NAIST-IS-MT1651133

Master's Thesis

A Deep-learning-based 3D Hand Pose Tracking System

Fan Yang

February 23, 2018

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Master of ENGINEERING

Fan Yang

Thesis Committee:

Professor Kazushi Ikeda	(Supervisor)
Professor Shoji Kasahara	(Co-supervisor)
Associate Professor Takatomi Kubo	(Co-supervisor)
Assistant Professor Yang Wu	(Co-supervisor)

A Deep-learning-based 3D Hand Pose Tracking System*

Fan Yang

Abstract

Although the deep-learning-based hand pose estimation has been popular for quite a while, most of the existing works solely focus on the pose estimation model, while simply suppose the input, which is the depth image of the hand part, is given or can be directly acquired by a depth threshold. In a realistic situation, however, the complex foreground and background of the hand area may exist, and aforementioned methods may not be applicable. Hence, the goal of this work is to develop a deep-learning-based 3D hand pose tracking system, which can efficiently and robustly detect the hand from the raw depth image before estimating the 3D hand pose. It mainly includes three parts, as the hand detector, the hand verifier and the pose estimator. The hand detector generates a mask to segment the hand area from the raw depth image. If the hand verifier confirm the segmented hand is correct, the pose estimator generates corresponding 3D hand pose using the depth image covered by the mask. We evaluated our system on the tracking task of *Hands In the Million* (HIM2017) challenge and placed second. In addition, we also applied our modified tracking system on the object-interactive task of *Hands In the Million* (HIM2017) challenge and placed first. We find that using hand detector to segment hand from its interactive objects before performing pose estimation can make better results than directly performing pose estimation.

Keywords:

3D hand pose estimation, depth video tracking

*Master's Thesis, Graduate School of Information Science,
Nara Institute of Science and Technology, NAIST-IS-MT1651133, February 23, 2018.

Contents

List of Figures

1 Introduction

1.1 Research motivation

Hands are essential for human beings to interact with surrounding environment. From the Human–computer interaction perspective, hand related applications, such as visual impaired people assistant, robotics control and Virtual Environments or Augmented Reality systems, are growing rapidly. To make algorithms that can recognize and respond to the hand command, using the 3D hand pose is an efficient and accurate approach, as the 3D hand pose can well-represent geometric information of the hand with less redundancy.

The techniques to acquire the 3D hand poses are broadly divided into two-types: utilizing electro-mechanical or magnetic sensing gloves, and vision-based methods. However, vision-based methods have become the dominant trend in applications as it does not need complex calibration and can be adapted to different applications conveniently [2].

Within vision-based 3D hand pose estimation approaches, the deep-learning-based hand pose estimation has been popular for quite a while. Nonetheless, most of the existing works solely focus on the pose estimation model, while simply suppose the input, which is the depth image of the hand part, is given or can be directly acquired by a depth threshold. In this work, we try to focus on a more realistic situation, where the hand part needs to be detected from the raw depth image when the complex foreground and background exist. Furthermore, we also treat the hand interactive objects as a kind of foreground or background. In such manner, the hand detector could also be exploited.

1.2 Research contribution

We introduce our 3D hand pose tracking system in this work, and hope some insights can contribute to the active ongoing research. Our main research contributions include:

- Proposed a hand detector based on the U-net structure, which directly uses depth image as input and output the corresponding hand mask;
- Proposed a 3D hand tracking system, whose performance is at the second place in hand pose tracking task of the *Hands In the Million* (HIM2017) challenge;
- With the state-of-art result in the hand-object interactive pose estimation task of the *Hands In the Million* (HIM2017) challenge, we suggest that, before performing pose estimation, using hand detector to segment hand from its interactive objects, can reach better performance than directly performing pose estimation.

1.3 Thesis overview

There are five chapters in this thesis. Chapter 2 introduces related works. Chapter 3 describes the detail of our 3D hand pose tracking system, including the model architecture and the training processes. Chapter 4 demonstrates the evaluation performance of the tracking system. Chapter 5 describes how we adapted the tracking system to the hand-object interactive pose estimation. In the end, Chapter 6 summarizes this work and introduce the feature work.

2 Related Works

Since our work contains two components, as the 3D hand pose estimator and the hand detector, we make a simple review for both of their related works. Here, we would like to appreciate X.H.Chen, who is organizing all kinds of hand pose estimation works and sharing them on his github [3], which considerably helps our literature review.

2.1 Related works of depth-based 3D hand pose estimator

A 3D hand pose estimator is used for generating 3D hand joint position, by giving the hand depth image.

2.1.1 Problems overview for 3D hand pose estimator

The main challenges of depth-image-based 3D hand pose estimation ([2, 4–6]) could be summarized as:

- The highly flexible hand pose;
- The self-occlusion or interactive-object occlusion;
- Depth camera noise;
- The diversity of depth image for the same pose, due to the difference of viewpoint. Hence, the training set could be computationally large but statistically small.

Thus, it remains an open problem to improve the accuracy and efficiency of estimate 3D hand pose estimation based on depth image. Our model design aims to alleviates the side effects from above challenges.

2.1.2 Generative-model-based approaches

By modeling physical properties of the hand, a simulated 3D hand model can be used to approximate the real hand. With this approach, to estimate the 3D hand pose is to optimize an objective function which models the difference between pre-defined hand model and observed depth image (see Fig.2.1).

In previous studies, optimization algorithms such as iterative closest point (ICP) [7], particle swarm optimization (PSO) [8] and ICP-PSO [9] were used. The main advantage of generative-model-based approach is to track 3D interacting hands pose [10], which is not easy for machine-learning-based approaches. Nonetheless, generative-model-based approaches also hold some drawbacks: the quality of depth image remarkably affect its estimation performance and the initialization is slow.

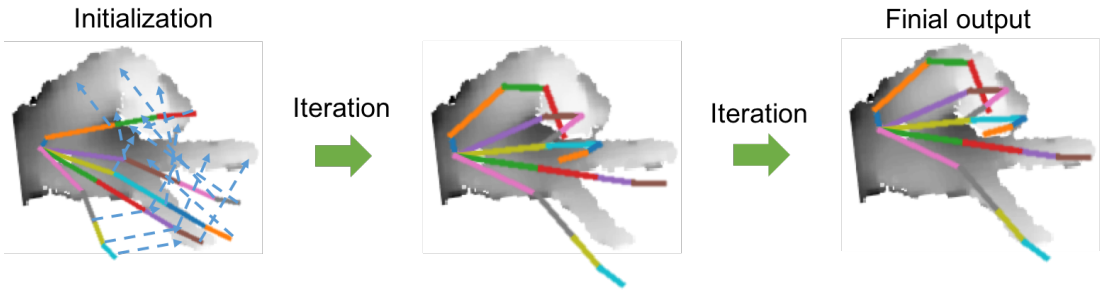

Figure 2.1: A Generative-model-based hand tracking system.

2.1.3 Machine-leaning-based approaches

As more and more research groups start to work on depth-based 3D hand pose estimation, the number of available datasets are growing ([11–16]). Meanwhile, the size of dataset is also increased significantly, which promotes the development of machine leaning based approaches.

In the earlier works, hand-crafted features and assemble models are utilized to estimate 3D hand pose [17–19]. These methods generally can run fast even on CPUs. Nevertheless, using CNN to automatically extract features from depth image has reached better performance. Therefore, recently, it becomes popular to use CNN as front layers to construct a deep-learning model. Many of such kind of models are able to generate very accurate results [20–26]. A summarization of for their structures could be demonstrated in Fig.2.2

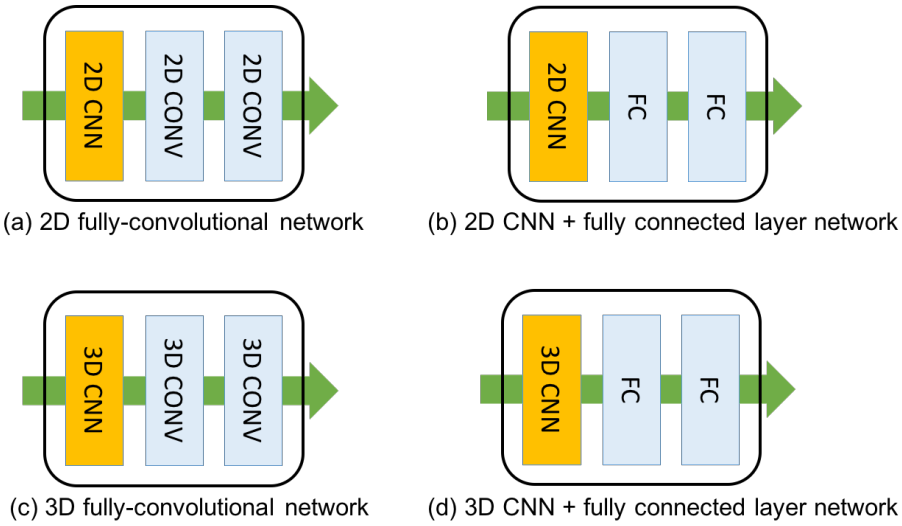

Figure 2.2: The categories of deep-learning based 3D hand pose estimator.

The design of our 3D hand estimator follows the strategy from previous works [20, 21], its structure could be denoted as Fig.2.2 (d). It takes a 3D depth image as input and outputs numerical 3D coordinates of all joints.

2.2 Related works of hand detector

A hand detector is used for segmenting the hand out from the raw depth image, its output is a binary hand mask.

2.2.1 Hand detectors using RGB image

Although segmenting the hand by RGB image [27,28], or RGB-D image [16] is the main approach, we argue that a hand detector trained by RGB image is sensitive to the texture and illumination. Besides, a heavy CNN network is needed to extract features from the RGB image.

2.2.2 Hand detectors only using depth image

To address these problems, we consider that segmenting hand only using the depth image is pertinent. In the simplest approach, this can be done by setting a depth threshold to separate the hand from the background [23, 29], but it is not suitable when other objects are in front of the hand. From a practical perspective, we aim to detect hand area when the complex foreground and background exist.

Mainly inspired by the original Kinect body segmentation work [30], hand-crafted features and random forest have been applied in hand detection [12]. Such kind of models can run very fast in testing, but has a large number of parameters. To keep a minimum size of hand detector, we suggest to utilize a fully convolutional network (FCN) [31]. The input is the depth image and the output is a mask corresponding to the hand area. More specifically, within several kinds of FCN structures, U-net [32] has a simple structure but outstanding performance. For this reason, we proposed our hand detector base on U-net structure.

2.3 Evaluation benchmark

It is natural to use the average error for all joints (or each joint) as the hand pose evaluation metrics. In order to visualize the proportion of testing samples whose error falls below a certain threshold, the success rate is also widely used [3].

Using precedent metrics, 3D hand pose estimation models were commonly evaluated on three datasets: ICVL dataset [11], NYU dataset [12] and MSRA dataset [13]. These three datasets has greatly promoted the machine-learning-based 3D hand pose estimation development, however, at the same time, they suffer problems of either have unrealistic synthetic hand depth image or high bias in pose annotation, due to the difficulty of annotating 3D hand pose. In addition, hand pose estimation from the first-person view is important, but it is not included in these datasets.

In 2017, the *Hands In the Million* (HIM2017) challenge [1] was launched to train and evaluate the 3D hand pose estimation performance. In HIM2017 challenge, samples from *BigHand2.2M* [15] and *First-Person Hand Action dataset* [14] are used for making a large-scale benchmark (see Table.2.1).

	Training	Single frame test	Tracking test	Interaction test
No. of samples	957K	295K	294K	2965

Table 2.1: The description of the HIM2017 benchmark

As it is the largest hand depth image dataset, and comes up with accurate 3D pose annotation, many research groups who proposed aforementioned studies start to evaluate their models on it. Moreover, it is the only platform that offers depth-based 3D hand tracking evaluation, thus, we use it to evaluate our 3D hand tracking system.

3 A Deep-learning-based 3D Hand Pose Tracking System

3.1 Overview of the proposed system

We developed a deep-learning-based 3D hand pose tracking system as Fig.3.1 shows. It mainly includes three parts, as the hand detector, the hand verifier and the pose estimator. The hand detector generates a mask to segment the hand area from the raw depth image, then the hand verifier will check whether the hand is correctly segmented. If the hand is successfully segmented, the pose estimator will generate corresponding 3D hand pose using the depth image covered by the mask, while in the failure case, the 3D hand pose from the previous frame will be used for the current frame. Other hand pose estimators can be easily integrated into our system.

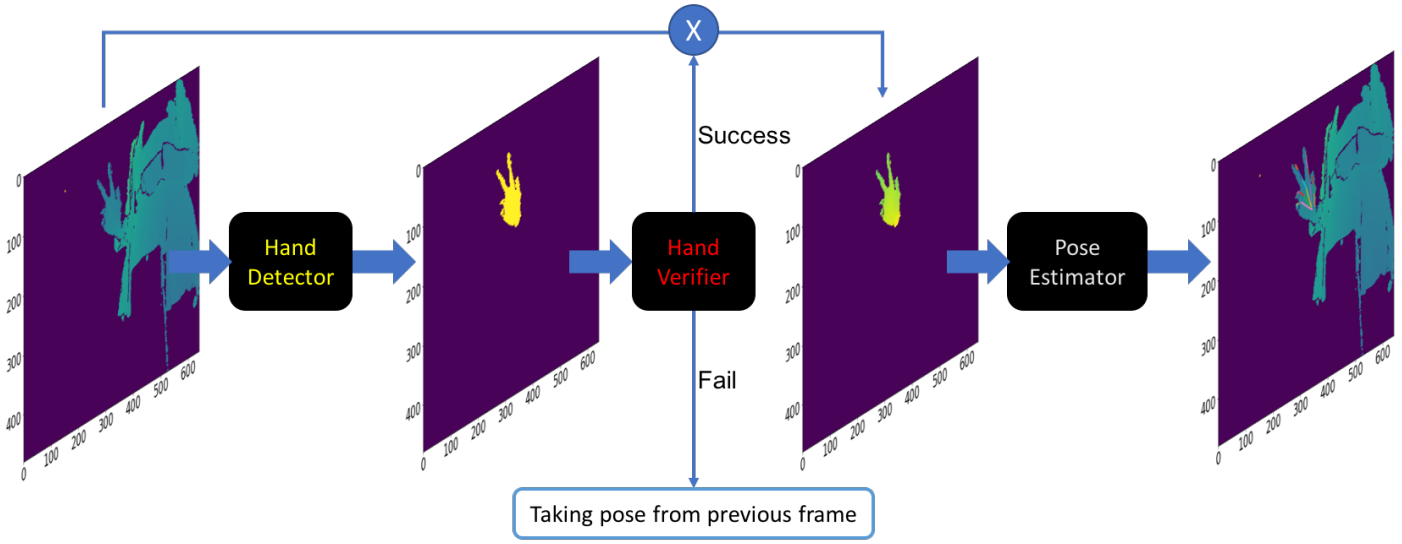

Figure 3.1: The structure of our proposed 3D hand pose tracking system.

3.2 Pose estimator

3.2.1 The architecture of pose estimator

Extending from previous studies ([20,21]), we developed our hand pose estimator (see Fig.3.3). The input is a 3D grid volume of size $50 \times 50 \times 50$. Referenced to the nearest point of the preprocessed hand area, thickened cloud points of the hand are fitted into the grid volume. Here, thickened cloud points means we give decreased values to unseen grids behind the 2.5D visible hand surface, with a threshold of three grids (see Fig.3.2). It is actually a kind of simplified Truncated Signed Distance Function (TSDF).

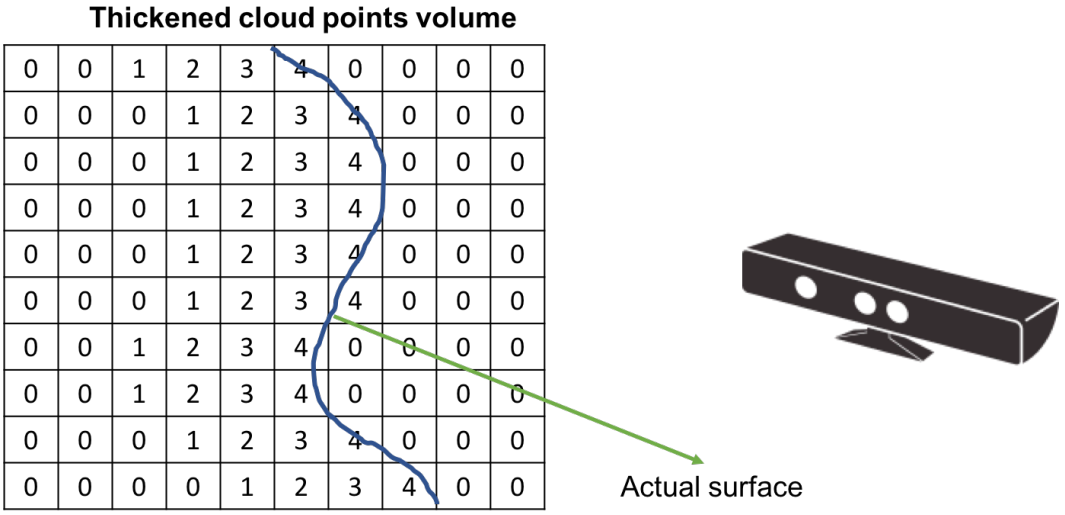

Figure 3.2: An illustration of proposed thickened cloud points.

Following the input, a hierarchical branch structure is used. It includes four kinds of convolutional blocks (see detail in Fig.3.4), and the 3D convolutional kennel is applied. Batch normalization and Rectified Linear units (ReLU) activation are associated with every layer, except for the final output layer, where no activation function is applied.

During the training stage, there are six outputs involved. Five of them are with respect to five fingers, and the final one presents the whole hand. All of the outputs are simple dense layers, corresponding to flattened 3D joint coordinates (see

Fig.3.5 and Table.3.1). Because the regression for each coordinate is independent in our model, most of the regression loss functions are reasonable to be used. A current study [24], however, points out that a modified smooth *L1 loss* [33] (see Eq.3.1) can help hand pose estimation model to get better performance, as the effect of outliers will be reduced. Our experiment agrees with this conclusion and we adapt it to our model.

$$smooth_{L_1} = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{3.1}$$

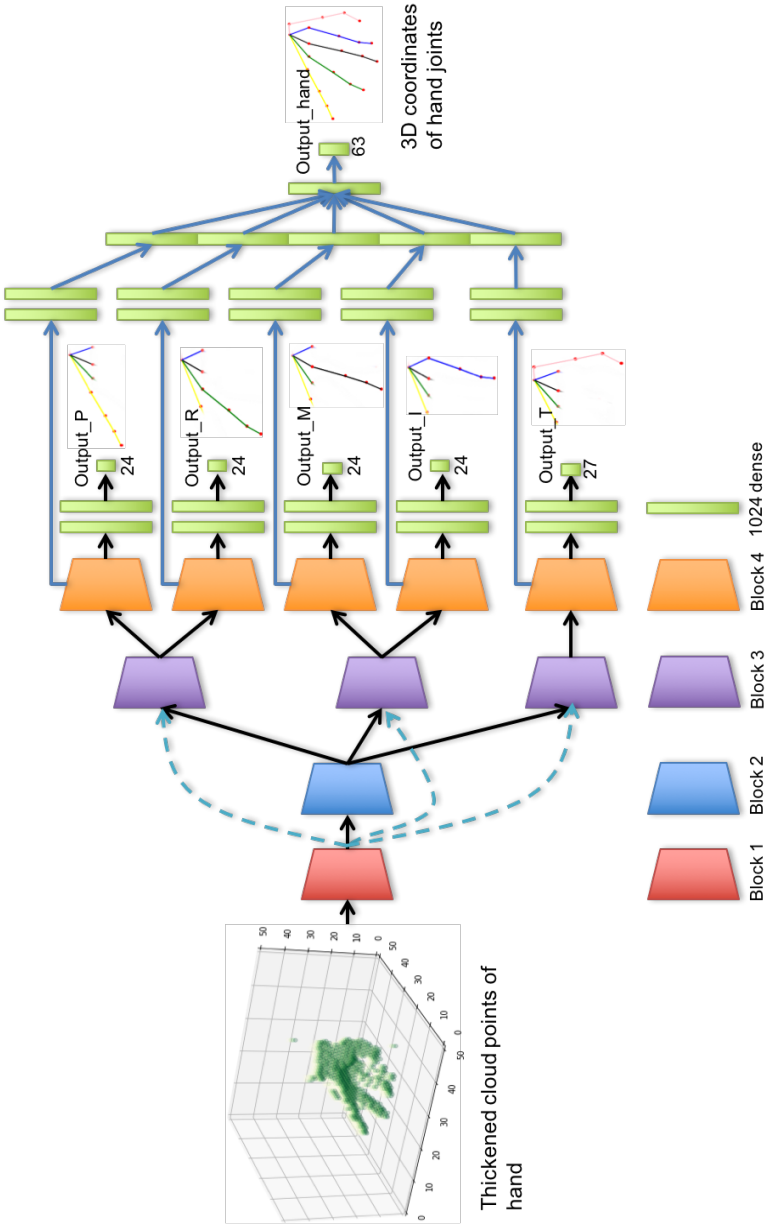

Figure 3.3: The architecture of our proposed 3D hand pose estimator.

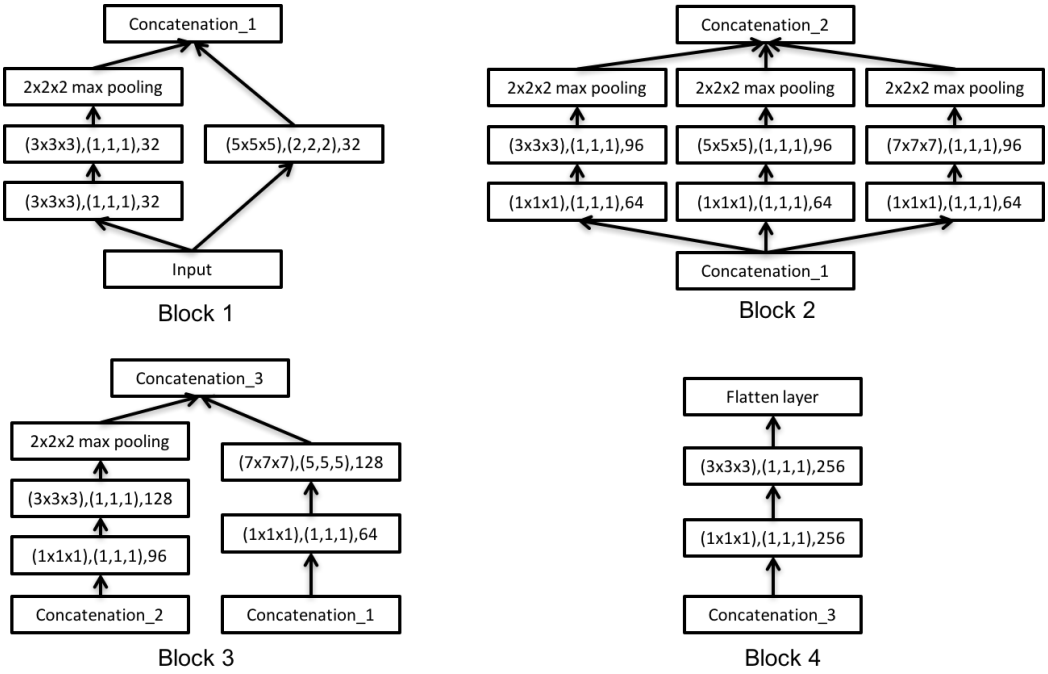

Figure 3.4: The architecture of each blocks.

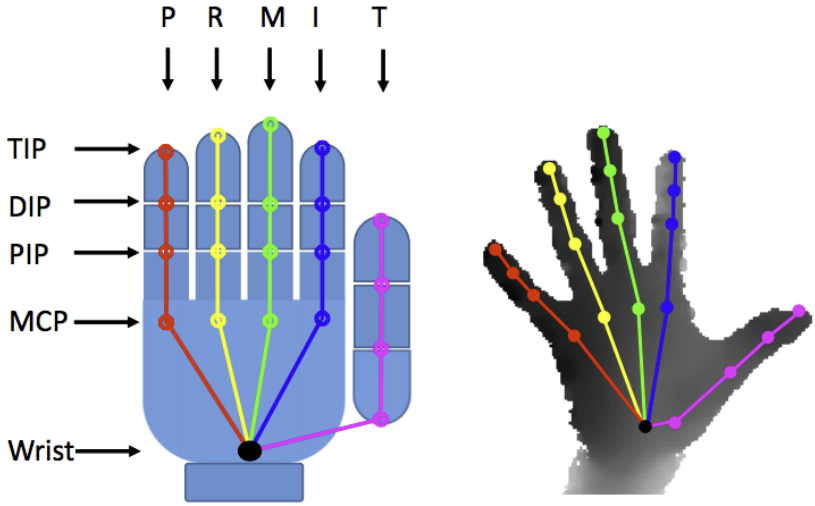

Figure 3.5: The denotation of hand joints(Image Source: [1]).

Output	Dense Layer Dimension	Include Joints ID
Out_T	27	Wrist, TMCP, TPIP, TDIP, TTIP,IMCP, MMCP, RMCP, PMCP
Out_I	24	Wrist, IMCP, IPIP, IDIP, ITIP, MMCP, RMCP, PMCP
Out_M	24	Wrist, IMCP, MMCP, MPIP, MDIP, MTIP, RMCP, PMCP
Out_R	24	Wrist, IMCP, MMCP, RMCP, RPIP, RDIP, RTIP, PMCP
Out_P	24	Wrist, IMCP, MMCP, RMCP, PMCP, PPIP, PDIP, PTIP
Out_Hand	63	All Joints

Table 3.1: The outputs of proposed 3D hand pose estimator

3.2.2 The training of pose estimator

In the HIM2017 challenge, there are totally 957,032 training samples, including the depth image and the corresponding 3D hand pose.

Due to the variety of distance between hand and camera, the same hand could be projected to different scale in the depth image. To remedy the scale variance, it may need to increase model parameters and perform massive data augmentation. Leveraging the depth information, nonetheless, could reduce the scale invariance (see Fig.3.6).

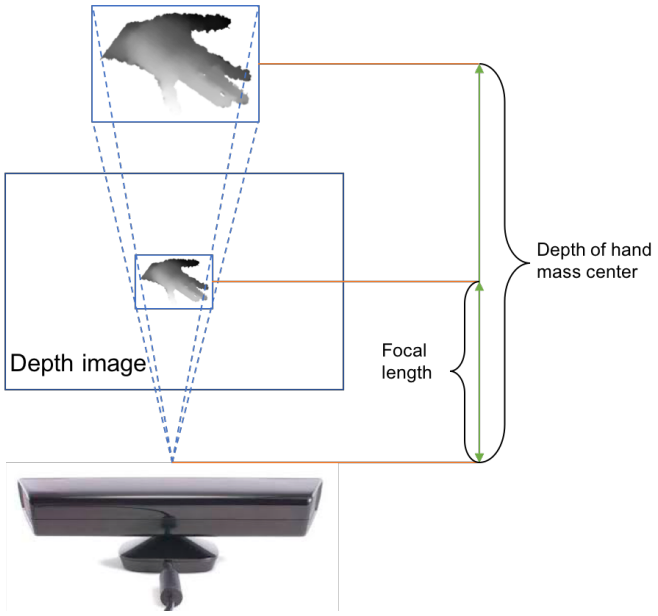

Figure 3.6: Geometric relationship between the real hand and its projected depth image

The focal length of depth camera is given, as $f = 475\ mm$. Using the depth of hand mass center, which is denoted by d, we can calculate a scale (see Eq.3.2) to rescale the hand area, making 1 *pixel* to be identify to 1 mm in the resized image. Despite a distortion caused by the depth difference of hand existing, for the sake of simplicity, we ignore it, because the distortion should be within 2%.

$$Scale = \frac{Real\ hand\ size}{Projected\ hand\ size} = \frac{d}{f} \tag{3.2}$$

$$l \ = \ 30 \ / \ Scale \tag{3.3}$$

where the Scale is from Eq.(3.2).

Referring to the 2D pose and adding a margin of 30 mm, which is rescaled to be l (see Eq.(3.3)), we generated a rectangle to bound each depth image in the training set. The hand area can be cropped by this rectangle. Afterwards, we removed all cloud points which are in front of the minimum pose depth and behind of the maximum pose depth, with a depth margin 30 mm. Through previous processing, the size of hand area is very close to the real hand. As we assume the real hand size within the bounding box is less than 250 mm, to unify the input size, we transform the resized hand area to the center of a blank image with the size 250 $\times$ 250.

Although the training set is computationally large, due to the high flexibility of hand pose, it could be statistically small. To alleviate such a potential discrepancy between the distribution of training samples and testing samples, we have to apply data augmentation. Since storing the augmentation data takes a large amount of space and the same data will be used more than once, we utilized an on-line augmentation technique [23]. More concretely, for each input, we applied a randomly sampled affine transformation matrix to both depth image and hand pose before feeding them to the model. The randomly sampled affine transformation matrix can be formulated as Eq.(3.4):

$$\begin{aligned} M = & \begin{bmatrix} s_x \cos\theta & -s_y \sin\theta & t_x \\ s_x \sin\theta & s_y \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \\ & and \\ & s_x \sim Uniform(0.9, 1.1) \\ & s_y \sim Uniform(0.9, 1.1) \\ & t_x \sim TruncatedNormal(0, 10, -15, 15) \\ & t_y \sim TruncatedNormal(0, 10, -15, 15) \\ & \theta \sim TruncatedNormal(0, 30, -90, 90) \end{aligned} \tag{3.4}$$

where s_x and s_y are scaling factors, while t_x and t_y are shifting distance, corresponding to x and y direction respectively, and θ is the rotation angle.

To summarize, from the original depth image, we used given pose and bounding box to segment the hand area out, then resized it to a real size, followed by locating it to an image of size 250×250. In the end, we did an on-line data augmentation and transformed the depth image into a 3D volume ($50 \times 50 \times 50$). The main procedure of preprocessing pipeline is shown in Fig.3.2.2.

For better inspecting how the 3D hand pose estimator was trained, we show experiment parameters in Table.3.2.

No. of model parameters	58,801,498
Optimizer	Adam (default setting of Keras)
Batch size	32
Epoch	20
Training time	160 hours (including online augmentation)
GPU	TITAN X $\times 1$
CPU	Intel E5-1650 v3 $\times 1$
GPU occupation	99%
CPU occupation	99%

Table 3.2: Experiment parameters for training the pose estimation model

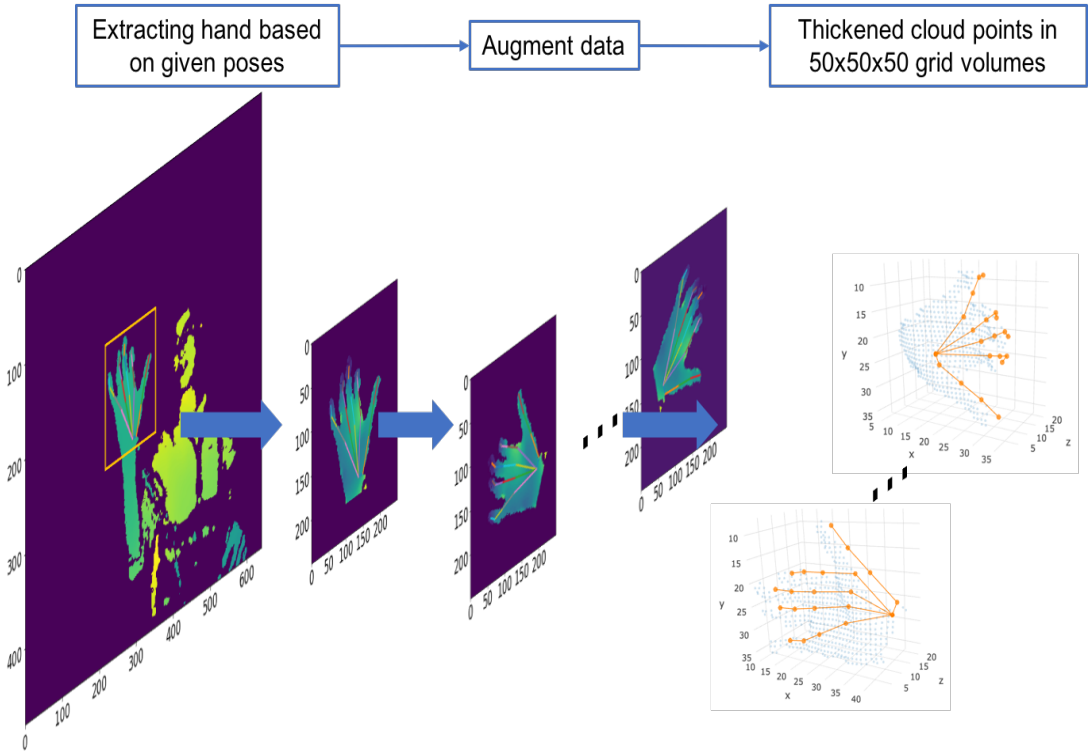

Figure 3.7: The pipeline of data preprocessing for hand estimator

3.3 Hand detector

The detector is developed from the U-net [32]. We modified the U-net to adapt to our task (see Fig.3.8), by substituting the last layer of each top-down box to be a dialed convolution, which increases the perception field [34]. For our model, the depth image is the input, while a binary mask (hand area with value 1, background with value 0) is the output.

To train the hand detector, we generated corresponding hand masks from depth images of the training set. In the previous section, we already got the hand area for all samples in the training set. If setting the value of hand area to be 1 while other places to be 0, a mask for hand area is generated. In order to ensure the computational efficiency, we resized both of depth images and masks to be of size 240×320. Furthermore, we also apply on-line data augmentation to train the hand detection model, with larger shifting distance but constrain the transformation of the hand mask within the image window. The randomly sampled affine transformation matrix can be formulated as Eq.(3.5):

$$
\begin{aligned}
M &= \begin{bmatrix} s_x \cos\theta & -s_y \sin\theta & t_x \\ s_x \sin\theta & s_y \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \\
&and \\
&\quad s_x \sim Uniform(0.9, 1.1) \\
&\quad s_y \sim Uniform(0.9, 1.1) \\
&\quad left = pose_{left} - 20 \\
&\quad right = 320 - pose_{right} - 20 \\
&\quad up = pose_{up} - 20 \\
&\quad down = 240 - pose_{down} - 20 \\
&\quad t_x \sim TruncatedNormal(0, 80, -left, right) \\
&\quad t_y \sim TruncatedNormal(0, 80, -up, down) \\
&\quad \theta \sim TruncatedNormal(0, 30, -45, 45)
\end{aligned}
\tag{3.5}
$$

where s_x and s_y are scaling factors, while t_x and t_y are shifting distance, corresponding to x and y direction respectively, and θ is the rotation angle. Here, we

use $left, right, up, down$ to guarantee the hand will not be shifted outside of the depth image.

For better inspecting how the hand detector was trained, corresponding experiment parameters are shown in Table.3.3

No. of model parameters	2,316,865
Optimizer	Adam (default setting of Keras)
Batch size	64
Epoch	5
Training time	25 hours (including online augmentation)
GPU	TITAN X $\times 1$
CPU	Intel E5-1650 v3 $\times 1$
GPU occupation	99%
CPU occupation	99%

Table 3.3: Experiment parameters for training the hand detection model

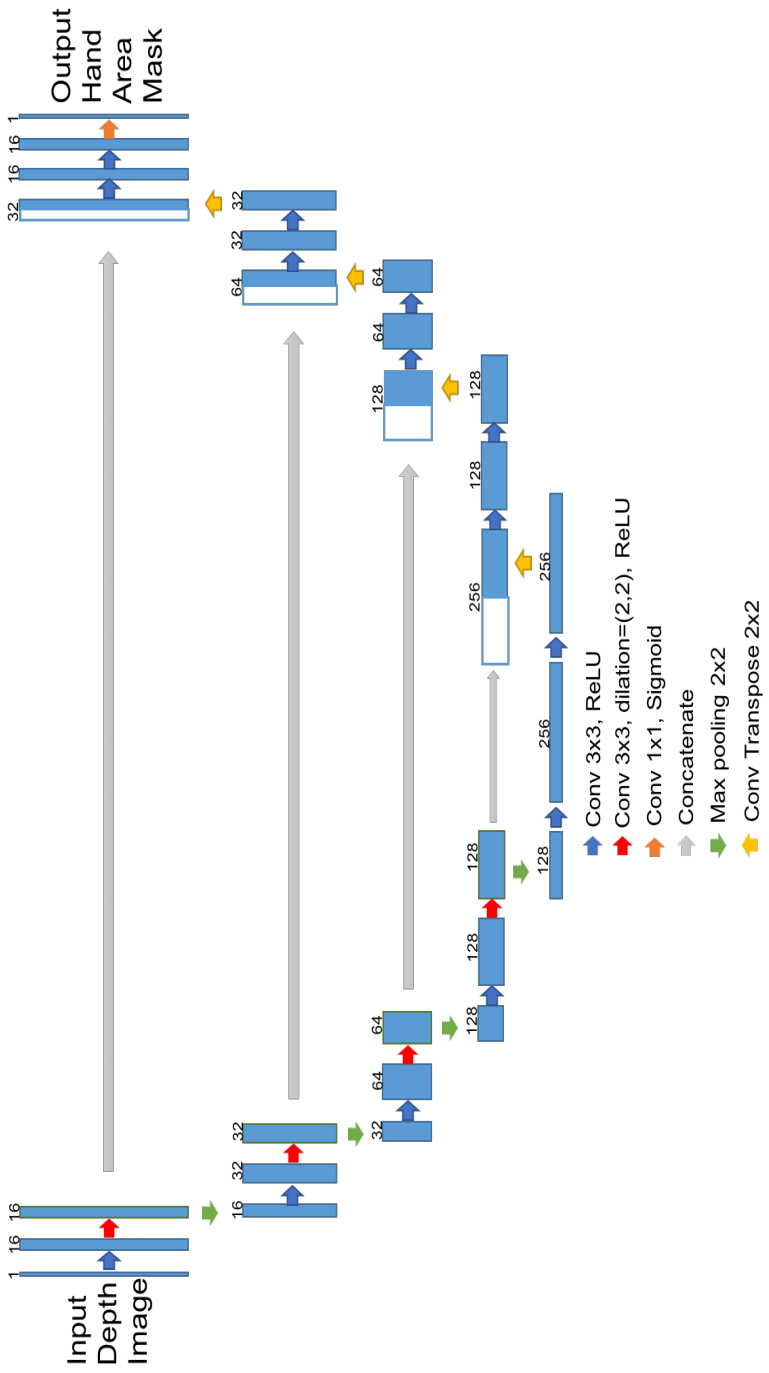

Figure 3.8: The architecture of hand detector

3.4 Hand verifier

Even though the detector can segment the hand area in most of the cases, it might fail when samples from the training set and testing set are quite different. To make our tracking model more robust, we used a hand verifier to verify whether a hand is correctly detected. If not, we use poses from the previous frame for current frame. What the hand verifier verifies can be described by following two items:

1. Comparing with the previous frame, whether the center of detected hand area shift more than 150 mm;

2. Whether the number of pixels for detected hand area is more than 1000.

4 Evaluating the tracking system on HIM2017 benchmark

On HIM2017 benchmark, we particularly evaluated our 3D hand pose estimator on the single-frame 3D hand pose estimation task, and evaluated the whole tracking system on the 3D hand pose tracking task.

4.1 Evaluating the 3D hand pose estimator

4.1.1 Experiment parameters of tracking system

Since experiment parameters are important for fully evaluate a neural network model, we demonstrate experiment parameters in testing the pose estimator in Table.4.1.

No. of model parameters	58,801,498
Batch size	1
Execution Speed	18 FPS (including preprocessing depth image)
GPU	TITAN X ×1
CPU	Intel E5-1650 v3 ×1
GPU occupation	30% - 45%
CPU occupation	80% - 92%

Table 4.1: Experiment parameters for testing the 3D hand pose estimator

From this table, we can see the number of parameters in our model is very large, as $58,801,498$, due to we constructed our estimator by beaches structure. This heavy structure causes the slow running speed. It can process 18 depth images per second, including the preprocessing.

4.1.2 The performance of 3D hand pose estimator

The single-frame 3D hand pose estimation task contains 295,510 depth images, which were collected from objects from the training set. Besides, bounding boxes for hand area are given in this testing set.

Starting from the nearest point of the image patch, we set all of the depth values that are 220 mm behind the nearest point to be 0. By doing this, for the majority of samples, only the hand area remains, nonetheless, we found that the hand area was also removed for some samples. Because some small noise points could be in front of the hand area, we may get the incorrect nearest point. To address this issue, we separated the whole depth image into multiple channels whenever the depth gap is larger than 5 mm. If the area ratio of point clouds to the image patch size is less than 0.04 (from our experiments), we delete all point clouds in this channel. Through this way, the small noise is removed in front of the hand area. Other preprocessing is the same to the training.

We applied such kind of preprocessing to 295,510 depth images, and used pose estimator to predict the corresponding pose one by one (i.e., batch size = 1). Six outputs are used in the training stage, however, for the testing stage, we only take the output of the whole hand. As the coordinate system of the output is different from the original image, we have to transform the output pose back to the original one. This transformation simply reverses the transformation of preprocessing.

We list the lead-board of HIM2017 single frame task in Table.4.2. Three error metrics, the average error (AVG ERROR) of all hand joints, visual hand joints(SEEN ERROR) and occluded hand joints (UNSEEN ERROR), are used for evaluating the performance.

The average error of our 3D hand pose estimator is 11.90 mm, and qualitative results can be visualized in Fig.4.1.2. Compare to the Baseline [35], which is the state-of-the-art method in *ECCV2016*, we have reduced the average error from 19.71 mm to 11.70 mm, achieving a significant improvement for 40%. However, we are still have a gap of 1.95 mm, referring to the state-of-art model performance.

By inspecting estimated results, we found our model is not generative enough for testing samples that are considerably different from training samples. Thus, the data augmentation method needs to be improved. In addition, the low reso-

Team name	AVG ERROR (mm)	SEEN ERROR (mm)	UNSEEN ERROR (mm)
SNU CVLAB	9.95	6.97	12.43
NVIDIA Research and UMontreal	9.97	7.55	12.00
NTU	11.30	8.86	13.33
THU VCLab	11.70	9.15	13.83
NAIST RVLab	11.90	9.34	14.04
Baseline	19.71	14.58	23.98

Table 4.2: Average error on the leader-board of HIM2017 single frame task. Our team name is NAIST RVLab

lution of 3D volume limits the performance of our model. However, it remains an issue as of how to balance the trade-off between computation cost and prediction accuracy.

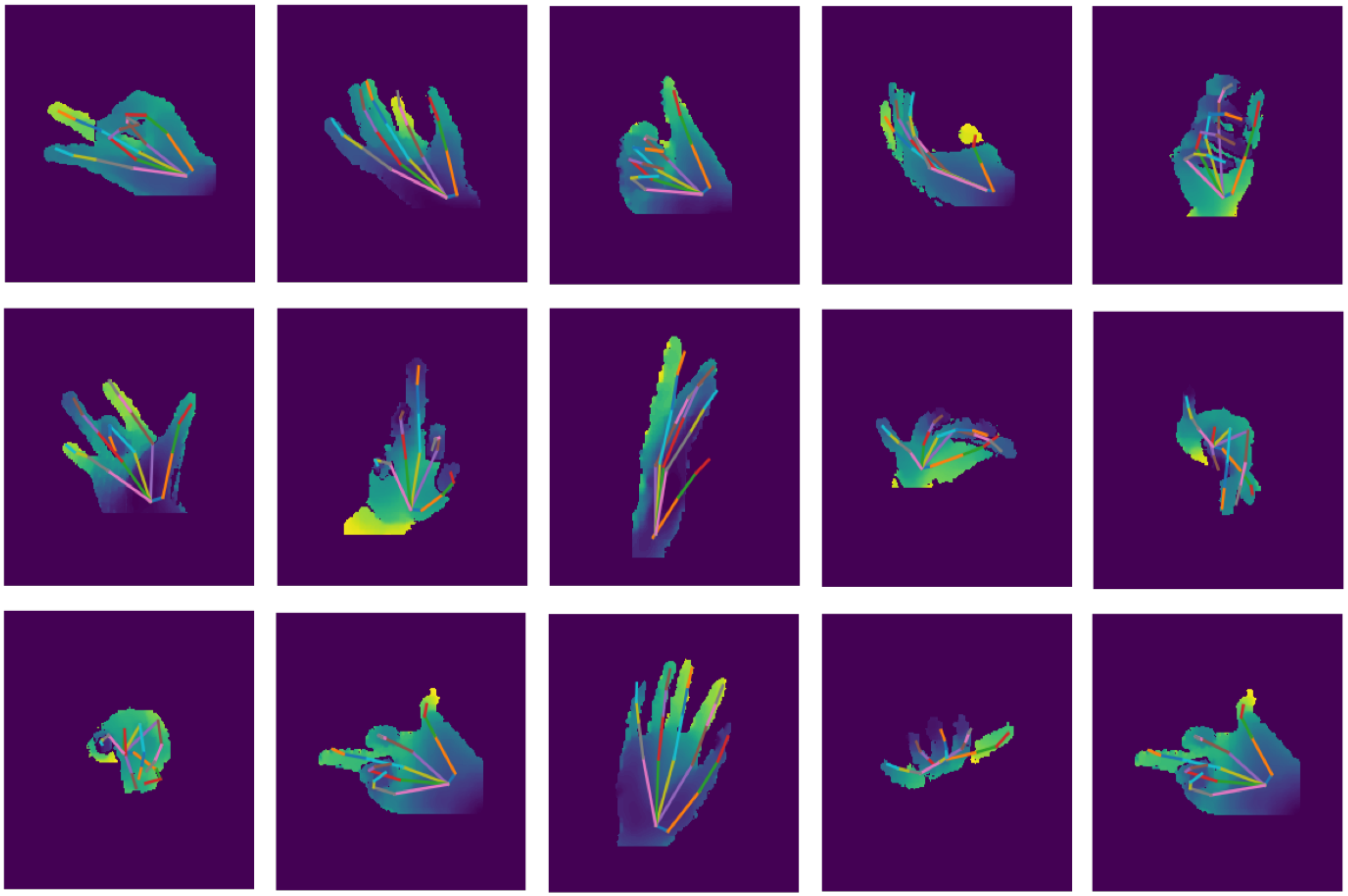

Figure 4.1: Qualitative results of 3D hand estimator

4.2 Evaluating the tracking system

4.2.1 Experiment parameters of tracking system

The tracking system includes two neural network models (the detector and the pose estimator), we have shown experiment parameters of the hand pose estimator in Table.4.3. Here, we demonstrate experiment parameters of the hand detector and their combination, in Table.4.3 and Table.4.4, respectively.

The hand detector has parameters of $2,316,865$, which is relatively smaller than the hand estimator, due to using fully-convolutional network. Including the preprocessing, its execution speed is 50 FPS. By the hand detector itself, real-time execution is achieved.

No. of model parameters	2,316,865
Batch size	1
Execution Speed	50 FPS (including preprocessing of testing images)
GPU	TITAN X $\times 1$
CPU	Intel E5-1650 v3 $\times 1$
GPU occupation	20% - 40%
CPU occupation	90% - 96%

Table 4.3: Experiment parameters for testing the hand detection model

Integrating the hand detector, the hand verifier and the hand pose estimator together, the execution speed of the whole hand pose tracking system drops to 12 FPS. The number of parameters also increased to $61,118,363$ dramatically. Therefore, the execution speed and model parameters are two components we need to optimize in the feature work.

4.2.2 The performance of tracking system

There are 99 videos in the 3D hand pose tracking task, which entirely contain 294,006 depth images. For each video, depth images are organized by sequence, besides, 3D hand pose of the first frame is given. As a result, the context information can be used for pose estimation.

No. of model parameters	61,118,363
Batch size	1
Execution Speed	12 FPS (including preprocessing of testing images)
GPU	TITAN X $\times 1$
CPU	Intel E5-1650 v3 $\times 1$
GPU occupation	30% - 50%
CPU occupation	100%

Table 4.4: Experiment parameters for testing the tracking system

The lead-board of HIM2017 tracking task is described in Table.4.5. The average error of our 3D hand tracking system is 12.64 mm, and qualitative results can be visualized in Fig.4.2.2. Comparing to the Baseline [35], we made an improvement for 39%.

Team name	AVG ERROR (mm)	SEEN ERROR (mm)	UNSEEN ERROR (mm)
NVIDIA Research and UMontreal	10.51	8.21	12.37
NAIST RVLab	12.64	10.20	14.62
THU VCLab	13.65	11.02	15.70
Baseline	20.63	16.04	24.36

Table 4.5: Average error on the leader-board of HIM2017 tracking task. Our team name is NAIST RVLab

Despite the 3D hand estimator of THU VCLab performs better than ours in the single-frame estimation task, by involving the hand detector and hand verifier in our tracking system, we got better results than THU VCLab. For NVIDIA Research and UMontreal, their 3D hand estimator and tracking system both perform better than ours, and differences are quite similar, as 1.95 mm and 2.13 mm, respectively. Therefore, in terms of the whole tracking system, our 3D hand estimator could be "the shortest stave" of "Liebig's barrel". Our hand estimator needs to be improved if we want to achieve better performance for the tracking system.

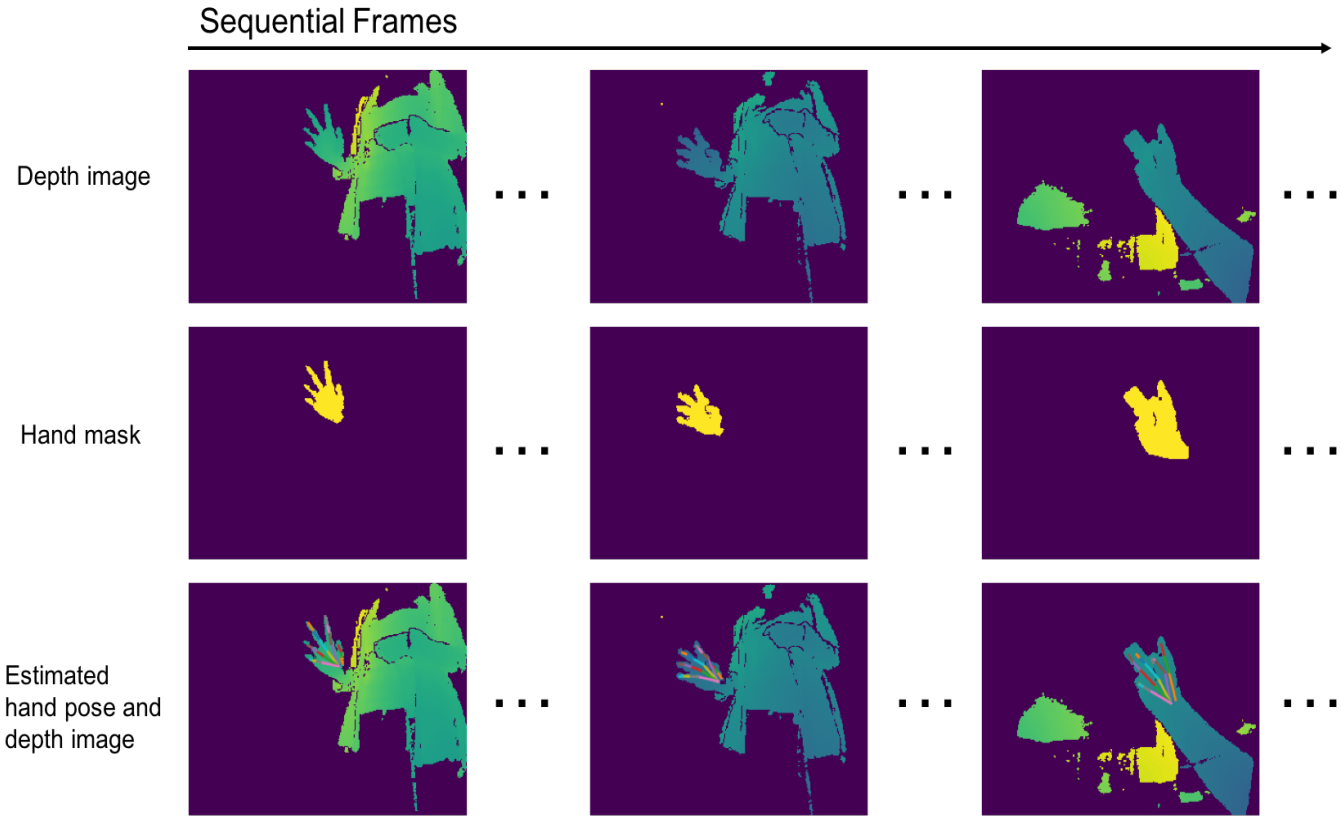

Figure 4.2: Qualitative results of 3D hand tracking system

5 Extension for hand-object interactive pose estimation

Although our main focus is the tracking system, we find the applying our tracking system can be simplify extended for on the hand-object interactive pose estimation, and our result placed first on the HIM2017 benchmark.

5.1 The system for hand-object interactive pose estimation

In the hand-object interactive pose estimation task of HIM2017 challenge, there are 2965 samples. We randomly chosen 80 samples from the testing set (2.7% of 2965 samples), and manually create 80 masks to train a hand-object segmentation model, which aims to segment the hand from the background and interactive objects. In the end, only the segmented hand area is used for pose estimation. Comparing to the tracking system, we removed the hand verifier. The system is shown in Fig.5.1.

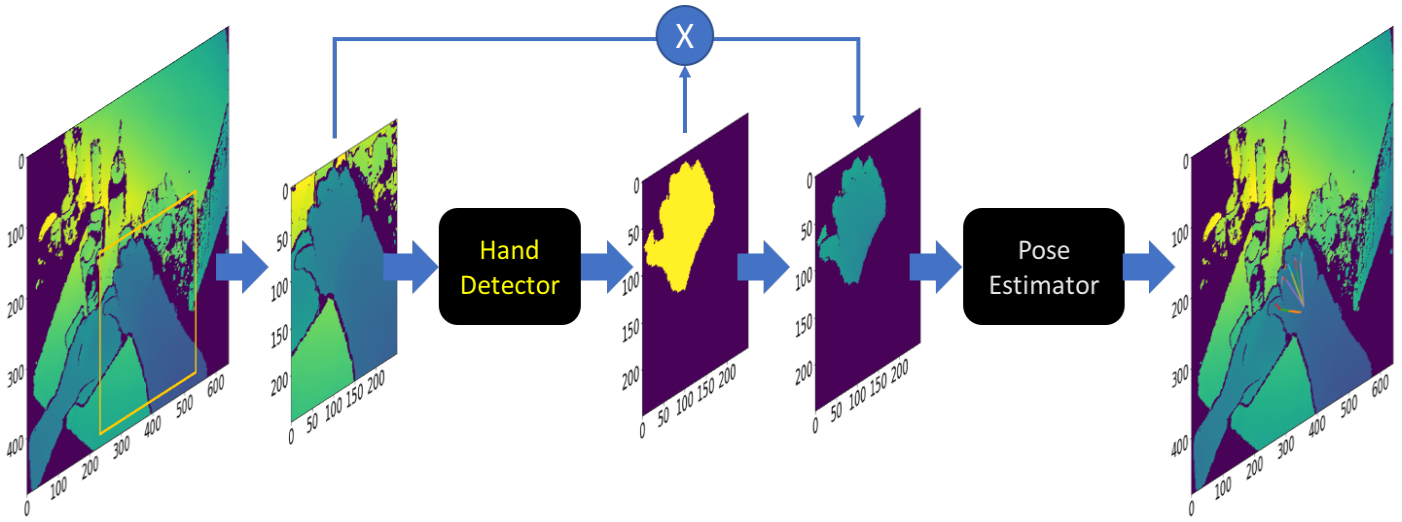

Figure 5.1: Hand-object interaction 3D hand pose estimation system

5.2 Evaluating the system for hand-object interactive pose estimation

The lead-board of HIM2017 hand-object interaction task is shown in Table.5.1. Our qualitative results can be visualized in Fig.5.2.

Team name	AVG ERROR (mm)
NAIST RVLab	24.98
THU VCLab	29.19
NVIDIA Research and UMontreal	32.44
Baseline	46.10

Table 5.1: Average error on the leader-board of HIM2017 Hand-object interaction task. Our team name is NAIST RVLab

Even though we got the largest average error in the hand-object interaction pose estimation as 24.98 mm, our performance is much better than other groups. We suppose, the reason for a large error is lacking of training data, since the correlated samples are not included in the training set. While the reason of our model performs better than other groups is that we use hand detector to segment hand from its interactive objects before performing pose estimation.

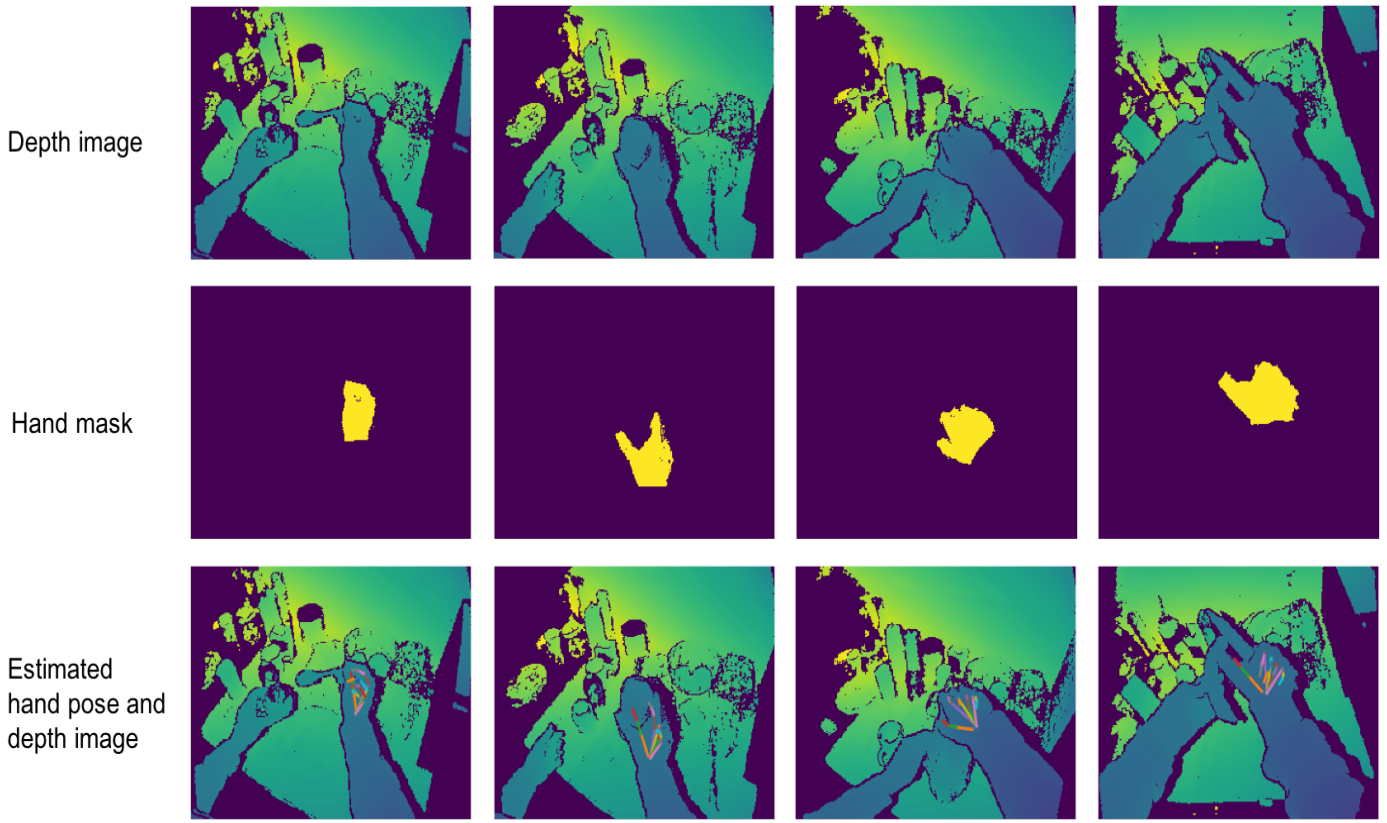

Figure 5.2: Qualitative results of 3D object-hand interactive pose estimation

6 Conclusion

In conclusion, this paper mainly presents two works: (1) Building a deep-learning-based 3D hand pose tracking system and evaluate it on the HIM2017 benchmark, and it ranks the second place. (2) Applying the modified tracking system on the object-interactive hand pose estimation. We also evaluate it on the HIM2017 benchmark and placed first. Overall, the performance is summarized in Fig.6.1. Even though our tracking system cannot archive the state-of-art, it is available to be applied in related applications, with a execution speed of 12 FPS and average error of 12.64 mm. More importantly, the performance of our modified tracking system on the object-interactive hand pose estimation indicates, using hand detector to segment hand from its interactive objects before performing pose estimation can make better results than directly performing pose estimation. In the feature work, we may use the similar system to tackle double-hand interactive pose estimation problem, and a related dataset, which involves the hand mask and 3D hand pose, will be built.

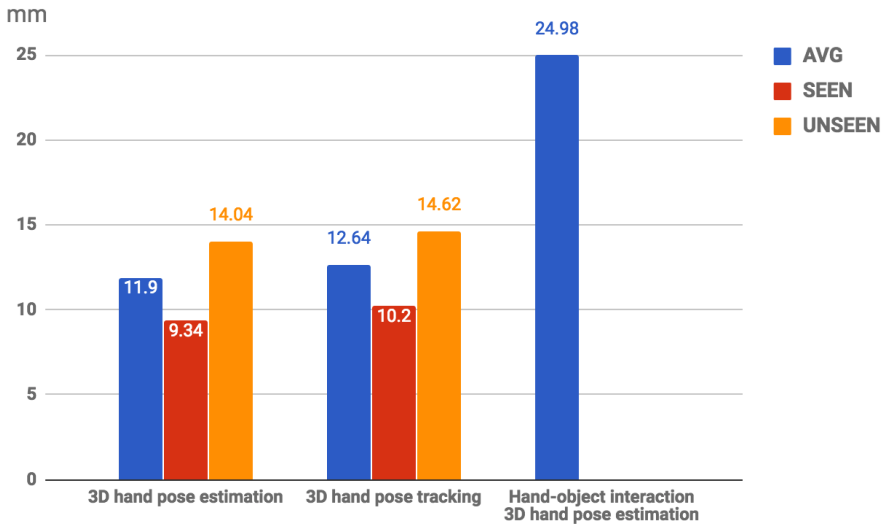

Figure 6.1: The evaluation performance of 3D hand pose estimation, 3D hand pose tracking and Hand-object interaction 3D hand pose estimation.

Acknowledgements

In the beginning, I am sincerely grateful to all of my advisers, as Prof. Kazushi Ikeda, Assoc. Prof. Takatomi Kubo and Asst. Prof. Yang Wu.

When I was seeking a study opportunity in Japan, Prof. Kazushi Ikeda nicely gave me an invitation to the mathematical informatics laboratory and accepted me as a master student latter. Although my research direction has shifted away from the mainstream of his laboratory, he still keeps supporting me and offering his generous help. I am wholeheartedly honored to be his student.

When I made my first step into the machine learning study, Assoc. Prof. Takatomi Kubo kindly gave me a tutorial in the Bayesian Nonparametric models, which remarkably helps me to further understand plenty of essential machine learning models. Without his guidance, it may be hard for me to start my study and research. I very appreciate his help.

When I tried to make my research direction on computer vision, Asst. Prof. Yang Wu, who is from the robotics vision laboratory, friendly invited me to cooperate with his lab members. With the resource and advice from him, I am quite enjoying my current research. In addition, I also want to say a lot of thanks to Prof. Takeo Kanade. Even we just met three times, but he gave me some excellent insights and encouragement.

In addition, I would like to thank all of the teachers, students, and staffs in NAIST, since they always take their pleasure to help me in my life and study.

In the end, I would like to thanks the JASSO scholarship, the Heiwa Nakajima Foundation Scholarship and the tuition exemption of NAIST. These financial supports significantly help my study in Japan. Without them, I may not be able to focus on my research. Many thanks again.

References

[1] Shanxin Yuan, Qi Ye, Guillermo Garcia-Hernando, and Tae-Kyun Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv preprint arXiv:1707.02237*, 2017.

[2] Ali Erol, George Bebis, Mircea Nicolescu, Richard D Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1):52–73, 2007.

[3] xinghaochen. Awesome works on hand pose estimation. `https://github.com/xinghaochen/awesome-hand-pose-estimation`, 2017.

[4] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*, pages 119–137. Springer, 2013.

[5] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3456–3462, 2013.

[6] James S Supancic, Grégory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: data, methods, and challenges. In *Proceedings of the IEEE international conference on computer vision*, pages 1868–1876, 2015.

[7] Hui Liang, Junsong Yuan, Daniel Thalmann, and Zhengyou Zhang. Model-based hand pose estimation via spatial-temporal hand parsing and 3d fingertip localization. *The Visual Computer*, 29(6-8):837–848, 2013.

[8] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Tracking the articulated motion of two strongly interacting hands. In *Computer Vision*

and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1862–1869. IEEE, 2012.

[9] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014.

[10] Dimitrios Tzionas, Luca Ballan, Abhilash Srikantha, Pablo Aponte, Marc Pollefeys, and Juergen Gall. Capturing hands in action using discriminative salient points and physics simulation. *International Journal of Computer Vision*, 118(2):172–193, 2016.

[11] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014.

[12] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin.

[13] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015.

[14] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *arXiv preprint arXiv:1704.02463*, 2017.

[15] Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhand Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. *arXiv preprint arXiv:1704.02612*, 2017.

[16] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[17] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision*, pages 852–863. Springer, 2012.

[18] Peiyi Li, Haibin Ling, Xi Li, and Chunyuan Liao. 3d hand pose estimation using randomized decision forest with segmentation index points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 819–827, 2015.

[19] Danhang Tang, Hyung Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d hand poses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[20] Meysam Madadi, Sergio Escalera, Xavier Baro, and Jordi Gonzalez. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*, 2017.

[21] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 5, 2017.

[22] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

[23] Markus Oberweger and Vincent Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV workshop*, volume 840, page 2, 2017.

[24] Hengkai Guo, Guijin Wang, Xinghao Chen, and Cairong Zhang. Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017.

[25] Hengkai Guo, Guijin Wang, Xinghao Chen, Cairong Zhang, Fei Qiao, and Huazhong Yang. Region ensemble network: Improving convolutional network for hand pose estimation. *arXiv preprint arXiv:1702.02447*, 2017.

[26] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. V2v-posenet: Voxel-to-voxel prediction network for accurate 3d hand and human pose estimation from a single depth map. *arXiv preprint arXiv:1711.07399*, 2017.

[27] Heung-Il Suk and Bong-Kee Sin. Dynamic bayesian network based two-hand gesture recognition. *Journal of KIISE: Software and Applications*, 35(4):265–279, 2008.

[28] MK Bhuyan, Debanga Raj Neog, and Mithun Kumar Kar. Fingertip detection for hand pose recognition. *International Journal on Computer Science and Engineering*, 4(3):501, 2012.

[29] Xia Liu and Kikuo Fujimura. Hand gesture recognition using depth data. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 529–534. IEEE, 2004.

[30] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[31] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[33] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[34] Chuck-Hou Yee. Heart disease diagnosis with deep learning. https://blog.insightdatascience.com/heart-disease-diagnosis-with-deep-learning-c2d92c27e730, 2017.

[35] Qi Ye, Shanxin Yuan, and Tae-Kyun Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *European Conference on Computer Vision*, pages 346–361. Springer, 2016.

Publication List

[1] Yang, Fan, et al. “A Hierarchical Mixture Density Network.” International Conference on Neural Information Processing. Springer, Cham, 2017.

[2] Yang, Fan, et al. “Application of SsVGMM to medical data-classification with novelty detection.” Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE. IEEE, 2017.

[3] Yang, Fan, et al. “A deep-learning-based 3D hand pose tracking system.” The 12th International Workshop on Robust Computer Vision (IWRCV), 2018.

[4] Shanxin Yuan, Guillermo Garcia-Hernando, Bjorn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, Junsong Yuan, Xinghao Chen, Guijin Wang, Fan Yang, Kai Akiyama, Yang Wu, Qingfu Wan, Meysam Madadi, Sergio Escalera, Shile Li, Dongheui Lee, Iason Oikonomidis, Antonis Argyros, Tae-Kyun Kim. “3D Hand Pose Estimation: From Current Achievements to Future Goals.” arXiv preprint arXiv:1712.03917 (2017). (Submitted to the IEEE International Conference on Computer Vision, 2018.)