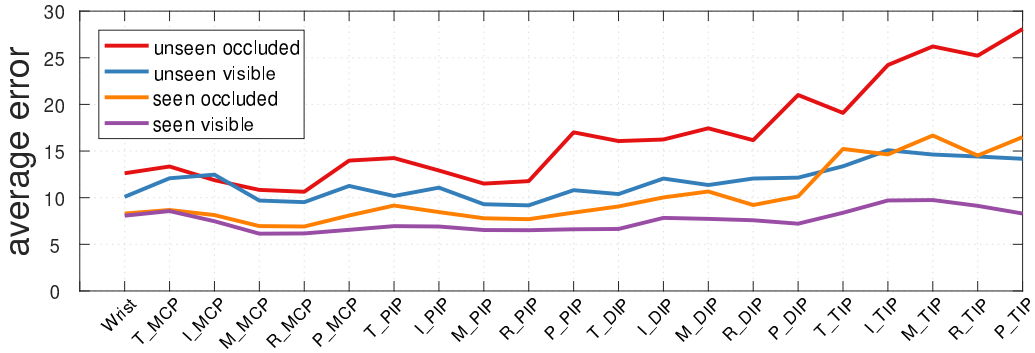


Figure 3.18: ***Average error of the top five methods for each joint in the Single frame pose estimation task.*** *Finger tips have larger errors than other joints. For non-tip joints, joints on ring finger and middle finger have lower average errors than other fingers. 'T', 'I', 'M', 'R', 'P' denotes 'Thumb', 'Index', 'Middle', 'Ring', and 'Pinky' finger, respectively.*

joints of seen subjects), all 21 joints have a similar average error of 6-10 millimeter.

For seen subjects, along the kinematic hand structure from the wrist to finger tips, occluded joints have increasingly larger errors, reaching 14 millimeter in the finger tips. Visible joints of unseen subjects have larger errors (10-13 millimeter) than that of seen subjects. Occluded joints of unseen subjects have the largest errors, with a relatively smaller error for the palm, and larger errors for finger tips (24-27 millimeter). We draw two conclusions: (1) all the top performers have difficulty in generalizing to hands from unseen subjects, (2) occlusions have more effect on finger tips than other joints. An interesting observation is that middle and ring fingers tend to have smaller errors in *MCP* and *PIP* joints than other fingers. One reason may be that the motion of these fingers is more restricted. The thumb's *MCP* joint has a larger error than for other fingers, because it tends to have more discrepancy among different subjects.

3.5.2 Hand pose tracking

In this task we evaluate three state-of-the-art methods, see Table 3.5, Figure 3.19 and Figure 3.20. Discriminative methods [Chen et al., 2017a,Molchanov et al., 2017,Yang et al., 2017] break tracking into two sub-tasks: detection and hand pose estimation, sometimes merging the sub-tasks [Chen et al., 2017a]. Based on the detection methods, 3D hand pose estimation can

Method	Model	AVG
RCN-3D_track [Molchanov et al., 2017]	scanning window + post-processing + pose estimator [Molchanov et al., 2017]	10.5
NAIST_RV_track [Yang et al., 2017]	hand detector [Ronneberger et al., 2015] + hand verifier + pose estimator [Yang et al., 2017]	12.6
THU_VCLab_track [Chen et al., 2017a]	Estimation [Chen et al., 2017a] with the aid of tracking + re-initialization [Ren et al.,]	13.7

Table 3.5: ***Methods evaluated on 3D hand pose tracking.*** *The last column is the average error in mm for all frames.*

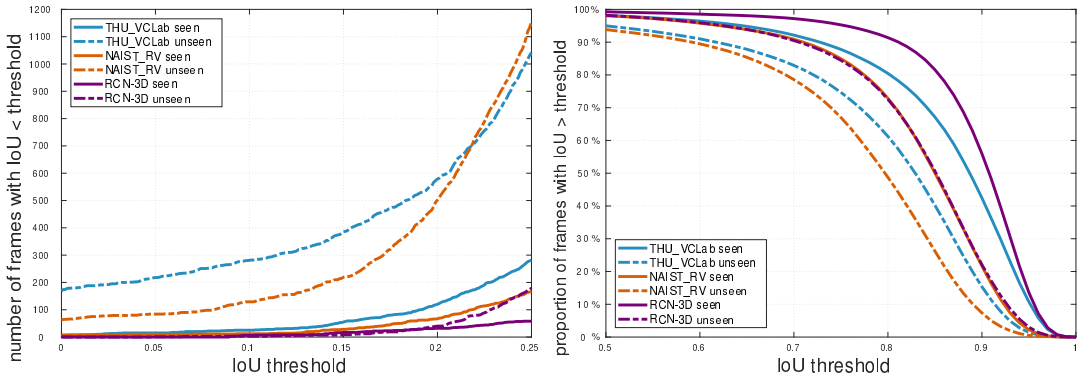


Figure 3.19: ***Error curves for hand tracking.*** *(Left) number of frames with IoU below threshold. (Right) success rate for the detection part of each method in two cases.*

be grouped into pure tracking [Molchanov et al., 2017], tracking-by-detection [Yang et al., 2017], and a combination of tracking and re-initialization [Chen et al., 2017a], see Table 3.5.

Pure tracking: *RCN-3D_track* estimate the bounding box location by scanning windows based on the result in the previous frame, including a motion estimate. Hand pose within the bounding box is estimated using *RCN-3D* [Molchanov et al., 2017]. The bounding box was extracted in two steps: (1) bounding box for the next frame was estimated as a tight box around estimated keypoints plus 25% from each side with constraints (minimum is 100px, maximum is 400px); 2) For post-processing, they took median of 9 consecutive frames to remove outliers. *RCN-3D_track* do tracking with the help of scanning windows, when the bounding box is chosen, it estimated the hand pose by using the *RCN-3D* [Molchanov et al., 2017] model. The bounding box was extracted in two steps: (1) bounding box for the next frame was estimated as

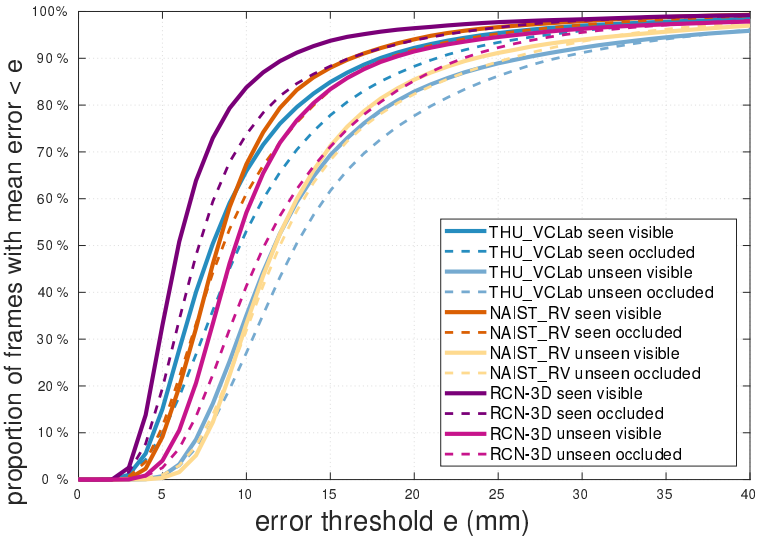


Figure 3.20: ***Error curves for hand tracking.*** *Success rate for three methods in four cases.*

a tight box around estimated key points plus 25% from each side with constraints (minimum is 100px, maximum is 400px); 2) For post-processing, they took median of 9 consecutive frames to remove outliers.

Tracking-by-detection: *NAIST_RV_track* is a tracking-by-detection method with three components: hand detector, hand verifier, and pose estimator. The hand detector is built on U-net [Ronneberger et al., 2015] to predict a binary hand-mask, which, after verification, is passed to the pose estimator *NAIST_RV* [Yang et al., 2017]. If verification fails, the result from the previous frame is chosen. It will loss if the hand moves very fast, and may have multiple detections where multiple hands appeared. One potential way to decrease the error could be design a better hand detector.

Hybrid tracking and detection: *THU_VCLab_track* [Chen et al., 2017a] makes use of the previous tracking result and the current frame's scanning window. The hand pose of the previous frame is used as a guide to predict the hand pose in the current frame. Specifically, the hand pose P^t for frame t is obtained by $P^t = \alpha R(P^{t-1}, D) + (1-\alpha)R(P^{init}, D)$, where D is depth image, R is *Pose-REN* [Chen et al., 2017a], P^{t-1} is previous frame's hand pose, P^{init} is

the result of *REN* [Guo et al., 2017]. The previous frame's bounding box is used for the current frame. During fast hand motion, Faster R-CNN [Ren et al.,] is used for re-initialization.

Detection accuracy: Based on how they do detection, the evaluated methods are separated into three categories: (1) pure scanning window (*RCN-3D_track* [Molchanov et al., 2017]), (2) pure detection (*NAIST_RV_track* [Yang et al., 2017]), (3) combining tracking, scanning window, and re-initialization (*THU_VCLab_track* [Chen et al., 2017a]). We first evaluate the detection accuracy by evaluating the bounding box overlap, *i.e*, the intersection over union (IoU) of the detection and ground truth bounding boxes, see Figure 3.19 (right). Overall, *RCN-3D_track* is more accurate than *THU_VCLab_track*, which itself outperforms *NAIST_RV_track*. Pure detection methods have a larger number of false negatives, especially when multiple hands appear in the scene, see Figure 3.19. There are 72 and 174 missed detections (IoU of zero), for *NAIST_RV_track* and *THU_VCLab_track*, respectively. By tracking and re-initializing, *THU_VCLab_track* achieves better detection accuracy overall. *RCN-3D_track*, using motion estimation and single-frame hand pose estimation, shows the lowest error.

Tracking accuracy is shown in Figure 3.20. Even through *THU_VCLab* performs better than *NAIST_RV* in the *Single frame pose estimation* task, *NAIST_RV_track* performs better on the tracking tasks due to per-frame hand detection. Figure 3.21 shows mores results using two other evaluation metrics, (left) The proportion of frames with the maximum error below a threshold [Taylor et al., 2012], *i.e*it is based on the most poorly estimated joint per frame, ignoring other joints, (right) the proportion of joints below an error threshold [Sharp et al., 2015].

3.5.3 Hand object interaction

For this task, we evaluate four state-of-the-art methods, see Table 3.6, Figure 3.22 and Figure 3.23. Compared to the other two tasks there is significantly more occlusion, see Figure 3.11 (top).

Methods explicitly handling occlusion achieve higher accuracy with errors in the range

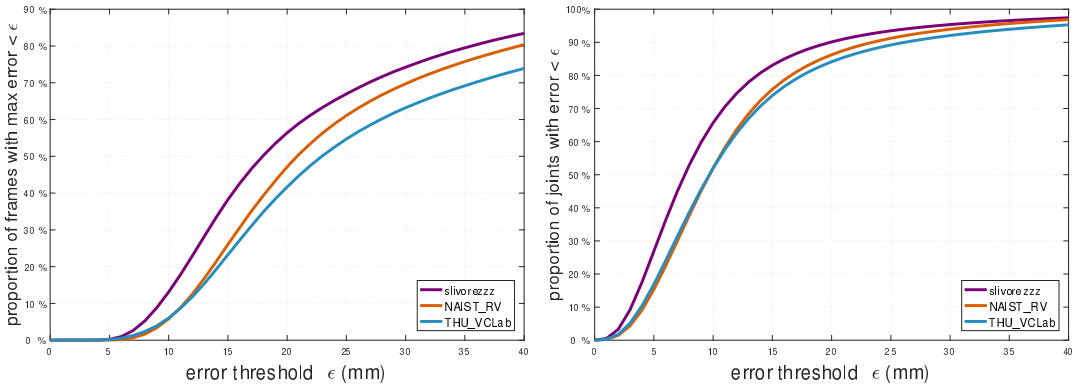


Figure 3.21: ***More results on tracking.*** *Left is for the proportion of frames within maximum error threshold [Taylor et al., 2012], right plot is the results for the proportion of joints within error threshold [Sharp et al., 2015]*

.

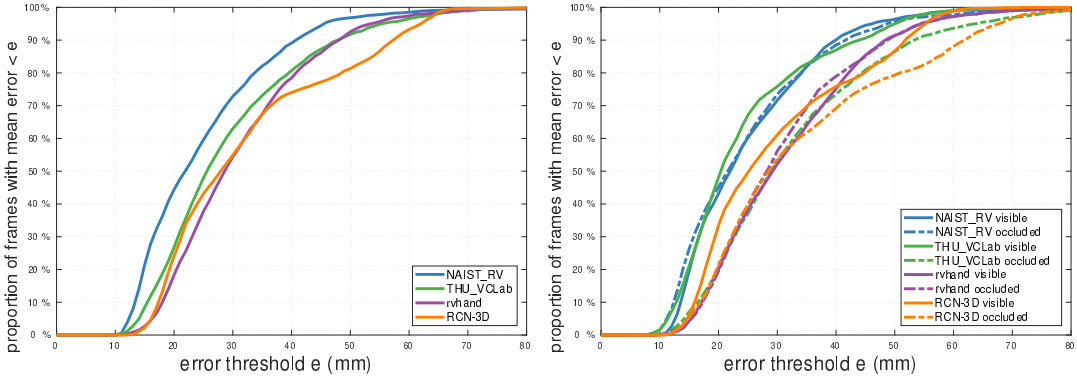


Figure 3.22: ***Error curves for hand-object interaction.*** *(Left): success rate for each method using average error per-frame. (Right) success rate for visible and occluded joints.*

of 25-29 millimeter: (1) *NAIST_RV_obj* [Yang et al., 2017] and rvhand_obj [Akiyama et al., 2017] segment the hand area from the object using a network. (2) *THU_VCLab_obj* [Chen et al., 2017a] removes the object region from cropped hand images with image processing operations [Serra, 1982]. (3) *RCN-3D_obj* [Molchanov et al., 2017] modify their original network to infer the depth values of 2D keypoint locations.

Current state-of-the-art methods have difficulty generalizing to the hand-object interaction scenario. However, *NAIST_RV_obj* [Yang et al., 2017] and rvhand_obj [Akiyama et al., 2017]

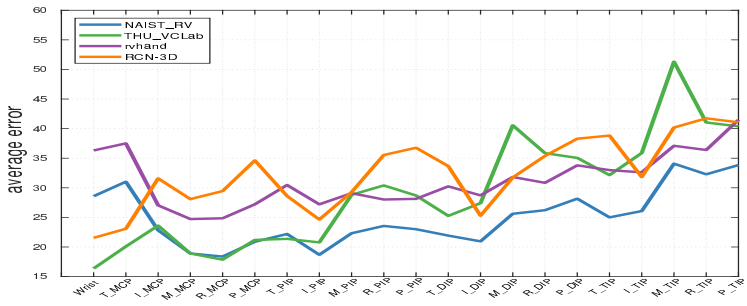


Figure 3.23: ***Error curves for hand-object interaction.*** *Average error for each joint.*

Method	Model	AVG
NAIST_RV_obj [Yang et al., 2017]	Hand-object segmentation CNN + pose estimation [Yang et al., 2017]	25.0
THU_VCLab_obj [Chen et al., 2017a]	Hand-object segmentation (intuitive) + pose estimation [Chen et al., 2017a]	29.2
rvhand_obj [Akiyama et al., 2017]	Hand-object segmentation CNN + pose estimation [Akiyama et al., 2017]	31.3
RCN-3D_obj [Molchanov et al., 2017]	Two RCNs: Feature maps of first are used in the second RCN's stage 2.	32.4

Table 3.6: ***Methods evaluated on hand pose estimation during hand-object interaction.*** *The last column is the average error (mm) for all frames.*

show similar performance for visible joints and occluded joints, indicating that CNN-based segmentation can better preserve structure than image processing operations, see Figure 3.22 (right). *NAIST_RV_obj* [Yang et al., 2017] tried to segment the hand area from the object. They randomly chosen 80 samples to train a hand-object segmentation model (CNN). The segmented hand area is used for pose estimation [Yang et al., 2017]. Large number of training data for hand-object segmentation will improve the results. *THU_VCLab_obj* [Chen et al., 2017a] proposed to remove the object from the cropped hand images with image processing techniques: (a) use *dilation* to separate the object and the hand, (b) apply *closing* operator [Serra, 1982] on the the cropped hand image and only retain the region that has maximum area. It concluded that accurate per-pixel hand segmentation is critical for hand pose estimation with hands interacting with different objects. *RCN-3D_obj* [Molchanov et al., 2017] tried to

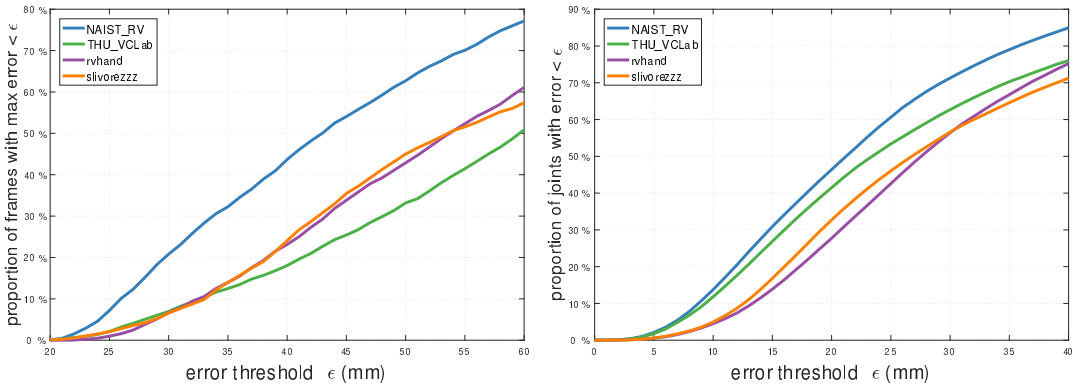


Figure 3.24: ***More results on hand object interaction.*** *Left is for the proportion of frames within maximum error threshold [Taylor et al., 2012], right plot is the results for the proportion of joints within error threshold [Sharp et al., 2015]*

.

by-pass the object that occluded the hand. Instead of using the depth image in it's second stage to estimate the joints' depth values, it replaced the depth image with another *RCN-3D*'s feature maps. In details, it did the following modifications: (1) bounding box is based on proposed bounding box plus constrains (minimum is 100px and max is 400px); (2) At the second stage of *RCN-3D*, it did not use the input depth image, but instead pre-process it by another RCN network (with 32 feature maps for intermediate layers) that takes depth image as input and provides 21 preprocessed maps as output (depth values are still computed as a product of these maps with probability density maps). (3) It subtracted mean value in (X, Y) from keypoints before stage 4 and add it back after stage 5. Figure 3.24 shows mores results using two other evaluation metrics: (left) the proportion of frames with the maximum error below a threshold [Taylor et al., 2012], *i.e*it is based on the most poorly estimated joint per frame, ignoring other joints, (right) the proportion of joints below an error threshold [Sharp et al., 2015].

3.6 Discussion and conclusions

The analysis of the top 10 among 17 participating methods from the *HIM2017* challenge [Yuan et al.,] provides insights into the current state of 3D hand pose estimation.

(1) 3D volumetric representations used with a 3D CNN seem show higher performance, possibly by better capturing the spatial structure of the input depth data.

(2) Detection-based methods that are tested in the challenge tend to outperform regression-based methods, however, regression-based methods can also achieve good performance using explicit spatial constraints. Making use of richer spatial models, *e.g*, bone structure [Sun et al., 2017], helps further. Regression-based methods perform better in extreme view point cases [Molchanov et al., 2017], where severe occlusion occurs. Similar trend also exists in human pose estimation [Sun et al., 2017]. As shown in human pose estimation [Sun et al., 2017], by making explicit use of structural information, (*e.g*bone structure rather than joints structure), regression-based methods can also achieve good performance.

(3) While joint occlusions pose a challenge for most methods, explicit modeling of structure constraints and spatial relation between joints can relatively narrow the gap between errors on visible and occluded joints [Akiyama et al., 2017].

(4) Discriminative methods still generalize poorly to unseen hand shapes. Data augmentation and scale estimation methods model only global shape changes, but not local variations. Integrating hand models with better generative capability may be a promising direction.

(5) Isolated 3D hand pose estimation achieves low mean errors (10millimeter) in the view point range of [70, 120] degrees. However, errors remain large for extreme view points, *e.g*, view point range of [0,10], where the hand is facing away from the camera. Multi-stage methods [Molchanov et al., 2017] tend to perform better in these cases.

(6) In hand tracking, current discriminative methods divide the problem into two sub-tasks: detection and pose estimation, without using the hand shape provided in the first frame. Hybrid methods may work better by using the provided hand shape.

(7) Current methods perform well on single hand pose estimation when trained on a million-scale dataset, but have difficulty in generalizing to hand-object interaction. Two directions seem promising, (a) designing better hand segmentation methods, and (b) training the model with large datasets containing hand-object interaction.

CHAPTER 4

POSE ESTIMATION FROM RGB WITH DEPTH AS PRIVILEGED INFORMATION

Contents

4.1	Overview	81
4.2	Related Work	85
4.3	Methods	88
4.4	Experiments	91
4.5	Discussion and conclusions	103

4.1 Overview

This chapter proposes a method for hand pose estimation from RGB images that uses both external large-scale depth image datasets and paired depth and RGB images as privileged information at training time. We show that providing depth information during training significantly improves performance of pose estimation from RGB images during testing. We explore

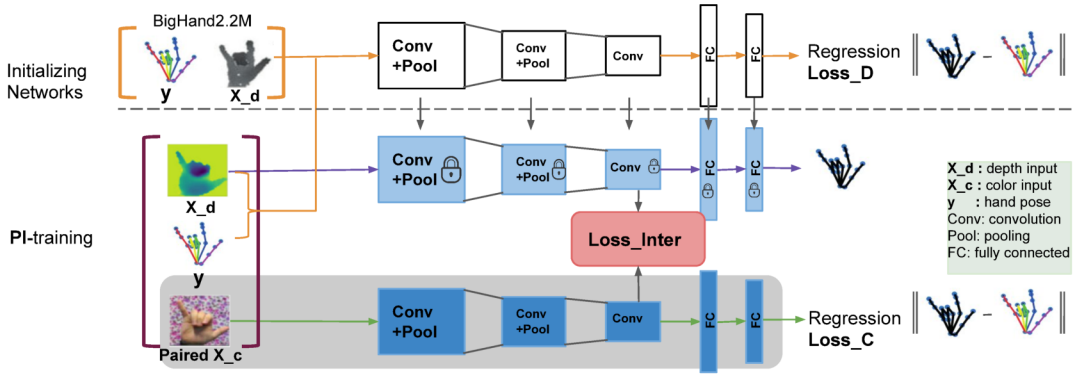


Figure 4.1: ***Proposed framework for 3D hand pose estimation from an RGB image using privileged depth data.*** *Training proceeds in two stages, a pre-training stage and privileged information (PI)-training stage. In the first stage, a depth-based network (top) and an RGB-based (bottom) network are trained independently to minimize 3D pose loss Loss_D and Loss_C. In the second stage, we freeze the parameters of the depth-based network and continue training with paired RGB and depth images, by minimizing a joint loss, which includes Loss_C and a mid-level feature regression loss Loss_Inter.*

different ways of using this privileged information: (1) using depth data to initially train a depth-based network, (2) using the features from the depth-based network of the paired depth images to constrain mid-level RGB network weights, and (3) using the foreground mask, obtained from the depth data, to suppress the responses from the background area. By using paired RGB and depth images, we are able to supervise the RGB-based network to learn middle layer features that mimic that of the corresponding depth-based network, which is well trained on large-scale, accurately annotated depth data. During testing, when only an RGB image is available, our method produces accurate 3D hand pose predictions. Our method is also tested on 2D hand pose estimation. Experiments on three public datasets show that the method outperforms the state-of-the-art methods for hand pose estimation using RGB image input.

3D hand pose estimation has been greatly improved in the past few years, especially with the availability of off-the-shelf depth cameras. While new methods [Oberweger et al., 2015b, Ye et al., 2016, Ge et al., 2017, Wan et al., 2017] and datasets [Tang et al., 2014, Tompson et al., 2014a, Sun et al., 2015, Yuan et al., 2017, Garcia-Hernando et al., 2018] have been published, state-of-the-art methods are still lacking in accuracy required for fine manipulations in gesture interaction for AR or VR systems. There is a large difference in performance between pose estimation from RGB input and depth image input, although several recent works have

attempted to narrow this gap [Simon et al., 2017, Zimmermann and Brox, 2017, Mueller et al., 2017b, Panteleris and Argyros, 2017]. One of the difficulties is the lack of large-scale realistic RGB datasets with accurate annotations. Recent papers have approached this issue by creating synthetic datasets [Zimmermann and Brox, 2017], or employing GANs to generate training data [Mueller et al., 2017a].

In this chapter we propose using depth data as *privileged information* during training. Fully annotated depth datasets [Tang et al., 2014, Tompson et al., 2014a, Sun et al., 2015, Yuan et al., 2017, Garcia-Hernando et al., 2018] are abundant in the literature, but so far no attempt has been made to use this data to support the task of 3D hand pose estimation from RGB images. There are also a few RGB-D datasets proposed recently [Zimmermann and Brox, 2017, Zhang et al., 2016] to tackle the problem of 3D hand pose estimation from RGB images, however all existing methods [Zimmermann and Brox, 2017, Mueller et al., 2017a, Zhang et al., 2016] utilise only RGB images for training. The available depth images, either paired with RGB images [Zhang et al., 2016, Zimmermann and Brox, 2017] or alone in the large scale *BigHand2.2M* dataset [Yuan et al., 2017] could be used to aid the training.

The use of privileged information in training [Vapnik and Vashist, 2009], also called training with hidden information [Wang and Ji, 2015], or side information [Xu et al., 2013], has been shown to improve performance in other domains, such as image classification [Chen et al., 2017b], object detection [Hoffman et al., 2016], and action recognition [Shi and Kim, 2017]. But the concept of using privileged information to help 3D hand pose estimation from RGB images has not been attempted. To the best of our knowledge, this work proposes the first solution. Existing methods related to 3D hand pose estimation and RGB images are pursued in two main directions: (1) using only RGB images for 3D hand pose estimation [Zhang et al., 2016, Zimmermann and Brox, 2017, Mueller et al., 2017a], with different CNN models being proposed. Given the limited size of real RGB datasets, a large number of synthetic images [Zimmermann and Brox, 2017, Mueller et al., 2017a] are created to help the training, whether they are purely synthetic [Zimmermann and Brox, 2017], or using CycleGAN [Zhu et al., 2017] to enforce a certain realism [Mueller et al., 2017a]. (2) Using RGB-D images for 3D hand pose tracking [Mueller et al., 2017b], where the input is the depth channel in addition

to the RGB channels. This works well when the paired RGB and depth images are available at testing. Lack of large-scale annotated training data also limits the success of this approach. Our study proposes a new framework for 3D hand pose estimation from RGB images, by using the existing abundant fully annotated depth data in training, as privileged information. This helps improve 3D hand pose estimation using a single RGB image input at testing.

Our method aims at transferring supervision from depth images to RGB images. Our method has two branches, an RGB-based network and a depth-based network, see Figure 4.1. We explore different ways to use depth data: (1) initially, we treat a large amount of independent external depth training data as privileged information to train the depth-based network. (2) After the initial training is completed, paired RGB and depth images are used to tune the RGB-based network and the depth-based network. The idea is to let the middle layer activations of the RGB network mimic that of the depth network. (3) We also explore the use of foreground hand masks to suppress background area activations in the middle layers of the RGB network. By doing this, we force the RGB network to extract features only from the foreground area.

4.1.1 Contributions

Compared to existing methods for 3D hand pose estimation by RGB images, our main contributions are:

- We introduce the concept of using privileged information (depth images) to help the training of a RGB-based hand pose estimator.
- We propose three ways to use the privileged information: as external training data for a depth-based network, as paired depth data to transfer supervision from the depth-based network to the RGB-based network, as hand masks to suppress the background activations in the RGB-based network.
- Our training strategy can be easily embedded into existing pose estimation methods. We demonstrate this in the experiments of 2D hand pose estimation with a RGB image

input by a different CNN model. Results on 2D hand pose estimation, using our training strategy improve over state-of-the-art methods for 2D hand pose estimation with RGB input.

Comprehensive experiments are conducted on three datasets: the Stereo dataset [Zhang et al., 2016], the RHD dataset [Zimmermann and Brox, 2017], and the Dexter-Object dataset [Sridhar et al., 2016]. The Stereo dataset and RHD dataset are used for evaluating 3D hand pose estimation from an RGB input. All three datasets are used for evaluation of 2D hand pose estimation from a single RGB input. Our method outperformed various state-of-the-art methods on all datasets.

4.2 Related Work

3D hand pose estimation. Hand pose estimation from depth data has made rapid progress in the past years [Oberweger et al., 2015b, Ge et al., 2017, Wan et al., 2017, Sharp et al., 2015, Choi et al., 2015], where comprehensive studies [Erol et al., 2007, Supancic et al., 2015, Yuan et al., 2018] have been instrumental in pushing this field forward. Random forests [Tang et al., 2014, Tang et al., 2015, Wan et al., 2016] and CNNs [Ye et al., 2016, Ge et al., 2017, Wan et al., 2017, Tompson et al., 2014a] trained on large-scale public depth image datasets [Tang et al., 2014, Tompson et al., 2014a, Sun et al., 2015, Yuan et al., 2017, Garcia-Hernando et al., 2018] have shown good performance. A recent benchmark evaluation [Yuan et al., 2018] showed that modern methods achieve mean 3D joint position errors of less than 10mm, making them applicable in certain real world scenarios. Hand pose estimation from RGB images is significantly more challenging [Simon et al., 2017, Zimmermann and Brox, 2017, Mueller et al., 2017b, Panteleris and Argyros, 2017]. Due to the difficulty in capturing real RGB datasets with accurate 3D annotations, recent methods employ synthetic CG data [Zimmermann and Brox, 2017], or *GANerated* images [Mueller et al., 2017a], which are more realistic synthetic images created with a CycleGAN [Zhu et al., 2017]. Mueller *et al* [Mueller et al., 2017a] use an image-to-image translation network to create a large amount of RGB training images and

combine a CNN with a kinematic 3D hand model for pose estimation. The method requires a predefined hand model, adapted for each user. Simon *et al*'s *OpenPose* [Simon et al., 2017] system generates an annotated RGB dataset using a panoptic studio setup, using multiple views to bootstrap 2D hand pose estimation. Zimmermann and Brox [Zimmermann and Brox, 2017] proposed combining hand segmentation and 2D hand pose estimation (using *CPM* [Wei et al., 2016]), followed by estimating 3D hand pose relative to a canonical pose. Panteleris and Argyros [Panteleris and Argyros, 2017] estimate absolute 3D hand pose by first estimating 2D hand pose and then optimizing a 3D hand model with inverse kinematics. Note that there also exists a large body of work on the related task of recovering full 3D human body pose from images. One line of work aims to directly estimate the 3D pose from images [Li and Chan, 2014, Zhou et al., 2016a, Toshev and Szegedy, 2014]. A second approach is to first estimate 2D pose, often in terms of joint locations, and then lift this to 3D pose. 2D key points can be reliably estimated using CNNs and 3D pose is estimated using structured learning or a kinematic model [Tome et al., 2017, Tompson et al., 2014b, Simo-Serra et al., 2012, Zhou et al., 2016b].

Learning with privileged information and transfer learning. Privileged information denotes training data that is available only during training but not at test time. The concept to provide teacher-like supervision at training time was introduced by Vapnik and Vashist [Vapnik and Vashist, 2009]. The idea has proven useful in other domains [Chen et al., 2017b, Hoffman et al., 2016, Shi and Kim, 2017]. Shi *et al* [Shi and Kim, 2017] treated skeleton data as privileged information in CNN-RNNs for action recognition from depth sequences. Chen *et al* [Chen et al., 2017b] manually annotated object masks in 10% of the training data and treated these as privileged information for image classification.

Network transfer through distillation The idea is related to network compression and mimic learning proposed by Ba and Caruana [Ba and Caruana, 2014] as well as network distillation by Hinton *et al* [Hinton et al., 2015], where intermediate layer outputs of one network are approximated by another, possibly smaller, network. These techniques can be used to significantly reduce the number of model parameters without a significant drop in accuracy. In

our case, the application target is similar to transfer learning and domain adaptation. Information from one task, prediction from depth images, is shared with another, prediction from RGB images. In transfer learning and domain adaptation information is shared across different data modalities. Chen *et al* [Chen et al.,] proposed recognition in RGB images by learning from labeled RGB-D data. A common feature representation is learned across two feature modalities. Hoffman *et al* [Hoffman et al., 2016] learned an additional *hallucination* representation, which is informed by the depth data in training. At testing, it used the softmax to select the final prediction between the predictions from the hallucination representation and the predictions from RGB representation. Luo *et al* [Luo et al.,] recently proposed graph distillation for action detection with privileged modalities (RGB, depth, skeleton, and flow), where a novel graph distillation layer was used to dynamically learn to distill knowledge from the most effective modality, depending on the type of action. In our case, we use paired depth and RGB images during training. Depth and RGB networks are first trained separately. Subsequently the RGB network are progressively updated, while the depth network parameters remain fixed.

Learning a latent space representation. Latent space representation also shows promising for 3D hand pose estimation from RGB images [Spurr et al., , Iqbal et al.,]. Spurr *et al* [Spurr et al.,] learned a cross-modal statistical hand model, via learning of a latent space representation that embeds sample points from multiple data sources such as 2D keypoints, images, and 3D hand poses. Multiple encoders were used to project different data modalities into a unified low-dimensional latent space, where a set of decoders reconstruct the hand configuration in either modality. Iqbal *et al* [Iqbal et al.,] used latent 2.5D heatmaps, containing the latent 2D heatmaps and latent depth maps, to ensure the scale and translation invariance. Absolute 3D hand poses are reconstructed from the latent 2.5D heatmaps. Cai *et al* [Cai et al.,] proposed a weakly-supervised method for 3D hand pose estimation from RGB image by introducing an additional depth regularizer module, which rendered a depth image from the estimated 3D hand pose. Training was conducted by minimizing an additional loss term, which is the $L1$ distance between the rendered depth image and the ground truth depth image.

4.3 Methods

We propose a framework to train a hand pose estimation model from RGB images by using depth images as privileged information. The model learns a new RGB representation which is influenced by the paired depth representation through mimicking the mid-level features of a depth network.

As shown in Figure 4.1, we use depth images in two ways: (1) to train an initial depth-based network with the aim of regressing 3D hand poses. Depth data that is annotated with 3D full hand pose information is abundant in the literature, and we choose the largest real dataset BigHand2.2M [Yuan et al., 2017] to train our depth-based model, see the top row of Figure 4.1. (2) Paired RGB and depth images are fed into the RGB-based and depth-based network with the parameters of the depth-based network being frozen. The training of the RGB-based network continues with the aim of minimizing a joint loss function. The joint loss function has two parts, the first part being the 3D hand pose regression loss, *Loss_C*, and the second part the mid-level regression loss, *Loss_Inter*.

4.3.1 Architecture

Figure 4.1 shows our training architecture. There are two base models, each for one input channel. Here we choose deep convolutional neural networks (CNNs) which have been widely used in hand pose estimation and proven useful in transferring information from one network to another [Hinton et al., 2015]. Prior work [Mueller et al., 2017b] has been shown useful in combining RGB and depth images as a four-dimensional RGB-D input to a single CNN model to estimate 3D hand pose. In our architecture, we share information in the middle layers of our two CNN models, one is a depth-based network and the other one is an RGB-based network. Each CNN model takes an input (a depth image or an RGB image) and produces a 3D hand pose estimation result.

For clarity, we denote the depth-based network *Depth_Net*, the RGB-based network

RGB_Net when this is trained before privileged information is used. When privileged information is introduced in the training, we denote the RGB-based network *RGB_PI_Net*. In summary, *RGB_Net* and *RGB_PI_Net* are the same CNN model trained before and after the paired RGB and images are used to train the RGB channel.

We aim at sharing information between the middle layers of our two CNN models, and in particular using *Depth_Net* to inform *RGB_PI_Net* in the training time when paired RGB and depth images are available. To let the *Depth_Net* channel share information with *RGB_PI_Net*, we introduce an intermediate regression loss between the paired layers in the two models. This intermediate regression loss is inspired by prior works [Hoffman et al., 2016, Hinton et al., 2015], where similar techniques are used for model distillation [Hinton et al., 2015], supervision transfer from well labeled RGB images to depth images with limited annotation [Gupta et al.,], and hallucination of different modalities [Hoffman et al., 2016]. We therefore introduce an intermediate loss, which helps *RGB_PI_Net* to extract middle level features that mimic the responses of the corresponding layer of the *Depth_Net* using the paired depth image.

The intermediate loss (or *Loss_Inter* as shown in Figure 4.1) is defined as:

$$Loss_Inter(k) = \|A_k^{Depth} - A_k^{RGB}\|_2^2, \tag{4.1}$$

where A_k^{Depth} and A_k^{RGB} are the k_{th} layer activations for Depth Network and RGB Network, respectively. During testing, where only an RGB image is available, we feed the RGB image into *RGB_PI_Net* to estimate the 3D hand pose.

4.3.2 Training with privileged information

This section explains the details of training the proposed architecture. We choose a base CNN for *Depth_Net* and *RGB_PI_Net* for 3D hand pose estimation. For the base model, we build on Convolutional Pose Machine (CPM)'s [Wei et al., 2016] feature extraction layers with two fully connected layers to regress a 63 dimensional 3D hand pose with 21 joints.

In this initial stage, we call this external depth images as privileged information. Our *Depth_Net* is initially independently trained on BigHand2.2M [Yuan et al., 2017] dataset,

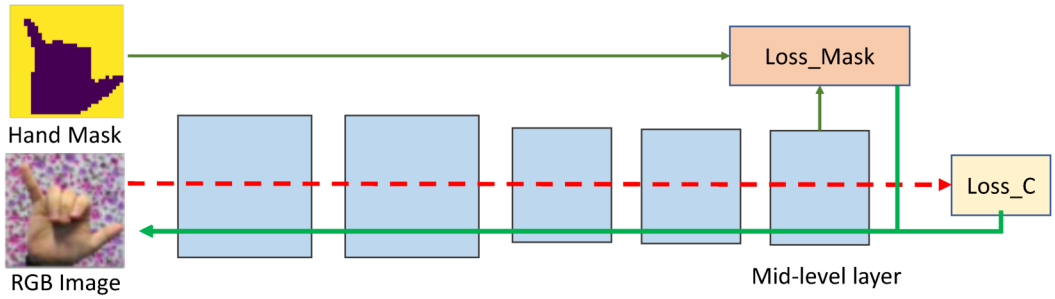


Figure 4.2: ***Treating hand mask as privileged information.*** *Hand mask are used as privileged information to suppress the responses from the background area in the middle layers.*

which has 2.2 million fully annotated (21 joints) depth images. After training, the model is further trained on the depth images of a smaller dataset (*e.g*, Stereo [Zhang et al., 2016] and RHD [Zimmermann and Brox, 2017] datasets) that has fully annotated paired RGB and depth images. The *RGB_Net* is initially trained on the RGB images from the same dataset.

When the initial training is completed for both CNN models, we freeze the parameters of the *Depth_Net* and start the training of *RGB_PI_Net* with privileged information. In this stage, our privileged information is the paired depth images, and comes into use in the form of the middle layer activations of the *Depth_Net*. During the privileged training stage, we want the *RGB_PI_Net*'s middle level layer's activations to match the activations of the corresponding layers of the *Depth_Net*. We have two losses to optimize: (1) *Loss_Inter* (Eqn. 4.1) is used to match the middle layer activations of the two CNN models. (2) *Loss_C* (see Figure 4.1) is the $L2$ loss between the ground truth and the estimated 3D hand pose. Here we use a joint loss:

$$Loss_Joint(k) = Loss_Inter(k) + \lambda \cdot Loss_C, \tag{4.2}$$

where λ is used to balance the two losses, a larger value of λ means less supervision is required from the privileged information, a smaller value means that the model depends more on the supervision. We set λ to 100 for all experiments.

Dataset	No. Training	No. Test	No. Joints	Annotation	Type
Stereo [Zhang et al., 2016]	15,000	3,000	21	2D, 3D	real
RHD [Zimmermann and Brox, 2017]	41,258	2,728	21	2D, 3D	synthetic
Dexter-Object [Sridhar et al., 2016]	-	3,111	5(tips)	2D, 3D	real

Table 4.1: ***Public datasets used in our experiments.***

4.3.3 Foreground mask as privileged information

In addition to the supervision from depth images, we also explore the idea of extracting hand masks from depth images and embedding the hand masks into CNN layers of *RGB_PI_Net* to suppress the background features. As shown in Figure 4.2, we treat the hand mask M_h as privileged information. At test time, when the hand mask is not available, the CNN model is viewed as a standard CNN with convolutional layers, pooling layers and full-connected layers, where the *Loss_Mask* is not used. In the training stage, the foreground hand mask is introduced in the last convolutional layer, as shown in Figure 4.2. Pixels of the mask M_h are zero on the hand region, and one otherwise. We suppress background features by minimizing the regression loss *Loss_Mask*:

$$Loss_Mask = \| A_k^{RGB} \odot M_h \|_2^2 \,, \tag{4.3}$$

where $\odot$ denotes element-wise multiplication.

By minimizing the regression loss, where the response on the hand is multiplied by zero and the response outside the hand is multiplied by one, the response from outside the hand area is suppressed, focusing the response on the hand region.

4.4 Experiments

We carried out experiments on both 3D and 2D hand pose estimation from RGB images. Our experiments are conducted on three public RGB-D datasets: the RHD dataset [Zimmermann and Brox, 2017], the Stereo dataset [Zhang et al., 2016], and the Dexter-Object dataset [Sridhar et al., 2016], as shown in Table 4.1. The RHD dataset is created synthetically and contains 41,238 training and 2,728 test images, with a resolution of 320×320. Each pair of RGB and

depth images contains 3D annotations for 21 hand joints, and intrinsic camera parameters. The RHD dataset is built from 20 different subjects performing 39 actions. The training set has 16 subjects performing 31 actions, while the test set has 4 subjects performing 8 actions. The dataset contains diverse backgrounds sampled from 1,231 Flickr images. The Stereo [Zhang et al., 2016] dataset is a real RGB-D dataset, which has 18,000 pairs of RGB and depth images with a resolution of 640×480 pixels. Each pair is fully annotated with 21 joints. The dataset contains six different backgrounds with respect to different difficulties (texture/textureless, dynamic/static, near/far, hightlight/no-highlight, *etc*). For each background, there are two sequences, each of which has 1,500 pairs of images. The dataset is manually annotated. In our experiments, we follow the same evaluation protocol as [Zimmermann and Brox, 2017], *i.e*, we train on 10 sequences (15,000 images) and test on the remaining 2 sequences (3,000 images). The Dexter-Object [Sridhar et al., 2016] dataset contains 3,111 images of two subjects performing manipulations with a cuboid. The dataset provides RGB and depth images, but only fingertips are annotated. The RGB images have a resolution of 640×320 pixels. Due to the incomplete hand annotation, we used this dataset for cross-dataset generalization.

During testing on a GTX 1080 Ti, the network forwardings take 6ms and 8ms for 3D and 2D hand pose estimation, respectively. The cropping of image patch and the normalization process is the same as in [Zimmermann and Brox, 2017]. To crop the hand region, we use ground truth annotations to obtain an axis aligned crop, resized to a resolution of 256×256 pixels by bilinear interpolation. The learning rate was decreased by half after 10000 iterations.Examples are shown in the first row of Figure 4.11 and Figure 4.12. For 3D hand pose estimation, we use the root joint's world coordinates and the hand's scale to normalize the results.

4.4.1 3D hand pose estimation from RGB

In this section, we investigate the usefulness of depth images to improve the performance of 3D hand pose estimation from an RGB image. Our base CNN model is built upon the feature extraction layers of Convolutional Pose Machine (CPM) [Wei et al., 2016] with two fully connected layers. The final output is a 63 dimensioal vector denoting the 21 joint 3D loca-

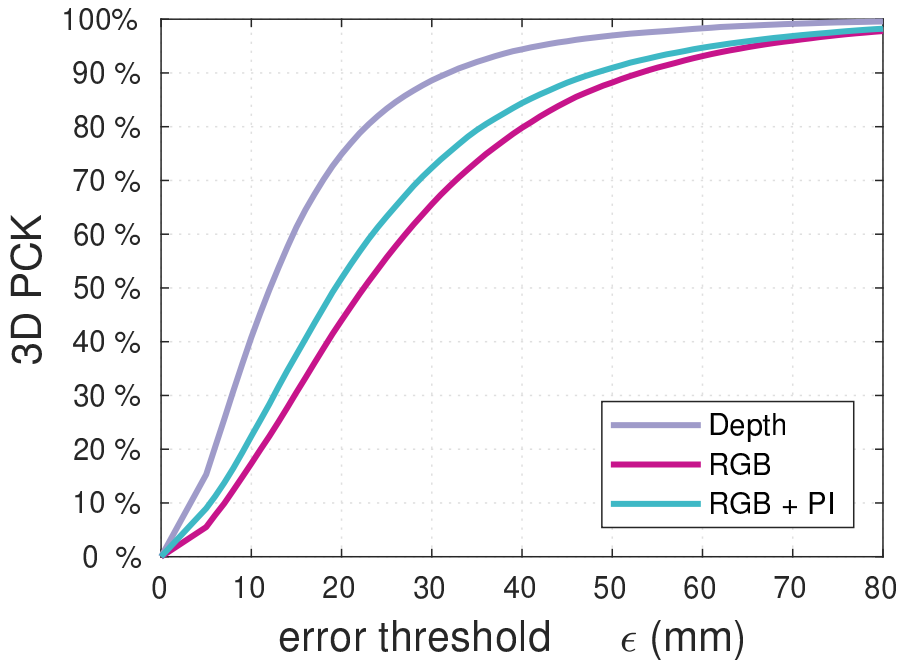


Figure 4.3: ***Results on RHD dataset for 3D hand pose accuracy****.This plot shows the self-comparison on RHD dataset.*

tions. Specifically, our base CNN model contains 14 convolutional layers, 4 pooling layers, and 2 fully-connected layers. At training stage, we have access to paired RGB and depth images. Initially the *Depth_Net* is trained on *BigHand2.2M* [Yuan et al., 2017]. We continue to train the *Depth_Net* using the depth images from the small dataset, *e.g*, Stereo dataset or RHD dataset. We train the *RGB_Net* with the RGB images from the small dataset. When the initial training is completed, we start PI-training with the paired RGB and depth images. We freeze the weights of the *Depth_Net* and add the intermediate regression loss *Loss_Inter* among the mid-level features of *Depth_Net* and *RGB_PI_Net*, then we continue the training of *RGB_PI_Net* by minimizing the joint loss $Loss_Joint$. We applied the intermediate loss to the last convolutional layers of both branches, where the parameter k is set to 18 in Equation 4.1 and Equation 4.2.

Self-comparison:

We conduct experiments with the two baseline CNNs and the CNN after PI training, see the

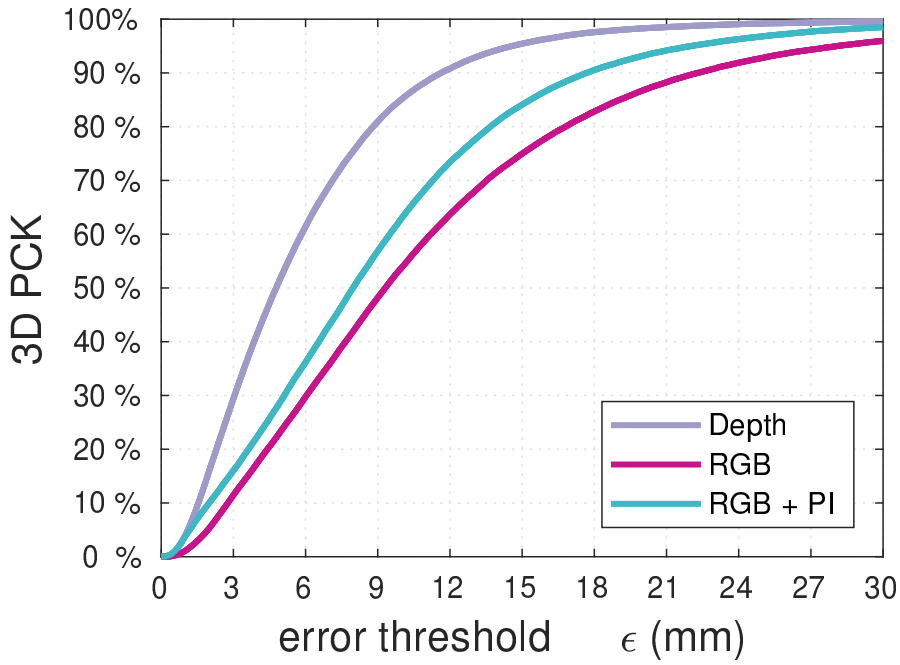


Figure 4.4: ***Results on Stereo dataset for 3D hand pose accuracy****.The plot shows the self-comparison on Stereo dataset.*

accuracy curves in Figure 4.3 and Figure 4.4. Our networks only estimate relative 3D pose from a cropped RGB image patch containing the hand, to yield 3D hand pose in world coordinates, we follow a similar procedure of [Zimmermann and Brox, 2017], *i.e*, by adding the absolute position of the root joint to our estimated results. For comparison we choose the Percentage of Correct Keypoints (PCK) over a varying threshold. Training with depth data significantly improves the performance of the RGB-based network, narrowing the gap to the depth-based network. Qualitative examples are shown in Figure 4.5 and Figure 4.6.

Comparison with the state of the art:

We compare our results with state-of-the-art methods, including PSO [Oikonomidis et al., 2011a], ICPPSO [Qian et al., a], Zhang *et al* [Zhang et al., 2016], Z&B [Zimmermann and Brox, 2017], and GANerated [Mueller et al., 2017a], see Figure 4.7. Our method outperforms all existing state-of-the-art methods. We significantly outperform (Z&B) [Zimmermann and Brox, 2017] and [Mueller et al., 2017a]. While both [Zimmermann and Brox, 2017]

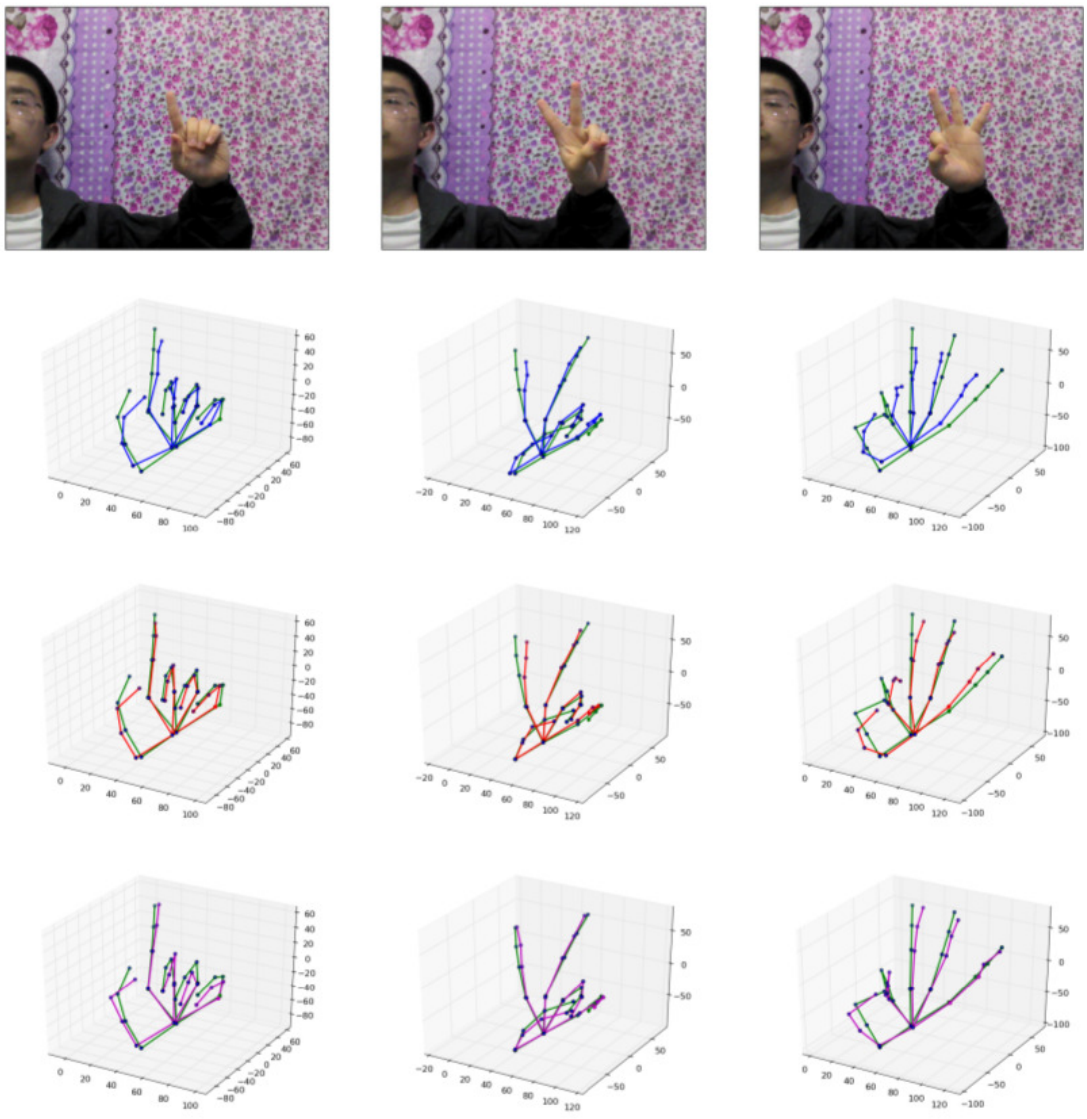


Figure 4.5: ***Qualitative 3D pose estimation results***. *Comparing the outputs of the RGB network (blue, second row), the RGB network with PI training (red, third row), and the Depth network (magenta, bottom row) with the ground truth 3D pose (green) on the Stereo dataset. Top row are the original images.*

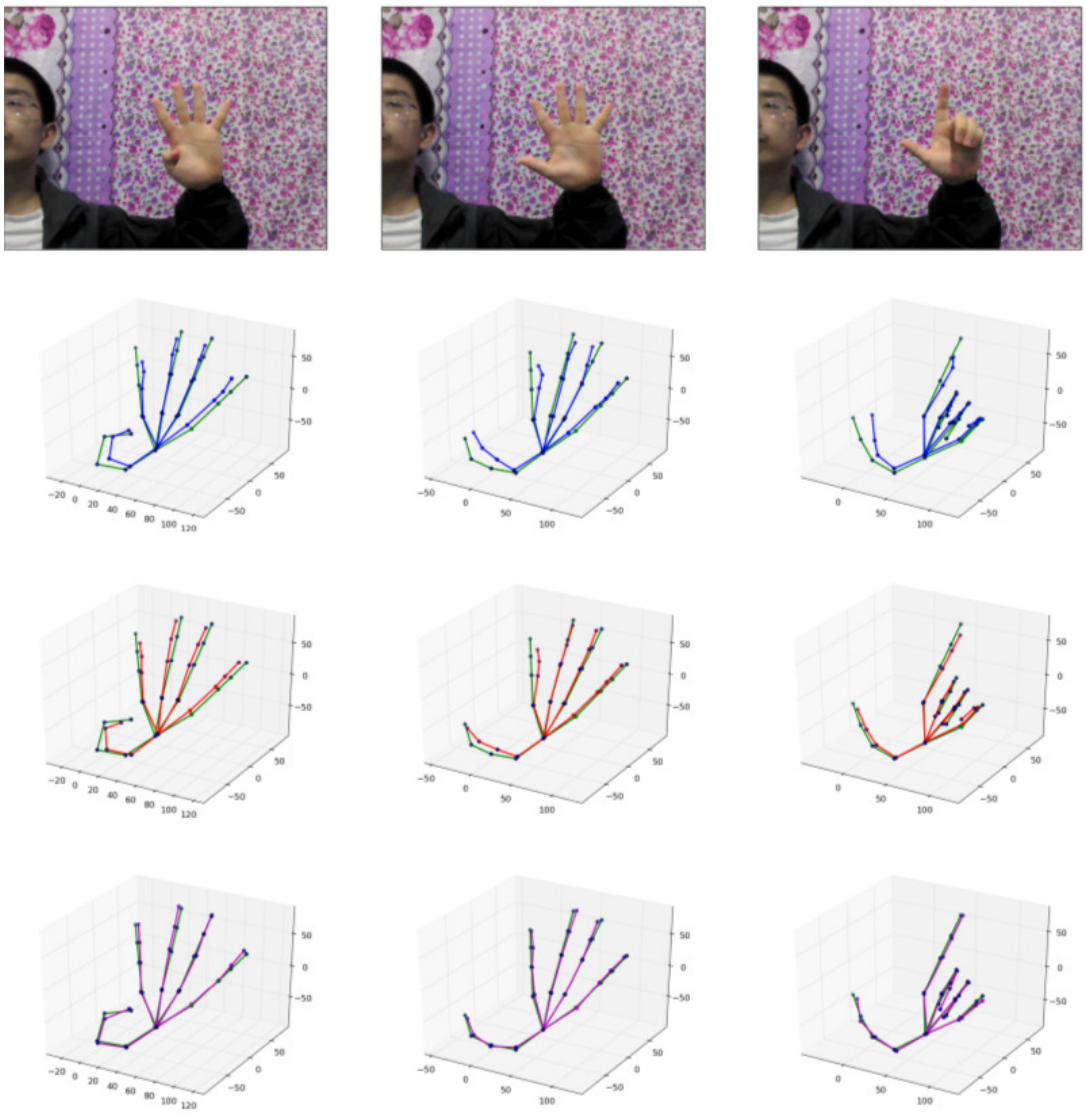


Figure 4.6: ***More qualitative results for 3D pose estimation***. *Comparing the outputs of the RGB network (blue, second row), the RGB network with PI training (red, third row), and the Depth network (magenta, bottom row) with the ground truth 3D pose (green) on the Stereo dataset. Top row are the original images.*

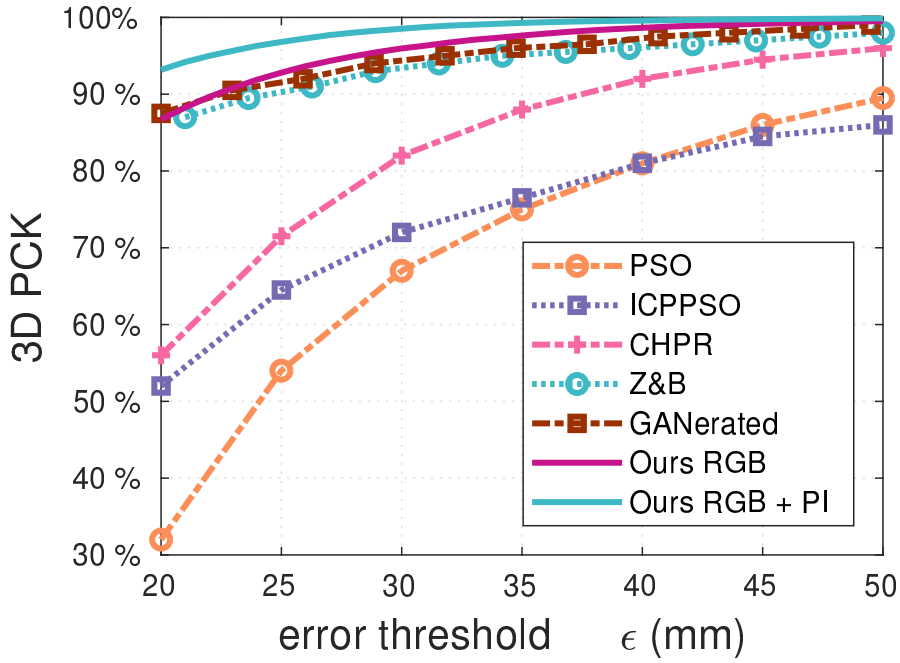


Figure 4.7: ***Results one Stereo and RHD dataset for 3D hand pose accuracy***. *The plot is comparison with state-of-the-art on Stereo dataset.*

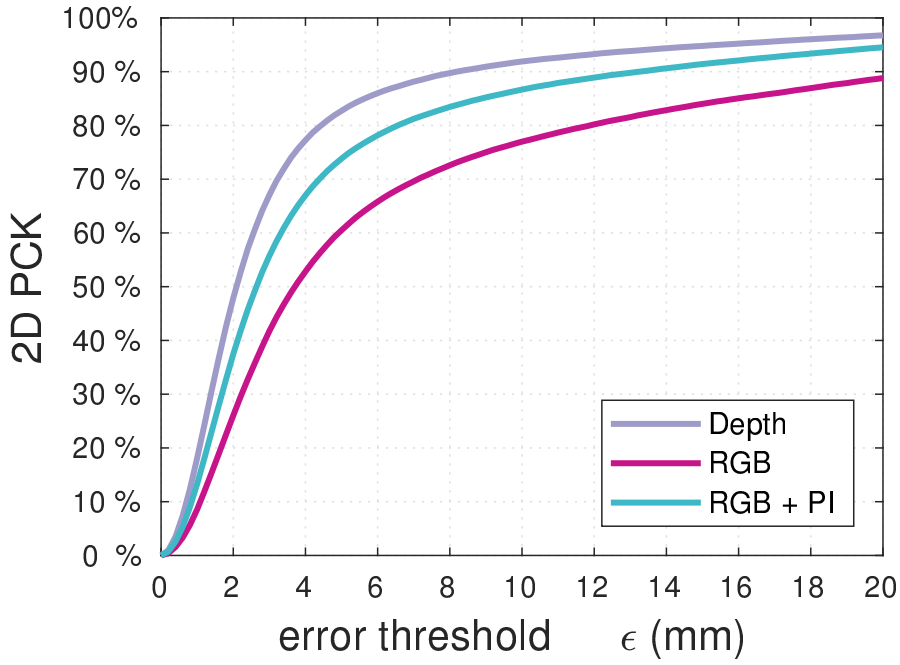


Figure 4.8: ***Results one Stereo and RHD dataset for 2D pose accuracy***.*The plot is self comparison on RHD dataset.*

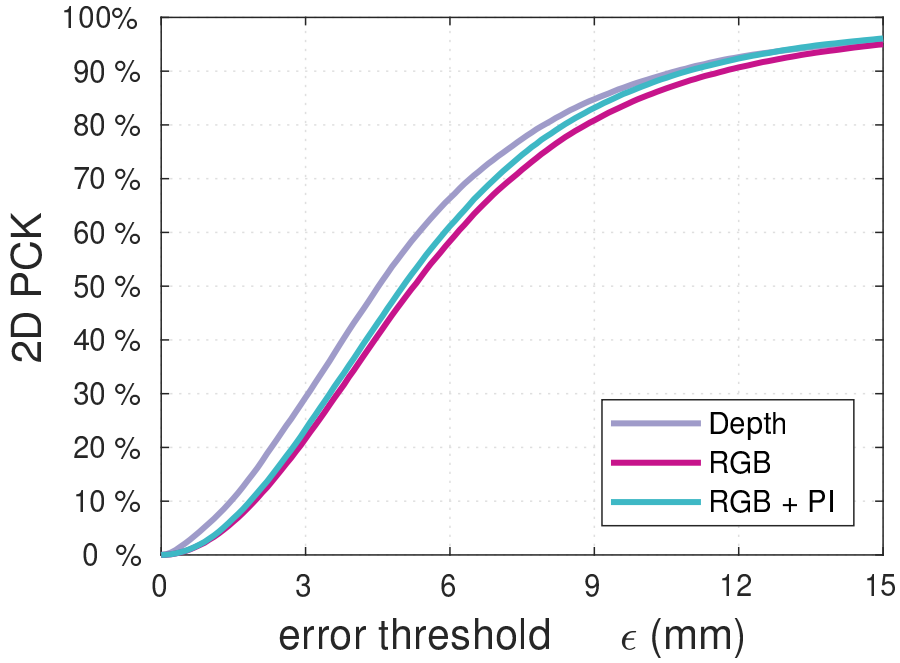


Figure 4.9: ***Results one Stereo and RHD dataset for 2D pose accuracy****.The plot is self comparison on Stereo dataset.*

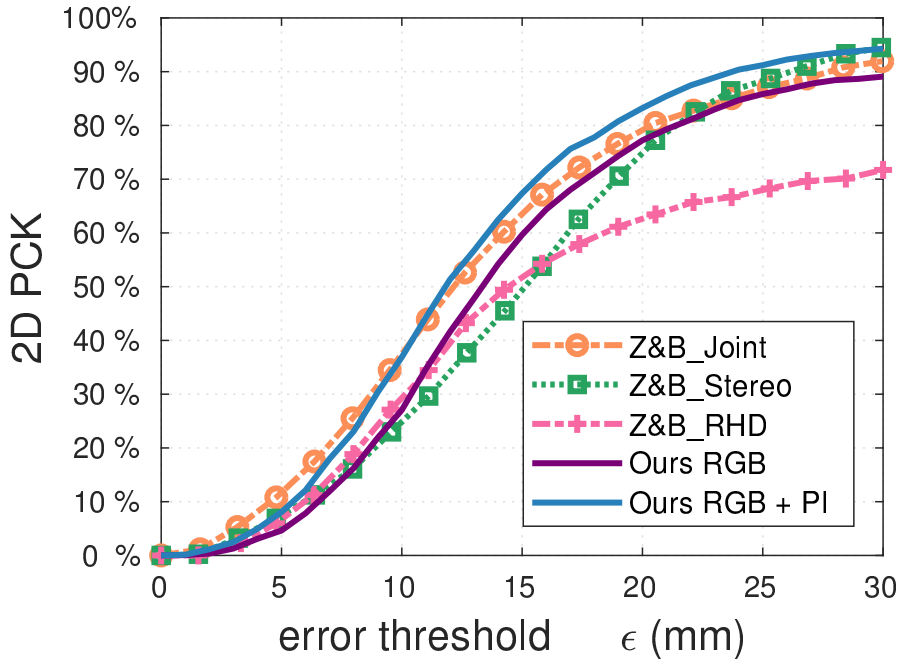


Figure 4.10: ***Results one Stereo and RHD dataset for 2D pose accuracy****.The plot is comparison with state-of-the-art on the Dexter-Object dataset.*

and [Mueller et al., 2017a] used extra training data, [Zimmermann and Brox, 2017] used both Stereo (real) and RHD (synthetic) data to train their network. [Mueller et al., 2017a] used synthetic (GANerated) data to train their network. The proposed method uses less RGB training data and achieved the best performance. we significantly outperformed both methods with our privileged training strategy.

Feature activation maps:

To give more intuitions on the effectiveness of training using additional privileged information, we visualize the activations of the mid-level feature for the three networks. Feeding an RGB image into each network, we aggregate all the mid-level feature maps into feature map by taking the maximum across all feature maps (similar to the maxout operation [Goodfellow et al., 2013]). As shown in Figure 4.11 and Figure 4.12, training with privileged information helps to select more representative features, where the visualized activations are close to the foreground (the hand).

Loss function evolution: We keep a record of the loss during our training on the Stereo dataset, see Figure 4.13. The loss for 3D hand pose (left plot) of the RGB network on the test data converges at iteration 15,000, we continue training for another 5,000 iterations. From iteration 20,000, we fix the depth network parameters and connect mid-level features between the RGB and depth networks, and continue training by minimizing the joint loss (right plot) using RGB-D image pairs. The intermediate loss (middle plot) is used to suppress the difference between the mid-level feature between the RGB and depth networks. Loss for 3D hand pose of the RGB network, and the joint loss stop decreasing at around iteration 30,000.

4.4.2 2D hand pose estimation from RGB

2D hand pose estimation from RGB images is also an related application. We also try to explore the idea of using privileged information to help hand pose estimation from RGB images. In this section, we choose the base CNN model as CPM [Wei et al., 2016], which has shown great performance for 2D human pose estimation [Wei et al., 2016], and 2D hand pose estima-

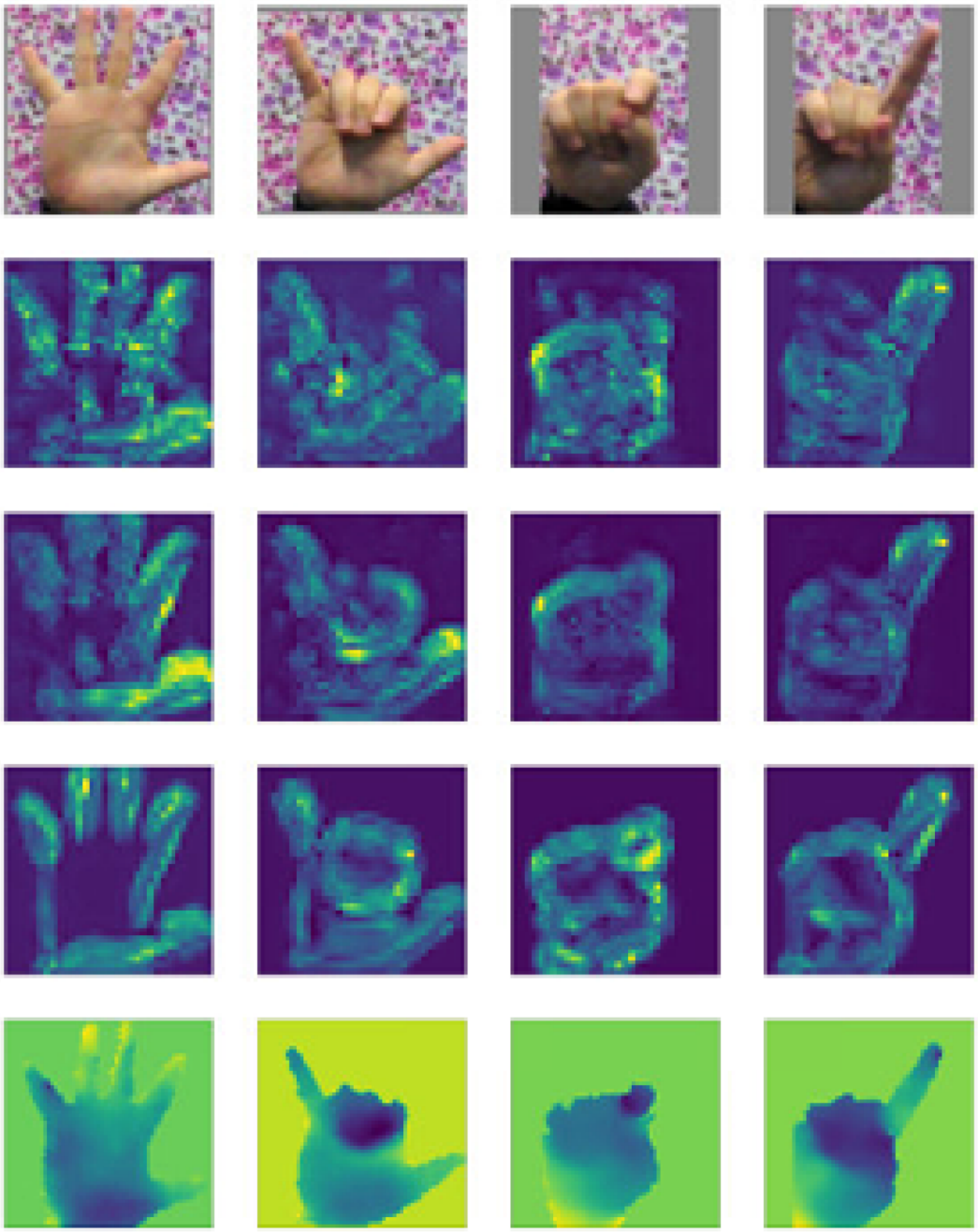


Figure 4.11: ***Feature activation maps***. *(top row) input images, (row 2) activations of the RGB network trained on RGB only, (row 3) activations of the RGB network trained with additional depth data, (row 4) activations of the depth network, and (row 5) depth images. During training, depth data helps the RGB network focus on the region of interest, reducing the influence of background regions.*

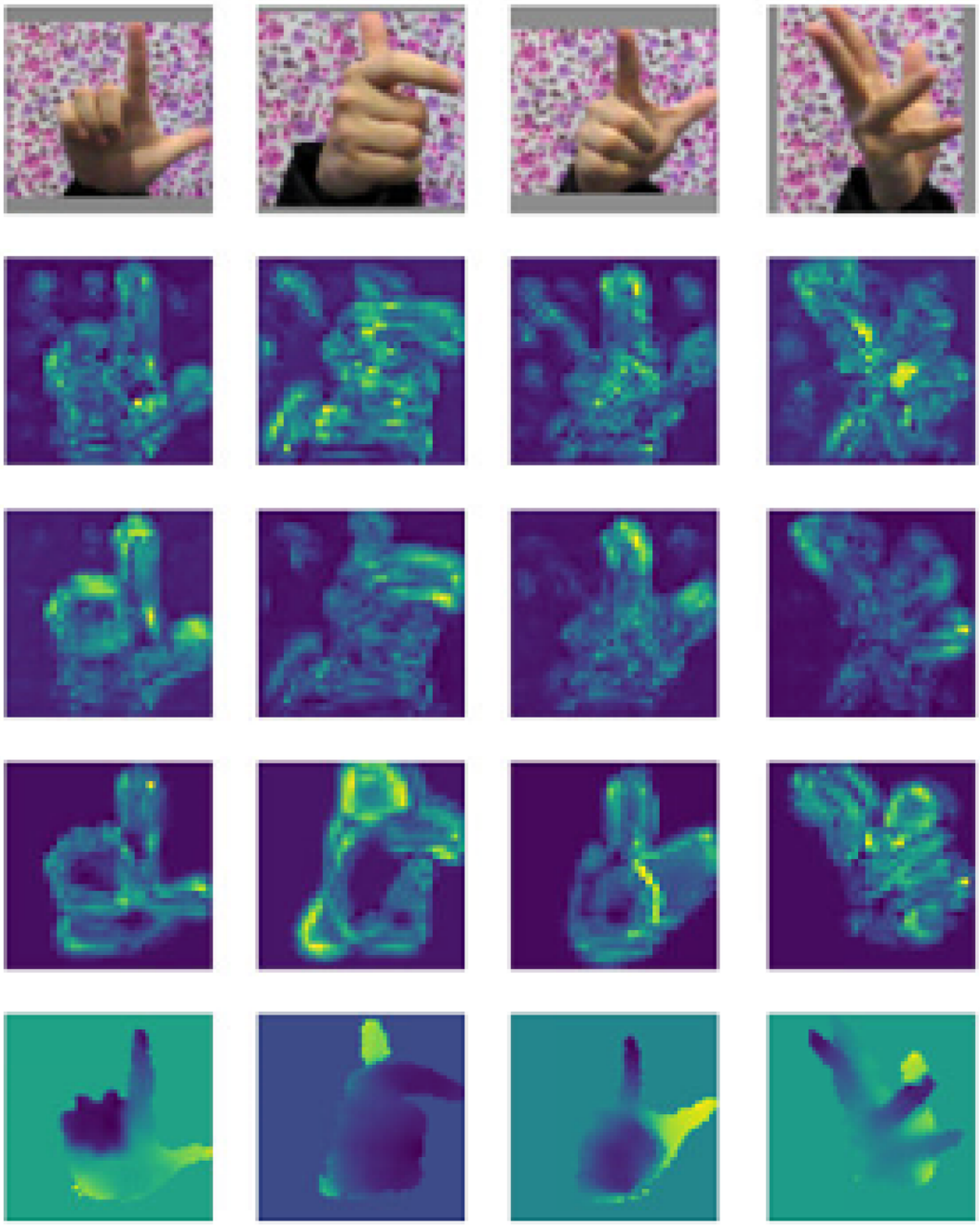


Figure 4.12: ***More feature activation maps***. *(top row) input images, (row 2) activations of the RGB network trained on RGB only, (row 3) activations of the RGB network trained with additional depth data, (row 4) activations of the depth network, and (row 5) depth images. During training, depth data helps the RGB network focus on the region of interest, reducing the influence of background regions.*

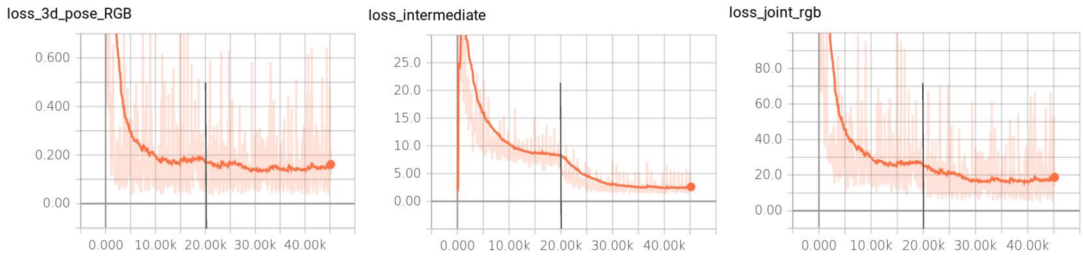


Figure 4.13: ***Loss function evolution on the stereo dataset [Zhang et al., 2016].*** *The loss for 3D hand pose (left plot) of the RGB network on the test data converges at iteration 15,000, we continue training for another 5,000 iterations. From iteration 20,000, we fix the depth network parameters and connect mid-level features between the RGB and depth networks, and continue training by minimizing the joint loss (right plot) using RGB-D image pairs. The intermediate loss (middle plot) is used to suppress the difference between the mid-level feature between the RGB and depth networks. Loss for 3D hand pose of the RGB network, and the joint loss stop decreasing at around iteration 30k.*

tion [Zimmermann and Brox, 2017]. Results are reported in Table 4.2, where 'EPE' stands for the 'average end point error' in pixels and an end point is a hand joint. In this part of experiments, we treat the hand mask as privileged data, the CNN base model is CPM [Simon et al., 2017], self-comparisons are shown in Figure 4.8 and Figure 4.9. The baseline is obtained by the normal training procedure, *i.e*, feeding the pre-processed hand image into CPM and obtaining the 2D hand pose by finding out the maximum location in each of the 21 heatmaps. For training with privileged information, we randomly select a certain proportion of RGB training data and use the hand masks, which are obtained by thresholding the depth images, in the *Loss_Mask* to suppress the background responses. As shown in Table 4.2, where 0.2 and 0.8 denotes the percentage of images when the *Loss_Mask* is used during the training for 2D hand pose estimation.

Performance on hand-object interaction dataset: In Figure 4.10, we show a comparison in terms of 2D PCK (in pixels) on the Dexter-Object [Sridhar et al., 2016] dataset. *Z&B_Joint* denotes the method of Z&B [Zimmermann and Brox, 2017] trained on both RHD and Stereo datasets, which is better than *&B_Stereo* (trained on Stereo) and *Z&B_RHD* (trained on RHD). Our approach outperformed *Z&B_Joint* even though we used less RGB training data.

Method	Testing	Training	EPE median	EPE mean
Z&B [Zimmermann and Brox, 2017]	RHD	RHD+Stereo	5.001	9.135
Baseline RGB	RHD	RHD	3.708	7.841
Baseline Depth	RHD	RHD	2.087	3.902
RGB + PI training	RHD	RHD	2.642	5.223
Z&B [Zimmermann and Brox, 2017]	Stereo	RHD+Stereo	5.522	5.013
Baseline RGB	Stereo	Stereo	5.250	6.533
Baseline Depth	Stereo	Stereo	4.775	5.883
RGB + PI training (0.2)	Stereo	Stereo	5.068	6.280
RGB + PI training (0.8)	Stereo	Stereo	4.515	5.801
Z&B [Zimmermann and Brox, 2017]	Dexter-Object	RHD+Stereo	13.684	25.160
Baseline RGB	Dexter-Object	RHD	13.360	18.278
RGB + PI training	Dexter-Object	RHD	11.809	14.593

Table 4.2: ***2D Hand Pose Accuracy***. *Results when training on the RHD and Stereo datasets.*

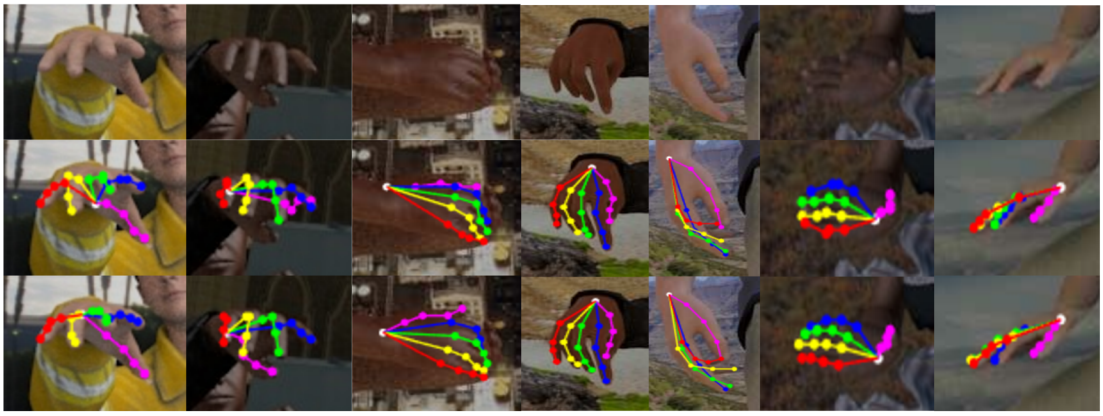


Figure 4.14: ***Qualitative 2D pose estimation results***. *Comparing the outputs of (middle) the RGB network and (bottom) the RGB network with PI training on the RHD dataset. Top row are the original images.*

4.5 Discussion and conclusions

In this chapter, we proposed a framework for 3D hand pose estimation from RGB images, with the training stage aided with privileged information, *i.e*depth data. To the best of our knowledge, our method is the first to introduce the concept of using privileged information (depth images) to support training a RGB-based 3D hand pose estimator. We proposed three ways to use the privileged information: as external training data for a depth-based network branch, as paired depth data to transfer supervision from the depth-based network to the RGB-



Figure 4.15: ***Qualitative 2D pose estimation results on Stereo dataset****. Comparing the outputs of (top) the RGB network and (bottom) the RGB network with PI training.*

based network, and as a hand mask to suppress background activations in the RGB-based network. Our training strategy can be easily embedded into existing pose estimation methods. As an illustration, we estimate 2D hand pose from an RGB image using a different CNN model. Results on 2D hand pose estimation, using our training strategy, are improved over state-of-the-art methods for 2D hand pose estimation from RGB input. During testing, when only RGB images are available, our model significantly outperforms the same model trained only using RGB images. This training strategy can be incorporated into existing models to boost the performance of hand pose estimation from an RGB image.

One limitation of our method is the difficulty of handling occlusion by objects, which can be addressed by systematically adding synthetic objects in the depth data (privileged information). Future work includes testing the idea of iteratively freezing parameters of the two branches of *RGB_Net* and *Depth_Net*, *i.e*, iteratively treating RGB and depth images as privileged information, quantifying the effect of data augmentation to the existing RGB training data, and extending the current work to 3D hand tracking by exploiting the hand shape prior and fitting a hand model.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Contents

5.1 Contributions and Relationship between Chapters 105

5.2 Future Work 108

5.1 Contributions and Relationship between Chapters

This thesis makes consistent contributions in three key sections for 3D hand pose estimation: methods, datasets, and challenges. Contributions from each chapter are made based on new discoveries in the literature.

5.1.1 Chapter 2

After finding out the bottleneck was the lack of a large scale dataset to deploy the Deep Learning models, *i.e*, the development in datasets lagged behind the algorithm advancement, Chapter

2 proposes a million-scale benchmark dataset of real hand depth images. The reason of lacking such a large scale dataset lies in the current annotation methods, whether it is labour-intensive manual annotation or semi-automatic capture methods followed with manual refinement. The only way to build a million scale annotated dataset accurately, which is beyond the capability of human annotation, is to develop an automatic annotation method. Following this direction, we found out that with the help of active 6D magnetic sensors along with a hand skeleton model, we were able to design an automatic annotation system with minimal restriction on the range of motion. To capture all possible hand articulations we designed a hand movement scheme. We first defined 32 *extremal poses* as hand poses where each finger assumes a maximally bent or extended position. For maximum coverage of the articulation space, we enumerated all $\binom{32}{2}$ = 496 possible pairs of these *extremal poses*, and captured the natural motion when transitioning between the two poses of each pair. To cover diverse view points, we varied the sensor height, the subject's position and arm orientation. The view point space (a hemisphere for the 3rd person view point) was divided into 16 regions (4 regions uniformly along each of two 3D rotation axes), and subjects were instructed to carry out random view point changes within each region. In addition, our dataset collected random changes in the egocentric view point. This new dataset, which made significant advancement in terms of completeness of hand data variations and quality of full annotations, allowed us to train a Convolutional Neural Network model that is able to generalize well to existing test datasets and to egocentric viewpoint. The contribution to the research society is that it will help to further advance the research field, allowing the exploration of new approaches.

5.1.2 Chapter 3

Chapter 3 makes detailed analysis of the participating methods in the *HIM2017* challenge from Appendix. This chapter conducts a series of analyses, looking at how methods perform in these scenarios. The useful conclusion drawn from this chapter includes: (1) 3D volumetric representations used with a 3D CNN show high performance than corresponding 2D CNN. (2) Making use of richer spatial models, regression-based methods can achieve good perfor-

mance. (3) Explicit modeling of structure constraints and spatial relation between joints can narrow the gap between errors on visible and occluded joints. (4) Isolated 3D hand pose estimation achieves low mean errors (10 millimeter) in the view point range of [70, 120] degrees. Multi-stage methods [Molchanov et al., 2017] tend to perform better in extreme view points.

5.1.3 Chapter 4

Chapter 4 argues that 3D hand pose estimation from an RGB image, which has much wider range of applications considering the ubiquitous availabilities of RGB cameras in our daily life, lagged far behind compared with that from a depth image, with only a few papers [Simon et al., 2017, Zimmermann and Brox, 2017, Mueller et al., 2017b, Panteleris and Argyros, 2017, Spurr et al., , Iqbal et al.,] published in the last years. For 3D hand pose estimation from an RGB image, the main obstacle is the lack of proper RGB datasets. Current state-of-the-art methods opt for synthetic datasets [Zimmermann and Brox, 2017], or using GAN [Goodfellow et al., 2014] to generate training data [Shrivastava et al., 2017, Mueller et al., 2017b]. This thesis proposes a method to leverage the existing large scale depth datesets for hand pose estimation from RGB images that uses both external large-scale depth image datasets and paired depth and RGB images as privileged information at training time. By exploring different ways of using this privileged information, this chapter tries to supervise the RGB-based network through a privileged training scheme, where the RGB-based network learns middle layer features that mimic the corresponding depth-based network, which is well trained with the available large-scale depth dataset. For testing, the proposed method produces accurate 3D hand pose predictions on RGB images.

5.1.4 Appendix

The BigHand2.2M dataset proposed in Chapter 2 provided the opportunities for deploying large deep learning models. Appendix states that even though with significant progress being made in the aspects of dataset and systems and a few comparisons have been made among

different systems and on a few small datasets, there lacked a systematic comparisons for these systems. Appendix proposes the *Hands in the Million Challenge (HIM2017)* with the intention of comparing existing and potential systems in a large scale and for fine-grained discussions. *HIM2017* [Yuan et al.,] provided the benchmark dataset includes data from *BigHand2.2M* [Yuan et al., 2017] and the *First-Person Hand Action dataset (FHAD)* [Garcia-Hernando et al.,], allowing the comparison of different algorithms in a variety of settings. The challenge considered three different tasks: single-frame pose estimation, tracking, and hand-object interaction. The challenge attracted 17 participating teams from top labs across the world, and provided the test bed for the analysis in Chapter 3.

5.2 Future Work

3D hand pose estimation has seen great improvements in the past few years, especially with the popularity of the off-the-shelf depth cameras, and with many methods and datasets published. One limitation of current work is the difficulty of handling occlusion by objects, which can be addressed by adding synthetic objects in the depth data systematically. Random forests [Tang et al., 2013, Tang et al., 2014, Tang et al., 2015, Wan et al., 2016, Li et al.,] and more recently CNNs [Ye et al., 2016, Wan et al., 2017, Ge et al., 2016b, Ge et al., 2017] have been applied to hand pose estimation. Works combining Random Forests and Deep Learning could be interesting. Random Forest is a good technique to do regression, especially when the search space has low dimensions. A human hand has 26 degrees of freedom (6 for global location and orientation, 20 for finger joint abduction and flexion). Regression in a 26-D space would be too challenging for Random Forest alone, even though many attempts have been made [Tang et al., 2013, Tang et al., 2014, Tang et al., 2015]. HSO [Tang et al., 2015] breaks the whole 26-D into several sub-spaces to make it easier for Random Forests. Based upon HSO [Tang et al., 2015], Qi *et al.* [Ye et al., 2016] replaced the Random Forests with CNNs to build a hierarchical CNN model. Both methods followed a predefined hierarchy for the hand joints, even though the predefined hierarchy could be good only for certain hand poses, *e.g.* an open palm, it may not good for other hand poses, *e.g.* a fist. A method that can use Random Forests to dynamically

build an unique hierarchy for each hand pose may be promising, with the aim of improving the all hand poses, not just certain hand poses that fit a predefined hierarchy. Within a certain subspace, CNNs can be introduced to achieve better results.

RGB-based 3D hand pose estimation is also gaining momentum in the past two years, starting from estimating relative hand poses to predicting global hand poses. As discussed in Chapter 4, the current method can achieve 9 mm average error. It could be fine for certain gestures, *e.g.* waving and pointing. But for fine manipulations in AR and VR, *e.g.* playing piano, which may require less than 1 mm average error, would still be challenging. One of the challenges for RGB-based hand pose estimation is that it's difficult to find the distance information from a single RGB image, *e.g.* a hand from an RGB image could be a small hand close to the camera, or it could be a big hand further away from the camera.

Future work includes testing the idea of iteratively freezing parameters of the two branches of *RGB_Net* and *Depth_Net*, *i.e.* iteratively treating RGB and depth images as privileged information, quantifying the effect of data augmentation to the existing RGB training data, and extending the current work to 3D hand tracking by exploiting the hand shape prior and fitting a hand model. Knowledge distillation across different modalities, *e.g.* paired RGB and depth images (un-annotated), is an interesting direction to pursue. Hand pose estimation from RGB image with depth estimation (from RGB image) as a intermediate step could also be an interesting direction, given the following facts: (1) hand pose estimation from depth has been more developed, and (2) the improvment in depth estimation from RGB images are also incresing [Eigen et al., 2014, Xu et al., 2018, Fu et al., 2018].

Appendix

A HAND IN THE MILLION CHALLENGE

Contents

A.1 Overview 111

A.2 Challenge tasks 114

A.3 Dataset details 114

A.4 Participation rules 117

A.5 Discussion and conclusions 122

A.1 Overview

In this appendix, we present the 2017 Hands in the Million Challenge, a public competition designed for the evaluation of the task of 3D hand pose estimation. The goal of this challenge is to assess how far is the state of the art in terms of solving the problem of 3D hand pose estimation as well as detect major failure and strength modes of both systems and evaluation metrics that can help to identify future research directions. The challenge follows up the recent publication of BigHand2.2M [Yuan et al., 2017] and First-Person Hand Action [Garcia-Hernando et al.,]

datasets, which have been designed to exhaustively cover multiple hand, viewpoint, hand articulation, and occlusion. The challenge consists of a standardized dataset, an evaluation protocol for two different tasks, and a public competition. In this appendix we describe the different aspects of the challenge and, jointly with the results of the participants, it was presented at the 3rd International Workshop on Observing and Understanding Hands in Action, HANDS 2017, with ICCV 2017.

There has been significant progress in the area of 3D hand pose estimation in the last years [Oikonomidis et al., 2011a, Tang et al., 2013, Tang et al., 2014, Qian et al., b, Sharp et al., 2015, Tang et al., 2015, Li et al., , Wan et al., 2016, Oberweger et al., , Ye et al., 2016, Wan et al., 2017, Simon et al., 2017], however, as noted in [Yuan et al., 2017], the field lacks a systematic public benchmark for fair evaluation of different methodologies. Public benchmarks and challenges in other areas such as ImageNet [Russakovsky et al., 2015] for scene classification and object detection, PASCAL [Everingham et al., 2015] for semantic and object segmentation or VOT challenge [Kristan et al., 2015] for visual object tracking, outlined a good general picture of the performance of different methodologies, with the extra competitive aspect that motivated researchers to obtain the best results and thus pushing the research activity in these fields. Motivated by this, we propose the *2017 Hands in the Million Challenge on 3D Hand Pose Estimation*.

The challenge consists of a dataset containing more than a million fully annotated images for two different tasks (tracking and single frame hand pose estimation), a standardized evaluation protocol, and a public competition. The dataset images have been sampled from the two recently proposed datasets: *BigHand2.2M* dataset [Yuan et al., 2017] and First-Person Hand Action dataset (*FHAD*) [Garcia-Hernando et al.,]. Images from *BigHand2.2M* dataset conform the core of the challenge and cover large range of hand viewpoints (including third and first-person viewpoints), hand configurations, and hand shapes in an occlusion-free setting. A smaller number of sequences extracted from *FHAD* aim to evaluate hand pose estimation in the presence of severe occlusion caused by objects, a more realistic scenario where such a benchmark does not currently exist. Participants of the challenge will receive full annotations for the training set, but the annotations for the test set will be kept secret until the presentation of the

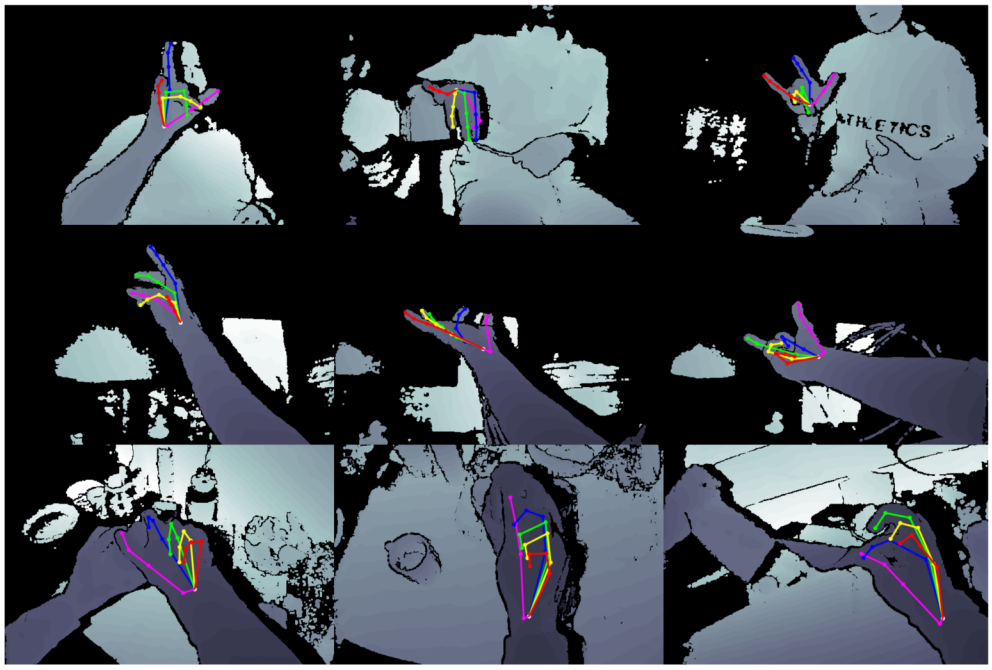


Figure A.1: ***Example images of the challenge. Top row:*** *third-person viewpoint hand poses.* ***Middle row:*** *first-person viewpoint hand poses in object-free scenario.* ***Bottom row:*** *first-person viewpoint hand poses involving manipulated objects.*

results in the 3rd International Workshop on Observing and Understanding Hands in Action, HANDS 2017, that will be hosted with the International Computer Vision Conference (ICCV) 2017, Venice, Italy.

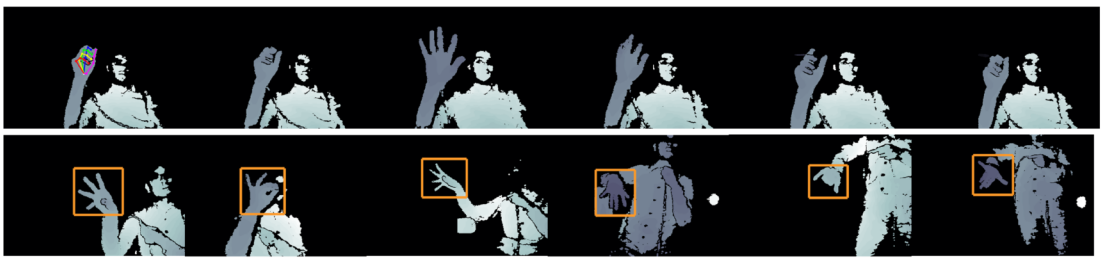


Figure A.2: ***Challenge tasks. Top row: 3D hand pose tracking****, the first frames of sequences are fully annotated.* ***Bottom row: 3D hand pose estimation****, each frame is annotated with a bounding box and frames are shuffled. The objective of both task is to infer the 3D position of the 21 joints at each given depth image.*

For up-to-date information, please visit the challenge website: http://icvl.ee.ic.ac.uk/hands17/challenge/

A.2 Challenge tasks

We present the three tasks evaluated in this challenge: 3D hand pose tracking, 3D hand pose estimation, and Hand-object interaction 3D hand pose estimation. See *Figure* A.2 for illustration.

3D hand pose tracking: This task is performed mainly on sequences of 2700-3300 frames each and a few short sequences of 150 frames each. Given the full hand pose annotation in the first frame, the system should be able to track the 21 joints' 3D locations in the whole sequence.

3D hand pose estimation: This task is performed on individual images, each image is randomly selected from a sequence and the bounding box of the hand area is provided. The system should be able to predict the 21 joints' 3D locations for each image.

Hand-object interaction 3D hand pose estimation: We provide 2965 frames of fully annotated hand interacting with different objects (e.g, juice bottle, salt bottle, knife, milk bottle, soda can, etc.) All the images are captured in egocentric setting. The system should be able to predict the 21 joints' 3D locations for each image.

A.3 Dataset details

The dataset is created by sampling images and sequences from *BigHand2.2M* dataset [Yuan et al., 2017] and First-Person Hand Action dataset (*FHAD*) [Garcia-Hernando et al.,], both datasets are fully annotated (21-joints) using an automatic annotating system with six 6D magnetic sensors [NDI trakSTAR,] and inverse kinematics. The depth images are captured with the latest Intel RealSense SR300 camera [Intel SR300,] at 640×480-pixel resolution. In the following subsections we expand on how the dataset has been constructed, see *Table* A.1. For more detailed information about the datasets, we refer the reader to the original papers [Yuan et al., 2017, Garcia-Hernando et al.,].

# of	Scenarios	Training	Tracking	Estimation
subjects	3rd view	5	10	10
	ego view	5	10	10
	action	0	0	2
seen	3rd view	5	5	5
subjects	ego view	5	5	5
	action	0	2	2
unseen	3rd view	0	5	5
subjects	ego view	0	5	5
	action	0	0	0
sequences	3rd view	30	67	67
	ego view	5	32	33
	action	0	-	-
frames	3rd view	873K	187K	187K
	ego view	83K	109K	109K
	action	0	-	2965

Table A.1: ***Size of the challenge data splits:*** *number of subjects, sequences, and frames.*

A.3.1 Training data

The training set is built entirely by sampling the *BigHand2.2M* dataset. This dataset contains ten subjects covering three different nature of hand poses: (1) schemed poses, (2) random poses, and (3) egocentric poses. In this challenge, we pick five out of ten subjects to build the training set, see *Figures* A.5 and A.4. The training set consists of two parts of these five chosen subjects (denoted as **seen subjects**): (1) schemed poses, and (2) half of the egocentric poses, which are chosen by splitting the egocentric pose sequences into halves and selecting the first half. The training data is randomly shuffled to remove temporal information. 21 joints ground truth annotation is provided.

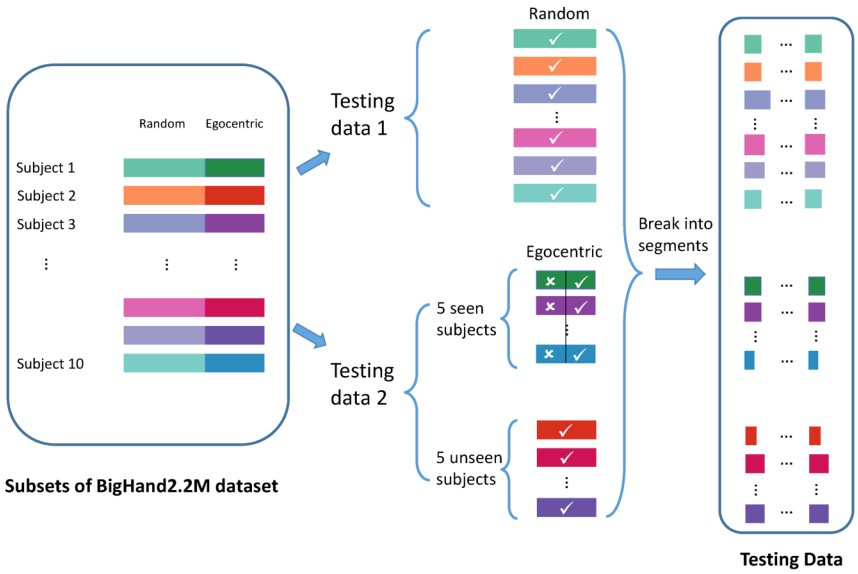


Figure A.3: ***Test data****. The test images consists of two part, (1) these images with random poses for all the ten subjects, (2) these images with egocentric poses for five unseen subjects and the second halves of egocentric poses for five seen subjects.*

A.3.2 Test data

The test data consists of three parts: (1) random hand poses of ten subjects (five seen in the training data and five unseen), (2) egocentric object-free hand poses (five seen in the training data and five unseen), (3) egocentric with object hand poses (from the FHAD dataset).

In *Figure* A.3 we show how (1) and (2) are built from *BigHand2.2M*. Test data is divided into half and half for each task.For examples of the images for each subset of data see Figure A.10, Figure A.11, Figure A.8, Figure A.9, Figure A.12, and Figure A.12.

3D hand pose tracking: the test data is segmented into small segments of consecutive frames with 21 joints ground truth annotations provided for the initial frame. In this task, there are 99 segments from *BigHand2.2M* dataset, each has 2700-3300 consecutive frames, and a few short sequences of 150 frames per each from *FHAD*.

3D hand pose estimation: the test data is randomly shuffled to remove motion information, with hand bounding box provided for each frame. In total, there are around 296K frames

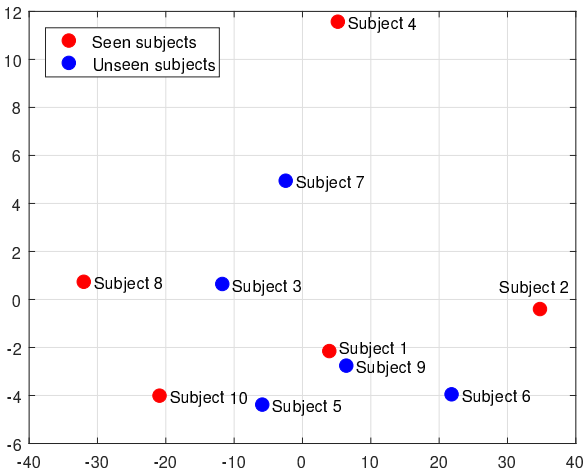


Figure A.4: ***Seen vs Unseen subjects***. *Seen subjects are chosen to build the training data. The plot shows the first two principal components of the hand shape parameters.*

Scenarios	3D pose tracking	3D pose estimation
3rd view	seen subjects unseen subjects	seen subjects unseen subjects
egocentric view	seen subjects unseen subjects	seen subjects unseen subjects
action	-	seen subjects

Table A.2: ***Test scenarios.*** *Detailed evaluation will be performed in different scenarios.*

of test data in this task.

Detailed evaluation will be performed for different methods in different scenarios as shown in *Table* A.2 after receiving the teams' submissions. However, during the challenge no information about seen subjects, unseen subjects, viewpoint or object are provided.

A.4 Participation rules

To have your method evaluated, run it on all of the test sequences and submit the results in the same format as that of the annotation training data. Check the challenge submission website

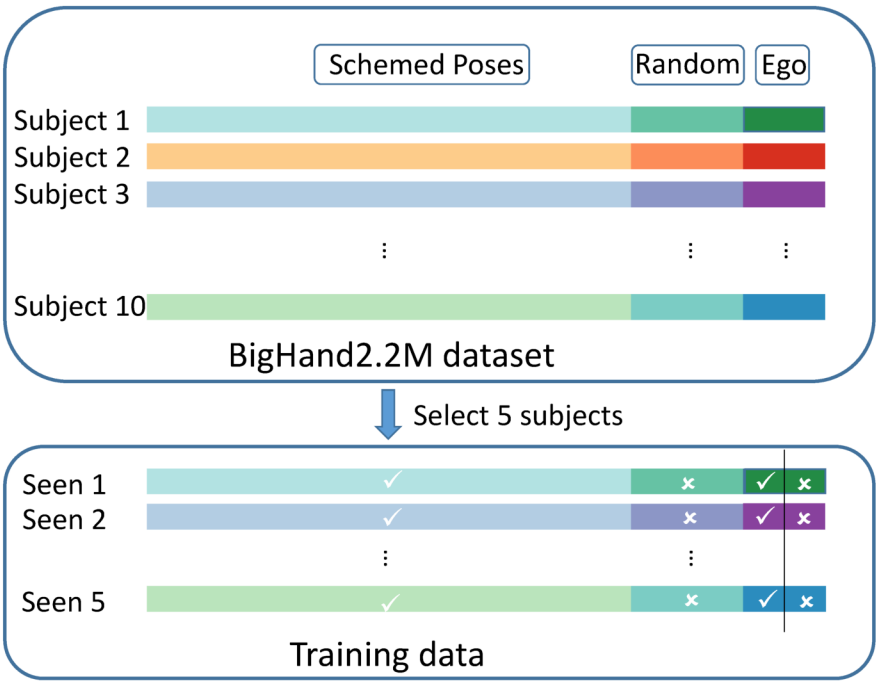


Figure A.5: ***Training data***. *To build the training data, five out of ten subjects in BigHand2.2M are selected. The training data consists of uniformly sampled schemed poses and the half of the egocentric poses.*

for detailed instructions on how to submit your results.

A.4.1 Challenge rules

- Only one submission per day per team is allowed. We will try to update the website regularly with the leaderboard for different metrics. Only the best result of each team will be posted.
- For each submission, you **must** keep the parameters of your method constant across all test data.
- If you want your results to be included in a publication about the challenge, a documentation of results is required. Without the documentation, your results will be listed

on the website but not included in the publication. The documentation must include an overview of the method with a related publication if it is published.

- For training, you can use the provided training images. You can also obtain extra training images by augmenting the existing images, e.g., by in-plane rotating the training images. Augmentations Any external data (other datasets, synthetic data, etc.) is **not** allowed. Any data augmentation technique must be reported in the documentation.

A.4.2 Annotation and results format

The pose annotations for each image follow the following format (in a text file):

- Each line has 64 elements, the first item is the frame name.
- The rest 63 elements are [x y z] values of the 21 joints.
- The joints are ordered in this way: [Wrist, TMCP, IMCP, MMCP, RMCP, PMCP, TPIP, TDIP, TTIP, IPIP, IDIP, ITIP, MPIP, MDIP, MTIP, RPIP, RDIP, RTIP, PPIP, PDIP, PTIP], where 'T', 'I', 'M', 'R', 'P' denote 'Thumb', 'Index', 'Middle', 'Ring', 'Pinky' fingers. 'MCP', 'PIP', 'DIP', 'TIP' are joints' names, as shown in *Figure* A.6.

A.4.3 Evaluation

The hand pose results will be evaluated using different error metrics. The aim of this evaluation is to identify what success and failure modes of different methodologies. We will use both standard error metrics and new proposed metrics that we believe will provide further insights into the performance of evaluated methodologies. For each submission, we will provide the results for each error metric and a overall score combining all of them, which will be used to decide the challenge winner and the order in the leaderboard. The evaluated metrics are detailed next:

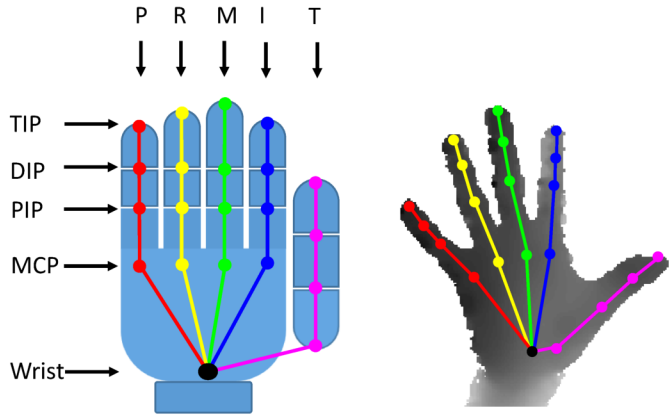


Figure A.6: ***Hand annotation format***. *The left figure shows the hand skeleton, the right figure is a real depth image with annotation. 'T', 'I', 'M', 'R', 'P' denotes 'Thumb', 'Index', 'Middle', 'Ring', and 'Pinky' finger, respectively. 'MCP', 'PIP', 'DIP', 'TIP', denotes metacarpophalangeal, proximal interphalangeal, and distal interphalangeal, and tip joints, respectively.*

A.4.3.1 Standard error metrics

Following the literature [Oikonomidis et al., 2011a, Taylor et al., 2012, Sharp et al., 2015], we use the following error metrics:

1. The mean error for all joints for each frame and average across all test frames [Oikonomidis et al., 2011a].

2. The ratio of joints within a certain error bound [Sharp et al., 2015] defined as:

$$r_j = \frac{N_j}{N * n}, \tag{A.1}$$

   where N is total number of frame, n is the number of joints of a hand (21 in this challenge), $N_j = f(\epsilon)$ is the total number of joints within a euclidean distance of ϵ to the ground truth. Accuracy curve will be drawn by varying the value of ϵ

3. A more challenging one, the ratio of frames r_f that have all joints within a certain distance to ground truth annotation [Taylor et al., 2012] defined as:

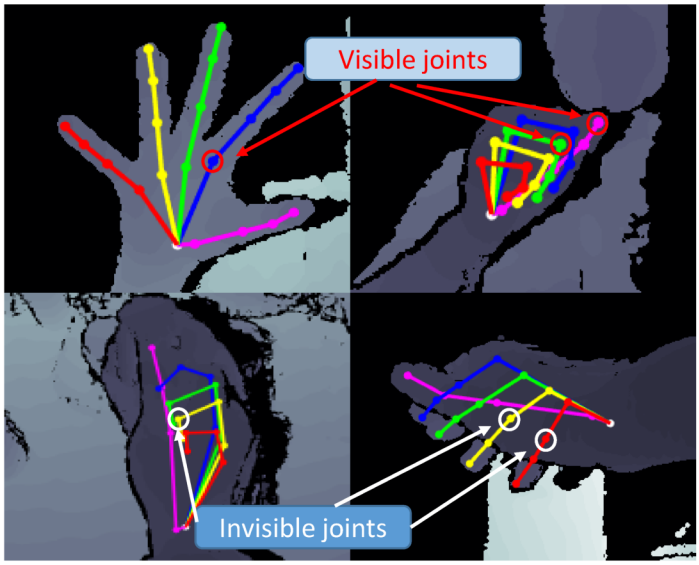


Figure A.7: ***Joint visibility illustration***. *Top-left: all joints are visible. Top-right: 'TMCP', 'TPIP' joints are invisible, they are occluded by other fingers. Bottom-left: 'RDIP', 'RTIP', 'MDIP', 'MTIP' are occluded by the object. Bottom-right: 'PPIP', 'RPIP','MPIP', 'IDIP' are occluded by the thumb in this egocentric viewed hand pose.*

$$r_f = \frac{N_f}{N}, \tag{A.2}$$

where N is total number of frame, $N_f = g(\epsilon)$ is the number of frames whose joints are all with in euclidean distance of ϵ to the ground truth.

Scenarios	all joints	visible joints	pose frequency
mean	✓	✓	✓
r_j	✓	✓	✓
r_f	✓	✓	✓

Table A.3: ***Evaluation metrics.*** *To evaluate a method, we take into account joints visibility as well as pose happening frequency.*

A.4.3.2 Proposed error metrics

We also propose new evaluation metrics, by taking into account the joint visibility, see *Table* A.3.

Visibility: As shown in *Figure* A.7, hand pose often present occlusions, *e.g.*, self occlusion and occlusion from objects. When occlusion happens, especially in the settings of egocentric view and hand-object interaction, measuring only the quality of the visible joints can be of interest.

Scenarios	Mean (seen)	Mean (unseen)
3rd view	13.39	23.39
egocentric view	18.91	24.46
action	51.51	51.54

Table A.4: *Average errors (in mm) for different viewpoint and different subjects.*

A.5 Discussion and conclusions

A standard CNN baseline following [Yuan et al., 2017] is provided for all test data. The baseline results on different scenarios are shown in *Table* A.4.

Based on the baseline results, we draw a few preliminary conclusions that we believe can be useful for participants:

1. Large amount of data matters. In this challenge, around 592K frames of test data are evaluated, while 957K frames of training data are used. Since previously existing methods are tested on small datasets, evaluating and comparing in a larger benchmark is important to find further research directions. The result of egocentric view test data is worse than that of third-person viewpoint test data. A reason for this the third view training data is four times larger than that of the egocentric view. This again indicates that big

data is important. This can be a consequence of data imbalance between third-person and egocentric viewpoint and the higher presence of severe self-occlusion making the egocentric scenario more challenging.

2. On all three scenarios, seen subject's results are consistently better than that of unseen subjects, this indicate the baseline model have some limit to generalize to new hand shapes, see *Figure* A.4. Models with generative capability could be more robust to solve this issue, *e.g.*, models that can handle shape variations, or models with built in hand shape handling module.

3. The performance on action test data is much worse than the other two scenarios. The reason behind this is that the training data consist of only isolated hands, while the action test data are frequently occluded by interacting objects. There still exists a gap between isolated hand data and hand with objects data. Further research can be pursued in this direction.

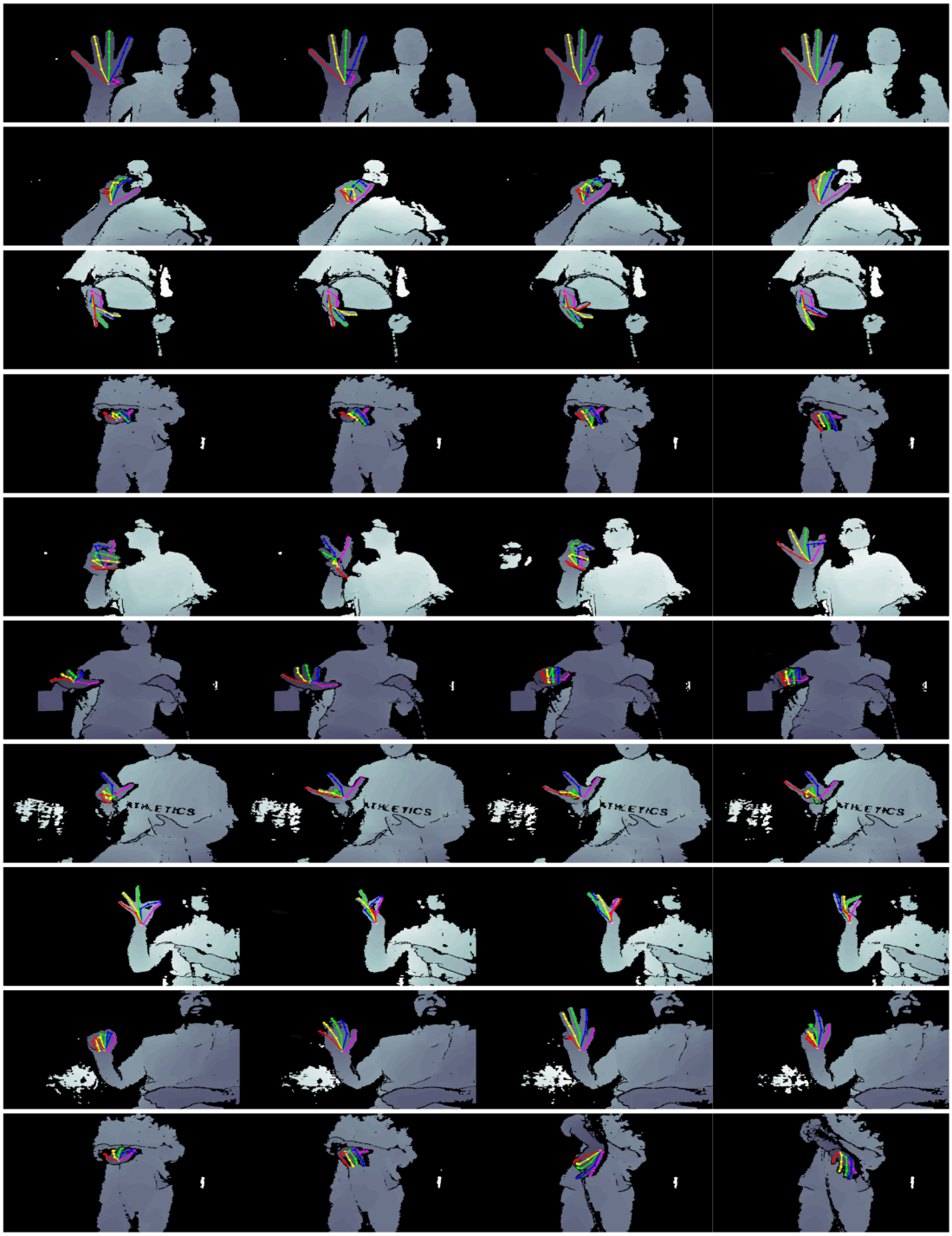


Figure A.8: ***Example sequences for third-person viewpoint.*** *The sequences are for all subjects in BigHand2.2M [Yuan et al., 2017], each row shows one sequence of a subject.*

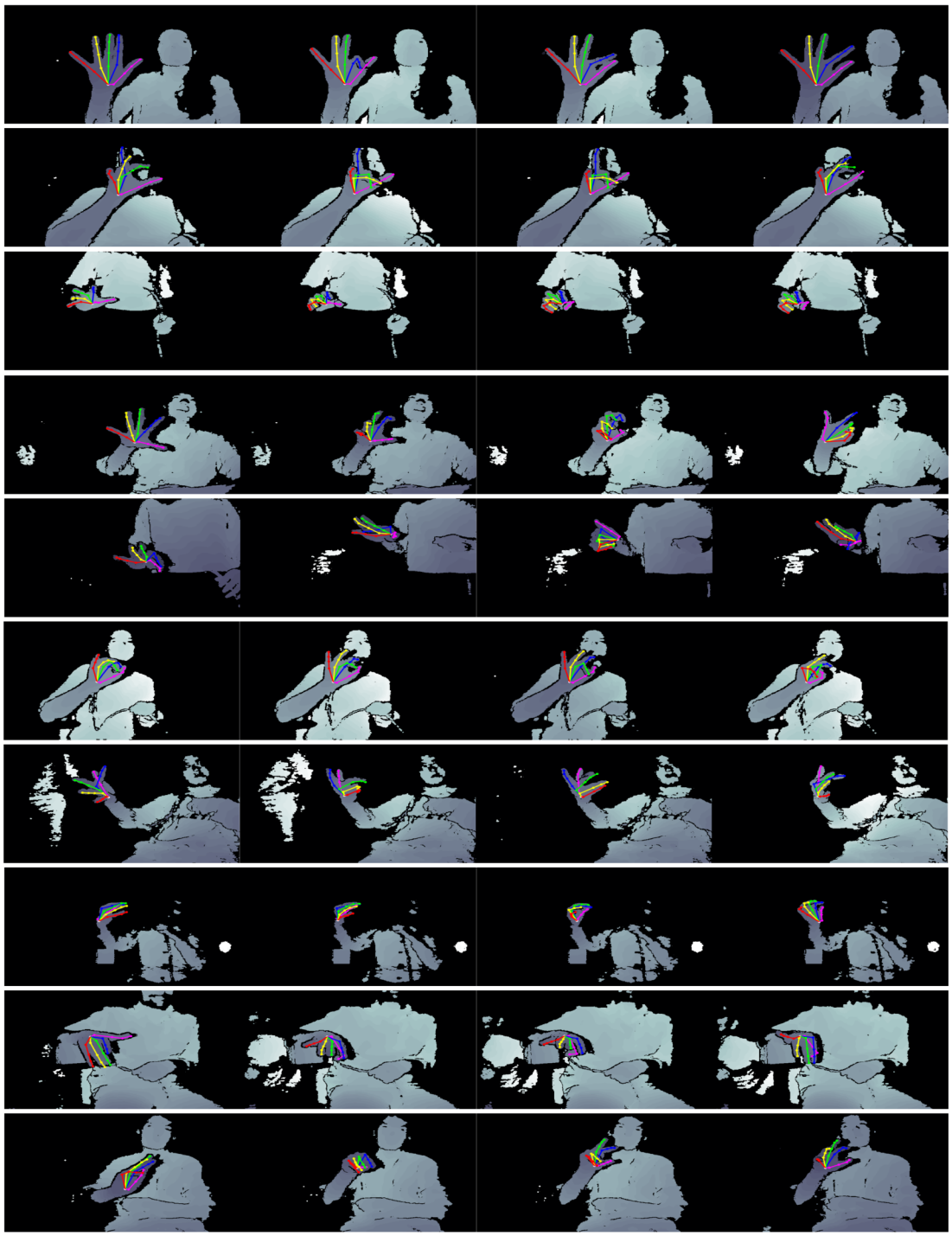


Figure A.9: *More example sequences for third-person viewpoint.*

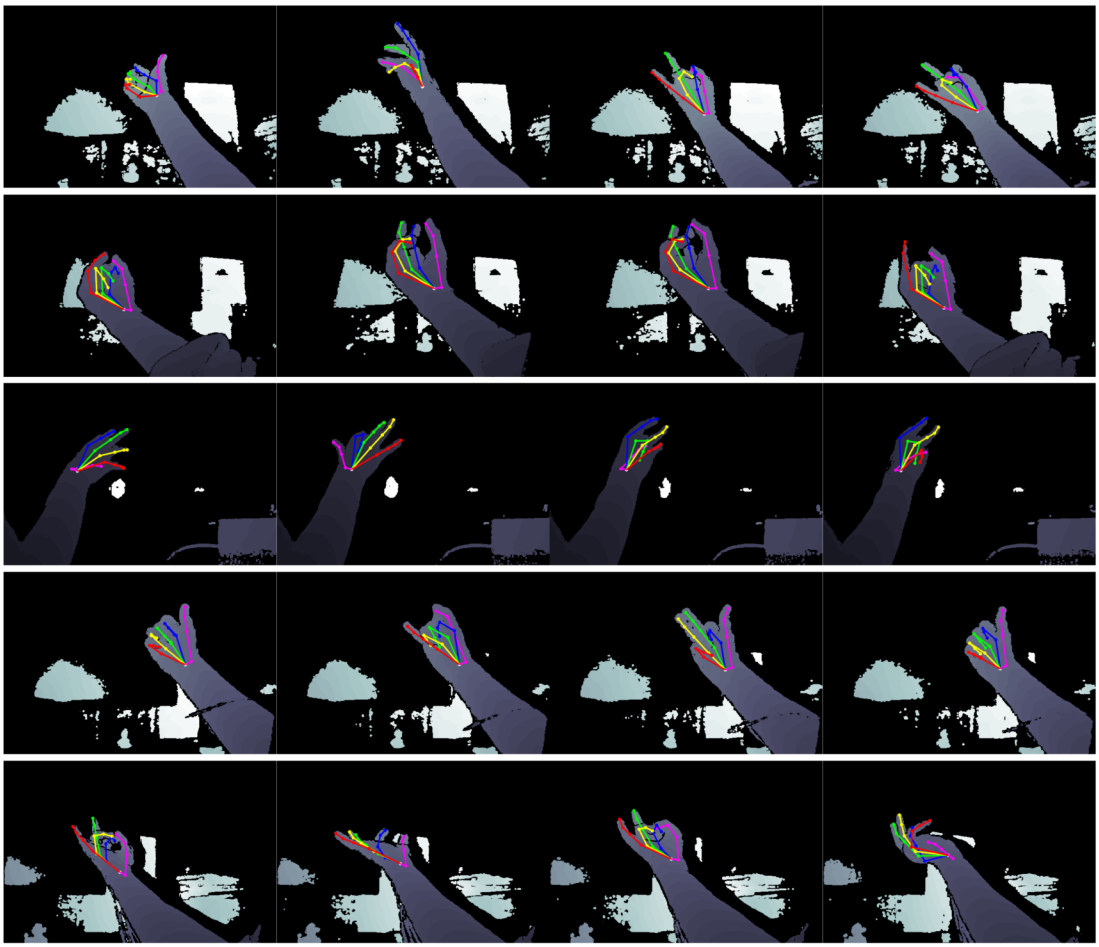


Figure A.10: ***Example sequences for egocentric view***. *The sequences are from the egocentric part of the BigHand2.2M [Yuan et al., 2017] dataset. The test sequences covers all the 32 extremal poses where each finger assumes a maximally bent or extended position.*

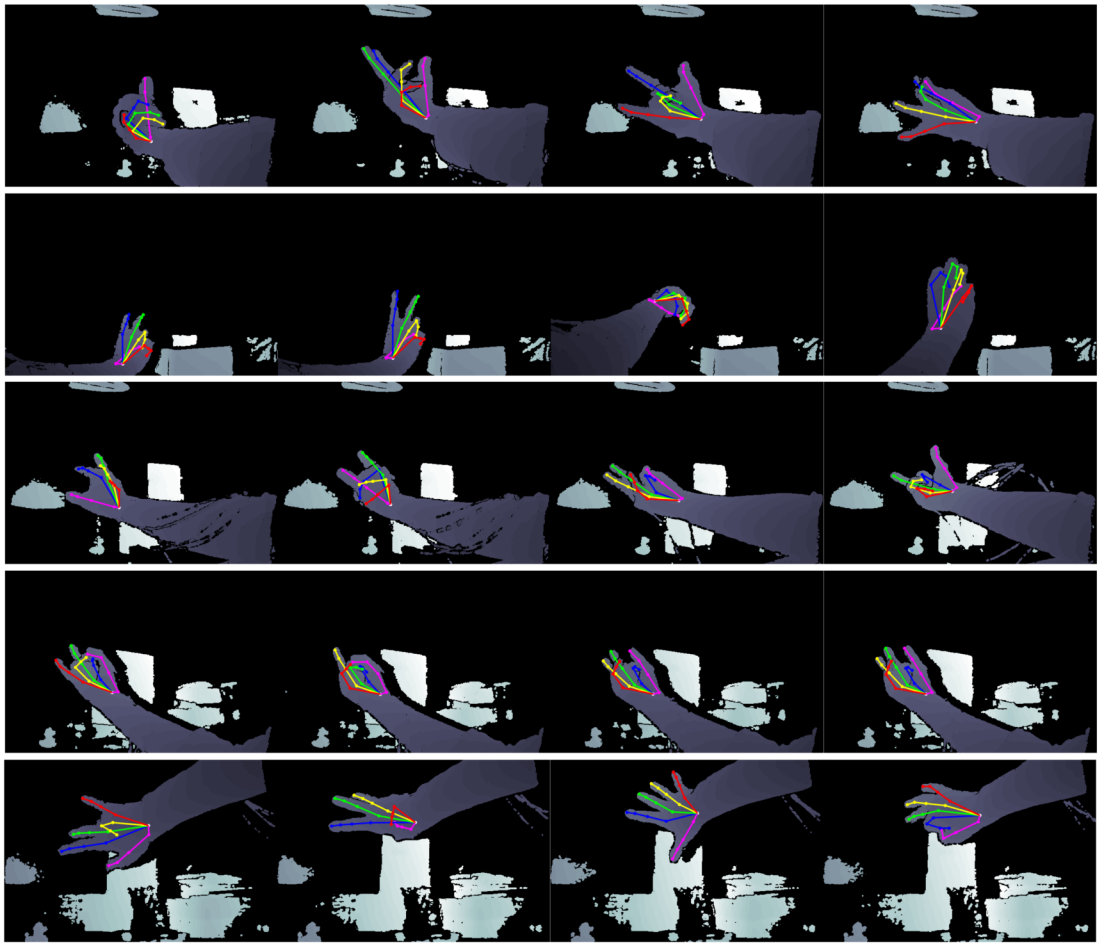


Figure A.11: *More example sequences for egocentric view.*

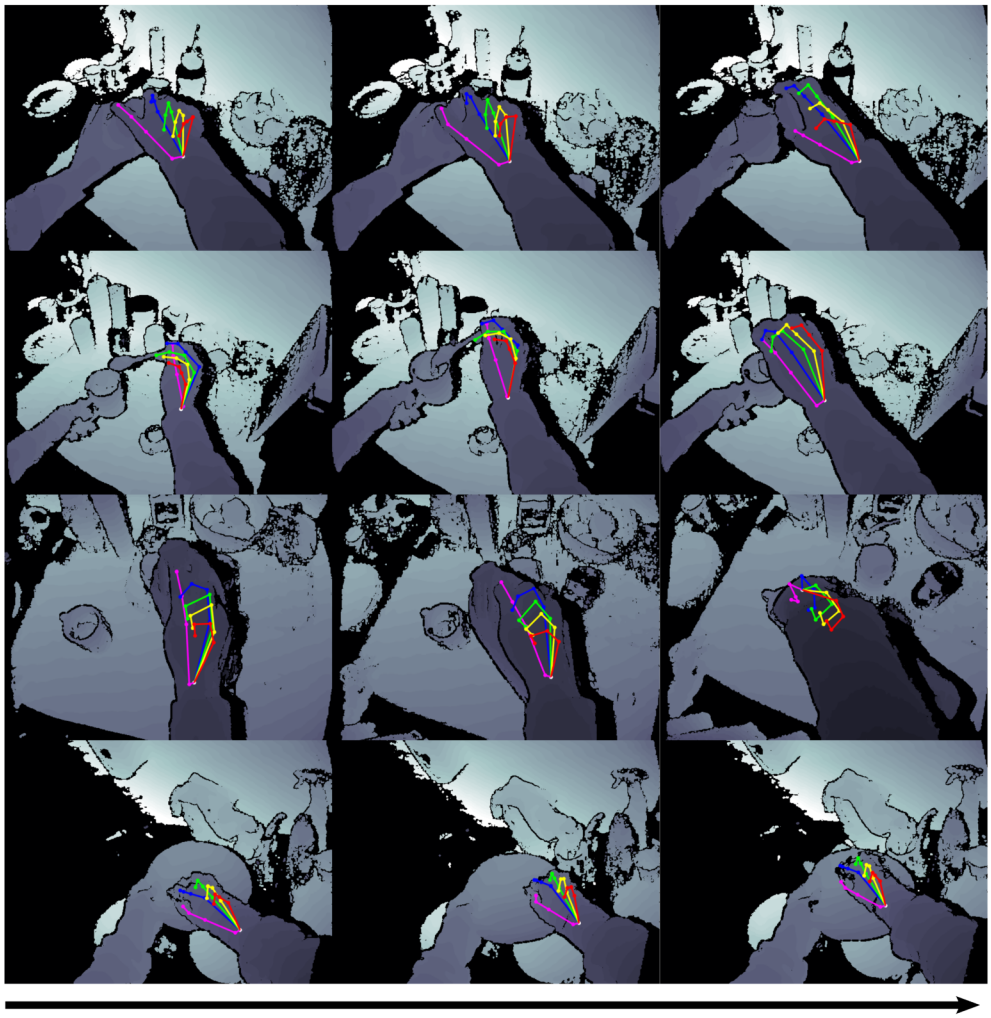


Figure A.12: ***Examples of sequences of hand actions [Garcia-Hernando et al.,]:*** *From top to bottom: 'open peanut butter', 'put sugar', 'pour milk' and 'wash with sponge' (all in kitchen). Figure courtesy of [Garcia-Hernando et al.,].*

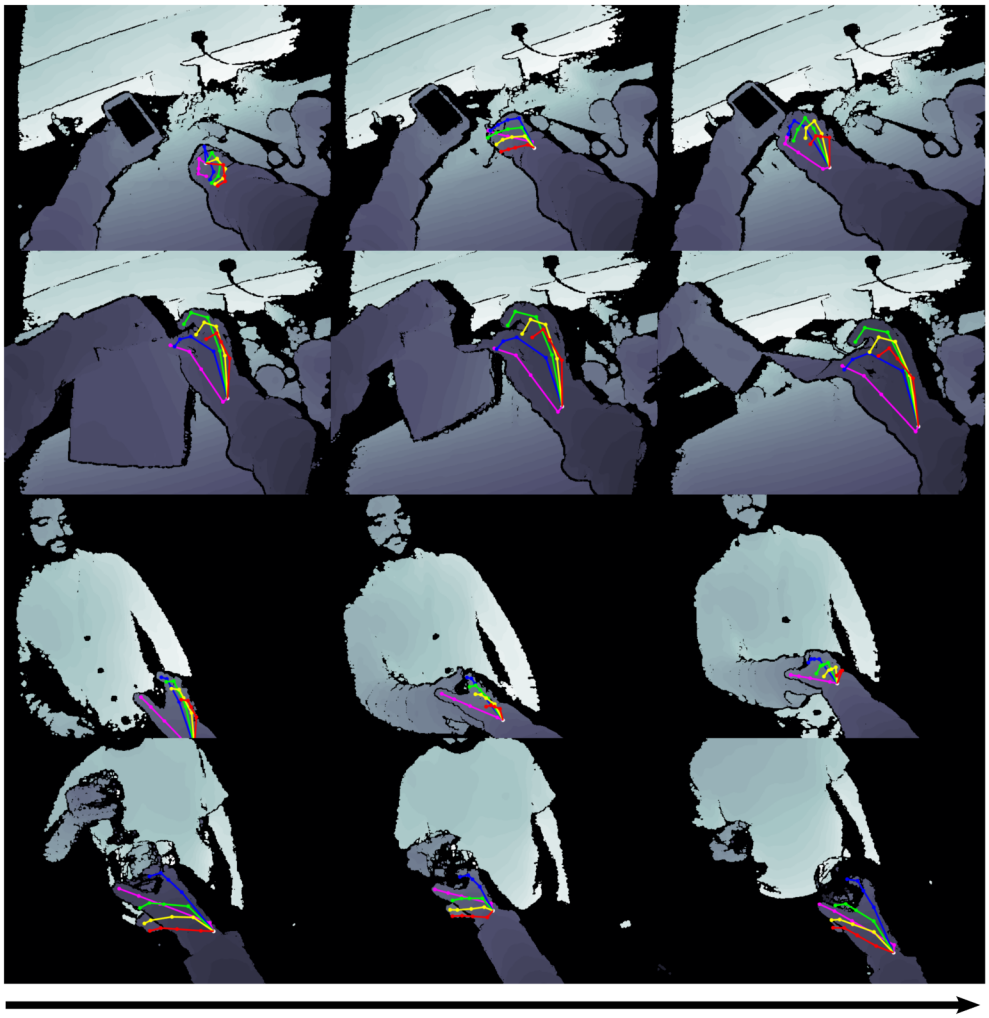


Figure A.13: ***More examples of sequences of hand actions [Garcia-Hernando et al.,]:*** *From top to bottom: 'charge cell phone' and 'tear paper' (office); 'handshake' and 'toast with wine glass' (social). Figure courtesy of [Garcia-Hernando et al.,].*

REFERENCES

[Agarwal and Triggs, 2006] Agarwal, A. and Triggs, B. (2006). Recovering 3D human pose from monocular images. *IEEE TPAMI*, 28(1):44–58. 13

[Akhter and Black, 2015] Akhter, I. and Black, M. J. (2015). Pose-conditioned joint angle limits for 3D human pose reconstruction. In *CVPR*. 13

[Akiyama et al., 2017] Akiyama, K., Yang, F., and Wu, Y. (Hands 2017). Naist rvlab g2's solution for 2017 hand challenge. xix, xx, xxi, 51, 52, 53, 55, 56, 60, 62, 64, 65, 66, 69, 70, 71, 76, 77, 79

[Ba and Caruana, 2014] Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *NIPS*. 86

[Baek et al.,] Baek, S., Kim, K. I., and Kim, T.-K. Augmented skeleton space transfer for depth-based hand pose estimation. In *CVPR, 2018*. 49

[Ballan et al.,] Ballan, L., Taneja, A., Gall, J., Van Gool, L., and Pollefeys, M. Motion capture of hands in action using discriminative salient points. In *ECCV, 2012*. 9

[Cai et al.,] Cai, Y., Ge, L., Cai, J., and Yuan, J. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *ECCV, 2018*. 87

[Chen et al.,] Chen, L., Li, W., and Xu, D. Recognizing rgb images by learning from rgb-d data. In *CVPR 2014*. 87

[Chen et al., 2017a] Chen, X., Wang, G., Guo, H., and Zhang, C. (2017a). Pose guided structured region ensemble network for cascaded hand pose estimation. *arXiv preprint*

arXiv:1708.03416. xix, xx, xxi, 51, 52, 53, 55, 56, 60, 62, 65, 66, 69, 70, 71, 72, 73, 74, 75, 76, 77

[Chen et al., 2017b] Chen, Y., Jin, X., Feng, J., and Yan, S. (2017b). Training group orthogonal neural networks with privileged information. *arXiv:1701.06772*. 83, 86

[Choi et al., a] Choi, C., Kim, S., and Ramani, K. Learning hand articulations by hallucinating heat distribution. In *ICCV, 2017*. 49

[Choi et al., 2015] Choi, C., Sinha, A., Hee Choi, J., Jang, S., and Ramani, K. (2015). A collaborative filtering approach to real-time hand pose estimation. In *ICCV*. 22, 85

[Choi et al., b] Choi, C., Yoon, S. H., Chen, C.-N., and Ramani, K. Robust hand pose estimation during the interaction with an unknown object. In *ICCV, 2017*. 49

[Cihan Camgoz et al.,] Cihan Camgoz, N., Hadfield, S., Koller, O., and Bowden, R. Subunets: End-to-end hand shape and continuous sign language recognition. In *ICCV, 2017*. 49

[Clerc and Kennedy, 2002] Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on*, 6(1):58–73. 5, 12

[de La Gorce et al.,] de La Gorce, M., Fleet, D. J., and Paragios, N. Model-based 3D hand pose estimation from monocular video. *IEEE Transactions on PAMI, 2011*. 9

[Deng et al., 2017] Deng, X., Yang, S., Zhang, Y., Tan, P., Chang, L., and Wang, H. (2017). Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*. 49

[Eigen et al., 2014] Eigen, D., Puhrsch, C., and Fergus, R. (2014). Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374. 109

[Ek et al., 2007] Ek, C. H., Torr, P. H., and Lawrence, N. D. (2007). Gaussian process latent variable models for human pose estimation. In *International workshop on machine learning for multimodal interaction*, pages 132–143. 13

[Erol et al., 2007] Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. (2007). Vision-based hand pose estimation: A review. *CVIU*, 108(1):52–73. 50, 85

[Everingham et al., 2015] Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136. 50, 112

[Fan et al., 2014] Fan, X., Zheng, K., Zhou, Y., and Wang, S. (2014). Pose locality constrained representation for 3D human pose reconstruction. In *ECCV*. Springer. 13

[Fu et al., 2018] Fu, H., Gong, M., Wang, C., Batmanghelich, K., and Tao, D. (2018). Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011. 109

[Garcia-Hernando et al.,] Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *CVPR, 2018*. xxv, 16, 17, 49, 51, 108, 111, 112, 114, 128, 129

[Garcia-Hernando et al., 2018] Garcia-Hernando, G., Yuan, S., Baek, S., and Kim, T.-K. (2018). First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. *CVPR*. 7, 82, 83, 85

[Ge et al., 2018] Ge, L., Cai, Y., Weng, J., and Yuan, J. (2018). Hand PointNet: 3d hand pose estimation using point sets. In *CVPR*. xix, xx, xxi, 51, 52, 53, 55, 60, 62, 66, 69, 70, 71

[Ge et al., 2016a] Ge, L., Liang, H., Yuan, J., and Thalmann, D. (2016a). Robust 3D hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *CVPR*. 7, 53

[Ge et al., 2016b] Ge, L., Liang, H., Yuan, J., and Thalmann, D. (2016b). Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs. In *CVPR*. 22, 108

[Ge et al., 2017] Ge, L., Liang, H., Yuan, J., and Thalmann, D. (2017). 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. CVPR. 7, 53, 82, 85, 108

[Ge and Yuan, 2017] Ge, L. and Yuan, J. (Hands 2017). 2017 hand challenge imi ntu team. xix, xx, xxi, 51, 52, 53, 55, 60, 62, 66, 69, 70, 71

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS*. 14, 107

[Goodfellow et al., 2013] Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. *ICML*. 99

[Guo et al., 2017] Guo, H., Wang, G., Chen, X., and Zhang, C. (2017). Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*. xix, 51, 52, 55, 56, 59, 64, 75

[Gupta et al.,] Gupta, S., Hoffman, J., and Malik, J. Cross modal distillation for supervision transfer. In *CVPR 2016*. 89

[Gustus et al., 2012] Gustus, A., Stillfried, G., Visser, J., Jörntell, H., and van der Smagt, P. (2012). Human hand modelling: kinematics, dynamics, applications. *Biological cybernetics*, 106(11-12):741–755. 8

[Hale and Stanney, 2002] Hale, K. S. and Stanney, K. M. (2002). *Handbook of virtual environments: Design, implementation, and applications*. CRC Press. 8

[Hamer et al., 2009] Hamer, H., Schindler, K., Koller-Meier, E., and Van Gool, L. (2009). Tracking a hand manipulating an object. In *Computer Vision, 2009 IEEE 12th International Conference On*, pages 1475–1482. IEEE. 12

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*. 52, 55, 56

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. In *arXiv:1503.02531*. 86, 88, 89

[Hoffman et al., 2016] Hoffman, J., Gupta, S., and Darrell, T. (2016). Learning with side information through modality hallucination. In *CVPR*. 83, 86, 87, 89

[Honari et al., a] Honari, S., Molchanov, P., Tyree, S., Vincent, P., Pal, C., and Kautz, J. Improving landmark localization with semi-supervised learning. In *CVPR, 2018*. 7, 52, 54, 56

[Honari et al., b] Honari, S., Yosinski, J., Vincent, P., and Pal, C. Recombinator networks: Learning coarse-to-fine feature aggregation. In *CVPR, 2016*. 7, 54

[Intel SR300,] Intel SR300. https://click.intel.com/intelrealsense-developer-kit-featuring-sr300.html. xviii, xxvi, 25, 32, 42, 44, 114

[Iqbal et al.,] Iqbal, U., Molchanov, P., Breuel, T., Gall, J., and Kautz, J. Hand pose estimation via latent 2.5 d heatmap regression. In *ECCV, 2018*. 87, 107

[Jang et al., 2015] Jang, Y., Noh, S.-T., Chang, H. J., Kim, T.-K., and Woo, W. (2015). 3D Finger CAPE: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *VR*. 22

[John et al.,] John, V., Trucco, E., and Ivekovic, S. Markerless human articulated tracking using hierarchical particle swarm optimisation. *IVC 2010*. 5

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *International Conference on Neural Networks*, volume 4, pages 1942–1948. 4

[Kennedy and Mendes, 2002] Kennedy, J. and Mendes, R. (2002). Population structure and particle swarm performance. 5

[Keskin et al.,] Keskin, C., Kıraç, F., Kara, Y. E., and Akarun, L. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *ECCV, 2012*. 6, 10, 22, 42

[Keskin et al., 2013] Keskin, C., Kıraç, F., Kara, Y. E., and Akarun, L. (2013). Real time hand pose estimation using depth sensors. In *Consumer Depth Cameras for Computer Vision*, pages 119–137. Springer. 6

[Khamis et al.,] Khamis, S., Taylor, J., Shotton, J., Keskin, C., Izadi, S., and Fitzgibbon, A. Learning an efficient model of hand shape variation from depth images. In *CVPR, 2015*. 22

[Krejov et al.,] Krejov, P., Gilbert, A., and Bowden, R. Combining discriminative and model based approaches for hand pose estimation. In *FG (2015)*. 11

[Kristan et al., 2015] Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernández, G., Vojir, T., Hager, G., Nebehay, G., and Pflugfelder, R. (2015). The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23. 50, 112

[Lee and Chen, 1985] Lee, H.-J. and Chen, Z. (1985). Determination of 3D human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168. 13

[Li et al.,] Li, P., Ling, H., Li, X., and Liao, C. 3d hand pose estimation using randomized decision forest with segmentation index points. In *ICCV, 2015*. 6, 22, 108, 112

[Li and Chan, 2014] Li, S. and Chan, A. B. (2014). 3D human pose estimation from monocular images with deep convolutional neural network. In *ACCV*. 13, 86

[Li and Lee, 2017] Li, S. and Lee, D. (Hands 2017). 2017 hand challenge/frame-based/team hcr: Method description. xix, xx, xxi, 51, 52, 53, 55, 56, 60, 62, 64, 66, 69, 70, 71

[Liang et al.,] Liang, H., Yuan, J., and Thalmann, D. Parsing the hand in depth images. *TMM, 2014*. 22

[Lu et al.,] Lu, S., Metaxas, D., Samaras, D., and Oliensis, J. Using multiple cues for hand tracking and model refinement. In *CVPR, 2003*. 9

[Luo et al.,] Luo, Z., Jiang, L., Hsieh, J.-T., Niebles, J. C., and Fei-Fei, L. Graph distillation for action detection with privileged information. In *ECCV, 2018*. 87

[MacCormick and Isard,] MacCormick, J. and Isard, M. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV, 2000*. 9

[Madadi et al., 2017] Madadi, M., Escalera, S., Baro, X., and Gonzalez, J. (2017). End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*. xix, xx, xxi, 7, 51, 52, 53, 55, 56, 60, 62, 63, 64, 66, 69, 70, 71

[Madadi et al.,] Madadi, M., Escalera, S., Carruesco, A., Andujar, C., Baró, X., and Gonzàlez, J. Occlusion aware hand pose recovery from sequences of depth images. In *FG, 2017*. 49

[Makris et al., 2015] Makris, A., Kyriazis, N., and Argyros, A. (2015). Hierarchical particle filtering for 3d hand tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 8–17. 9

[Melax et al.,] Melax, S., Keselman, L., and Orsten, S. Dynamics based 3D skeletal hand tracking. In *i3D, 2013*. 26, 42

[Mendes et al., 2002] Mendes, R., Cortez, P., Rocha, M., and Neves, J. (2002). Particle swarms for feedforward neural network training. *learning*, 6(1). 5

[Molchanov et al., 2017] Molchanov, P., Kautz, J., and Honari, S. (Hands 2017). 2017 hand challenge nvresearch and umontreal team. xix, xx, xxi, 7, 51, 52, 53, 54, 56, 60, 62, 66, 67, 69, 70, 71, 72, 73, 75, 76, 77, 79, 107

[Moon et al.,] Moon, G., Chang, J. Y., and Lee, K. M. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. In *CVPR, 2018*. xix, xx, xxi, 7, 51, 52, 53, 54, 55, 56, 60, 62, 66, 69, 70, 71

[Mori and Malik, 2006] Mori, G. and Malik, J. (2006). Recovering 3D human body configurations using shape contexts. *IEEE TPAMI*, 28(7):1052–1062. 13

[MPII Leader Board,] MPII Leader Board. http://human-pose.mpi-inf.mpg.de. 63

[Mueller et al., 2017a] Mueller, F., Bernard, F., Sotnychenko, O., Mehta, D., Sridhar, S., Casas, D., and Theobalt, C. (2017a). GANerated hands for real-time 3D hand tracking from monocular RGB. In *arXiv:1712.01057*. 14, 83, 85, 94, 99

[Mueller et al.,] Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *ICCV, 2017*. 49

[Mueller et al., 2017b] Mueller, F., Mehta, D., Sotnychenko, O., Sridhar, S., Casas, D., and Theobalt, C. (2017b). Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *ICCV*. 14, 83, 85, 88, 107

[NDI trakSTAR,] NDI trakSTAR. https://www.ascension-tech.com/products/trakstar-2-drivebay-2/. xvi, 28, 30, 32, 114

[Neverova et al.,] Neverova, N., Wolf, C., Taylor, G. W., and Nebout, F. Hand segmentation with structured convolutional learning. In *ACCV, 2014*. 22

[Niu et al., 2012] Niu, X., Terekhov, A. V., Latash, M. L., and Zatsiorsky, V. M. (2012). Reconstruction of the unknown optimization cost functions from experimental recordings during static multi-finger prehension. *Motor control*, 16(2):195–228. 15

[Oberweger and Lepetit,] Oberweger, M. and Lepetit, V. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV Workshop, 2017*. 55, 59

[Oberweger et al.,] Oberweger, M., Riegler, G., Wohlhart, P., and Lepetit, V. Efficiently creating 3D training data for fine hand pose estimation. In *CVPR, 2016*. 22, 25, 26, 27, 43, 44, 45, 49, 112

[Oberweger et al., 2015a] Oberweger, M., Wohlhart, P., and Lepetit, V. (2015a). Hands deep in deep learning for hand pose estimation. In *CVWW*. 22, 42, 51, 52, 55, 56, 59

[Oberweger et al., 2015b] Oberweger, M., Wohlhart, P., and Lepetit, V. (2015b). Training a feedback loop for hand pose estimation. In *ICCV*. 22, 42, 82, 85

[Oikonomidis et al., 2011a] Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011a). Efficient model-based 3D tracking of hand articulations using kinect. In *BMVC*. xviii, 2, 3, 5, 8, 9, 10, 22, 42, 43, 52, 57, 59, 94, 112, 120

[Oikonomidis et al., 2011b] Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2011b). Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*. 5, 9

[Oikonomidis et al., 2012] Oikonomidis, I., Kyriazis, N., and Argyros, A. A. (2012). Tracking the articulated motion of two strongly interacting hands. In *CVPR*. 5, 9

[Ong and Bowden,] Ong, E.-J. and Bowden, R. A boosted classifier tree for hand shape detection. In *IEEE International Conference on Automatic Face and Gesture Recognition, 2004*. 10

[Panteleris and Argyros, 2017] Panteleris, P. and Argyros, A. A. (2017). Back to RGB: 3D tracking of hands and hand-object interactions based on short-baseline stereo. In *ICCV Workshop*. 14, 83, 85, 86, 107

[Parameswaran and Chellappa, 2004] Parameswaran, V. and Chellappa, R. (2004). View independent human body pose estimation from a single perspective image. In *CVPR*. 13

[Park et al.,] Park, G., Argyros, A. A., and Woo, W. Efficient 3d hand tracking in articulation subspaces for the manipulation of virtual objects. In *CGI 2016*. 5

[Parkhi et al.,] Parkhi, O. M., Vedaldi, A., and Zisserman, A. Deep face recognition. In *BMVC, 2015*. 37

[Pham et al., 2015] Pham, T.-H., Kheddar, A., Qammaz, A., and Argyros, A. A. (2015). Towards force sensing from vision: Observing hand-object interactions to infer manipulation forces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2810–2819. 15

[Pham et al., 2017] Pham, T.-H., Kyriazis, N., Argyros, A. A., and Kheddar, A. (2017). Hand-object contact force estimation from markerless visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 15

[Poier et al.,] Poier, G., Roditakis, K., Schulter, S., Michel, D., Bischof, H., and Argyros, A. A. Hybrid one-shot 3D hand pose estimation by exploiting uncertainties. *BMVC (2015)*. 5

[Pons-Moll et al.,] Pons-Moll, G., Baak, A., Gall, J., Leal-Taixe, L., Mueller, M., Seidel, H.-P., and Rosenhahn, B. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *ICCV, 2011*. 28

[Qian et al., a] Qian, C., Sun, X., Wei, Y., Tang, X., and Sun, J. Realtime and robust hand tracking from depth. In *CVPR, 2014*. 5, 8, 9, 94

[Qian et al., b] Qian, C., Sun, X., Wei, Y., Tang, X., and Sun, J. Realtime and robust hand tracking from depth. In *CVPR, 2014*. 22, 25, 26, 27, 49, 112

[Ramakrishna et al., 2012] Ramakrishna, V., Kanade, T., and Sheikh, Y. (2012). Reconstructing 3D human pose from 2D image landmarks. In *ECCV*. 13

[Rehg and Kanade,] Rehg, J. M. and Kanade, T. Visual tracking of high dof articulated structures: an application to human hand tracking. In *ECCV, 1994*. 9

[Remelli et al.,] Remelli, E., Tkach, A., Tagliasacchi, A., and Pauly, M. Low-dimensionality calibration through local anisotropic scaling for robust hand model personalization. In *ICCV, 2017*. 49

[Ren et al.,] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS, 2015*. 73, 75

[Riegler et al.,] Riegler, G., Ferstl, D., Rüther, M., and Bischof, H. A framework for articulated hand pose estimation and evaluation. In *SCIA, 2015*. 29

[Rogez et al., 2012] Rogez, G., Rihan, J., Orrite-Uruñuela, C., and Torr, P. H. (2012). Fast human pose detection using randomized hierarchical cascades of rejectors. *International journal of computer vision*, 99(1):25–52. 6

[Rogez et al., 2014] Rogez, G., Supancic, J. S., Khademi, M., Montiel, J. M. M., and Ramanan, D. (2014). 3D hand pose detection in egocentric RGB-D images. *ECCV*. 6, 10, 13

[Rogez et al., 2015a] Rogez, G., Supancic, J. S., and Ramanan, D. (2015a). First-person pose recognition using egocentric workspaces. In *CVPR*. 25, 27, 28, 44, 45

[Rogez et al., 2015b] Rogez, G., Supancic, J. S., and Ramanan, D. (2015b). Understanding everyday hands in action from RGB-D images. In *ICCV*. 15, 29

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. 63, 73, 74

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252. 50, 112

[Schaffelhofer and Scherberger, 2012] Schaffelhofer, S. and Scherberger, H. (2012). A new method of accurate hand- and arm-tracking for small primates. *Journal of Neural Engineering*, 9(2). 31

[Serra, 1982] Serra, J. (1982). *Image analysis and mathematical morphology, v. 1*. Academic press. 76, 77

[ShapeHand, 2009] ShapeHand (2009). `http://www.shapehand.com/shapehand.html`. 28

[Sharifi et al.,] Sharifi, A., Harati, A., and Vahedian, A. Marker based human pose estimation using annealed particle swarm optimization with search space partitioning. In *ICCKE 2014*. 5

[Sharp et al., 2015] Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S. (2015). Accurate, robust, and flexible real-time hand tracking. In *CHI*. xix, xxi, 3, 11, 22, 25, 29, 35, 37, 49, 57, 59, 75, 76, 78, 85, 112, 120

[Shi and Eberhart,] Shi, Y. and Eberhart, R. A modified particle swarm optimizer. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*. 5

[Shi and Kim, 2017] Shi, Z. and Kim, T.-K. (2017). Learning and refining of privileged information-based RNNs for action recognition from depth sequences. In *CVPR*. 83, 86

[Shotton et al., 2013] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124. 6

[Shrivastava et al., 2017] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. (2017). Learning from simulated and unsupervised images through adversarial training. In *CVPR*. 14, 107

[Simo-Serra et al., 2012] Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C., and Moreno-Noguer, F. (2012). Single image 3D human pose estimation from noisy observations. In *CVPR*. 13, 86

[Simon et al.,] Simon, T., Joo, H., Matthews, I., and Sheikh, Y. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR, 2017*. 49

[Simon et al., 2017] Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*. 14, 83, 85, 86, 102, 107, 112

[Sinha et al.,] Sinha, A., Choi, C., and Ramani, K. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *CVPR, 2016*. 55

[Slota et al., 2011] Slota, G. P., Latash, M. L., and Zatsiorsky, V. M. (2011). Grip forces during object manipulation: experiment, mathematical model, and validation. *Experimental brain research*, 213(1):125–139. 15

[Song and Xiao,] Song, S. and Xiao, J. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR, 2016*. 7, 53

[Spurr et al.,] Spurr, A., Song, J., Park, S., and Hilliges, O. Cross-modal deep variational hand pose estimation. In *CVPR, 2018*. 87, 107

[Sridhar et al., 2016] Sridhar, S., Mueller, F., Zollhöfer, M., Casas, D., Oulasvirta, A., and Theobalt, C. (2016). Real-time joint tracking of a hand manipulating an object from rgb-d input. In *ECCV*, pages 294–310. Springer. 85, 91, 92, 102

[Sridhar et al.,] Sridhar, S., Oulasvirta, A., and Theobalt, C. Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV, 2013*. 22, 25, 26, 49

[Sridhar et al., 2013] Sridhar, S., Oulasvirta, A., and Theobalt, C. (2013). Interactive markerless articulated hand motion tracking using rgb and depth data. In *ICCV*. 8, 11

[Stenger et al.,] Stenger, B., Mendonça, P. R., and Cipolla, R. Model-based 3D tracking of an articulated hand. In *CVPR, 2001*. 9

[Stenger et al., 2006] Stenger, B., Thayananthan, A., Torr, P. H., and Cipolla, R. (2006). Model-based hand tracking using a hierarchical bayesian filter. *IEEE Transactions on PAMI*. 9

[Sturman and Zeltzer, 1994] Sturman, D. J. and Zeltzer, D. (1994). A survey of glove-based input. *Computer Graphics and Applications, IEEE*, 14(1):30–39. 8

[Sudderth et al.,] Sudderth, E. B., Mandel, M. I., Freeman, W. T., and Willsky, A. S. Visual hand tracking using nonparametric belief propagation. In *CVPR Workshop, 2004*. 9

[Sueda et al.,] Sueda, S., Kaufman, A., and Pai, D. K. Musculotendon simulation for hand animation. In *ACM Transactions on Graphics (TOG), 2008*, volume 27, page 83. 8

[Sun et al., 2017] Sun, X., Shang, J., Liang, S., and Wei, Y. (2017). Compositional human pose regression. In *ICCV*. 51, 52, 56, 79

[Sun et al., 2015] Sun, X., Wei, Y., Liang, S., Tang, X., and Sun, J. (2015). Cascaded hand pose regression. In *CVPR*. 2, 6, 7, 22, 25, 26, 27, 41, 49, 82, 83, 85

[Supancic et al., 2015] Supancic, J. S., Rogez, G., Yang, Y., Shotton, J., and Ramanan, D. (2015). Depth-based hand pose estimation: methods, data, and challenges. *ICCV*. 24, 37, 38, 49, 50, 85

[Tagliasacchi et al., 2015] Tagliasacchi, A., Schröder, M., Tkach, A., Bouaziz, S., Botsch, M., and Pauly, M. (2015). Robust articulated-icp for real-time hand tracking. *Computer Graphics Forum*. 49

[Tang et al., 2014] Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2014). Latent regression forest: Structured estimation of 3D articulated hand posture. In *CVPR*. 2, 6, 7, 10, 22, 25, 26, 35, 42, 49, 82, 83, 85, 108, 112

[Tang et al., 2017] Tang, D., Chang, H. J., Tejani, A., and Kim, T.-K. (2017). Latent regression forest: structured estimation of 3d hand poses. *TPAMI*. 49

[Tang et al., 2015] Tang, D., Taylor, J., Kohli, P., Keskin, C., Kim, T.-K., and Shotton, J. (2015). Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV*. 6, 37, 41, 42, 49, 85, 108, 112

[Tang et al., 2013] Tang, D., Yu, T.-H., and Kim, T.-K. (2013). Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*. 10, 22, 49, 108, 112

[Taylor, 2000] Taylor, C. J. (2000). Reconstruction of articulated objects from point correspondences in a single uncalibrated image. *CVIU*, 80(3). 13

[Taylor et al., 2016] Taylor, J., Bordeaux, L., Cashman, T., Corish, B., Keskin, C., Soto, E., Sweeney, D., Valentin, J., Luff, B., Topalian, A., Wood, E., Khamis, S., Kohli, P., Sharp, T.,

Izadi, S., Banks, R., Fitzgibbon, A., and Shotton, J. (2016). Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *TOG*. 50

[Taylor et al., 2012] Taylor, J., Shotton, J., Sharp, T., and Fitzgibbon, A. (2012). The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE. xix, xxi, 3, 4, 52, 57, 59, 61, 75, 76, 78, 120

[Tome et al., 2017] Tome, D., Russell, C., and Agapito, L. (2017). Lifting from the deep: Convolutional 3D pose estimation from a single image. *CVPR*. 13, 86

[Tompson et al., 2014a] Tompson, J., Stein, M., Lecun, Y., and Perlin, K. (2014a). Real-time continuous pose recovery of human hands using convolutional networks. In *TOG*. 6, 7, 22, 25, 26, 35, 49, 82, 83, 85

[Tompson et al., 2014b] Tompson, J. J., Jain, A., LeCun, Y., and Bregler, C. (2014b). Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*. 13, 86

[Toshev and Szegedy, 2014] Toshev, A. and Szegedy, C. (2014). DeepPose: Human pose estimation via deep neural networks. In *CVPR*. 13, 86

[Vapnik and Vashist, 2009] Vapnik, V. and Vashist, A. (2009). A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5):544–557. 83, 86

[von Marcard et al.,] von Marcard, T., Pons-Moll, G., and Rosenhahn, B. Human pose estimation from video and imus. *TPAMI, 2016*. 28

[Wan et al.,] Wan, C., Probst, T., Van Gool, L., and Yao, A. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *CVPR, 2017*. 49

[Wan et al., 2017] Wan, C., Probst, T., Van Gool, L., and Yao, A. (2017). Crossing nets: Combining GANs and VAEs with a shared latent space for hand pose estimation. In *CVPR*. 7, 82, 85, 108, 112

[Wan et al., 2016] Wan, C., Yao, A., and Van Gool, L. (2016). Hand pose estimation from local surface normals. In *ECCV*. 6, 22, 49, 85, 108, 112

[Wan, 2017] Wan, Q. (Hands 2017). 2017 hand challenge fudan university team. xix, xx, xxi, 52, 53, 55, 56, 60, 62, 64, 66, 69, 70, 71

[Wang and Popović, 2009] Wang, R. Y. and Popović, J. (2009). Real-time hand-tracking with a color glove. In *ACM Transactions on Graphics (TOG)*, volume 28, page 63. 8, 10

[Wang and Ji, 2015] Wang, Z. and Ji, Q. (2015). Classifier learning with hidden information. In *CVPR*. 83

[Wei et al., 2016] Wei, S.-E., Ramakrishna, V., Kanade, T., and Sheikh, Y. (2016). Convolutional pose machines. In *CVPR*. 14, 86, 89, 92, 99

[Wetzler et al.,] Wetzler, A., Slossberg, R., and Kimmel, R. Rule of thumb: Deep derotation for improved fingertip detection. In *BMVC, 2015*. 22, 23, 25, 28, 33, 37

[Wu et al.,] Wu, Y., Lin, J. Y., and Huang, T. S. Capturing natural hand articulation. In *ICCV, 2001*. 33

[Xu et al., 2015] Xu, C., Ashwin, N., Zhang, X., and Cheng, L. (2015). Estimate hand poses efficiently from single depth images. *IJCV*. 23, 25, 28

[Xu and Cheng, 2013] Xu, C. and Cheng, L. (2013). Efficient hand pose estimation from a single depth image. In *ICCV*. 29

[Xu et al., 2018] Xu, D., Ricci, E., Ouyang, W., Wang, X., and Sebe, N. (2018). Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *IEEE transactions on pattern analysis and machine intelligence*. 109

[Xu et al., 2013] Xu, M., Jin, R., and Zhou, Z.-H. (2013). Speedup matrix completion with side information: Application to multi-label learning. In *NIPS*, pages 2301–2309. 83

[Yang et al., 2017] Yang, F., Akiyama, K., and Wu, Y. (Hands 2017). Naist rv's solution for 2017 hand challenge. xix, xx, xxi, 51, 52, 53, 54, 55, 60, 62, 63, 64, 66, 69, 70, 71, 72, 73, 74, 75, 76, 77

[Ye et al., 2016] Ye, Q., Yuan, S., and Kim, T.-K. (2016). Spatial attention deep net with partial PSO for hierarchical hybrid hand pose estimation. In *ECCV*. 7, 22, 36, 37, 42, 49, 52, 82, 85, 108, 112

[Yuan et al., 2018] Yuan, S., Garcia-Hernando, G., Stenger, B., Moon, G., Chang, J. Y., Lee, K. M., Molchanov, P., Kautz, J., Honari, S., Ge, L., et al. (2018). 3D hand pose estimation: From current achievements to future goals. In *CVPR*. 7, 18, 85

[Yuan et al.,] Yuan, S., Ye, Q., Garcia-Hernando, G., and Kim, T.-K. The 2017 hands in the million challenge on 3d hand pose estimation. In *arXiv preprint arXiv:1707.02237, 2017*. 16, 17, 19, 49, 51, 53, 79, 108

[Yuan et al., 2017] Yuan, S., Ye, Q., Stenger, B., Jain, S., and Kim, T.-K. (2017). BigHand2.2M benchmark: Hand pose dataset and state of the art analysis. *CVPR*. xxiv, 2, 7, 15, 17, 49, 51, 82, 83, 85, 88, 89, 93, 108, 111, 112, 114, 122, 124, 126

[Zhang et al., 2016] Zhang, J., Jiao, J., Chen, M., Qu, L., Xu, X., and Yang, Q. (2016). 3D hand pose tracking and estimation using stereo matching. In *arXiv preprint arXiv:1610.07214*. xxiii, 2, 49, 83, 85, 90, 91, 92, 94, 102

[Zheng et al.,] Zheng, Y., Sugimoto, S., and Okutomi, M. ASPnP: An accurate and scalable solution to the perspective-n-point problem. *IEICE TIS, 2013*. 33

[Zhou et al., 2016a] Zhou, X., Sun, X., Zhang, W., Liang, S., and Wei, Y. (2016a). Deep kinematic pose regression. In *ECCV*. 13, 86

[Zhou et al.,] Zhou, X., Wan, Q., Zhang, W., Xue, X., and Wei, Y. Model-based deep hand pose estimation. In *IJCAI, 2016*. 7, 22, 51, 52, 55, 56, 59

[Zhou et al., 2016b] Zhou, X., Zhu, M., Leonardos, S., Derpanis, K. G., and Daniilidis, K. (2016b). Sparseness meets deepness: 3D human pose estimation from monocular video. In *CVPR*. 13, 86

[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*. 83, 85

[Zhu et al., 2014] Zhu, X., Jia, X., and Wong, K.-Y. K. (2014). Pixel-level hand detection with shape-aware structured forests. In *Computer Vision–ACCV 2014*, pages 64–78. Springer. 6

[Zimmermann and Brox, 2017] Zimmermann, C. and Brox, T. (2017). Learning to estimate 3D hand pose from single RGB images. In *ICCV*. 14, 49, 83, 85, 86, 90, 91, 92, 94, 99, 102, 103, 107