Hand Pose Estimation: A Survey

Bardia Doosti
School of Informatics, Computing and Engineering
Indiana University Bloomington
Email: bdoosti@indiana.edu

Abstract

The success of Deep Convolutional Neural Networks (CNNs) in recent years in almost all the Computer Vision tasks on one hand, and the popularity of low-cost consumer depth cameras on the other, has made Hand Pose Estimation a hot topic in computer vision field. In this report, we will first explain the hand pose estimation problem and will review major approaches solving this problem, especially the two different problems of using depth maps or RGB images. We will survey the most important papers in each field and will discuss the strengths and weaknesses of each. Finally, we will explain the biggest datasets in this field in detail and list 21 datasets with all their properties. To the best of our knowledge this is the most complete list of all the datasets in the hand pose estimation field.

1 Introduction

Hand pose estimation is currently getting a lot of attention in the computer vision field. Since the invention of Deep Learning, researchers started to apply it in all computer vision fields and get a breakthrough result and hand pose estimation was not an exception. In addition the RGBD cameras which produce depth map have become cheap, which lowers the cost of making and using hand base systems. On the other hand, huge investment of big tech companies like Google, Microsoft and Facebook on Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR) technology as new interactive personal computers, has broadened the applications of this field. Consequently, a relatively new branch in Human Computer Interaction (HCI) has been introduced to study the systems controlled by understanding user's hands. In [20], Lee *et al.* detected hands to render an object in AR environment on the hand which was proportional to hand size. Piumsomboon *et al.* [27] focused on *guessability* in 40 different tasks in AR environment with studying hand gestures. Jang *et al.* [16] build an AR/VR system in egocentric viewpoint which was completely controllable via

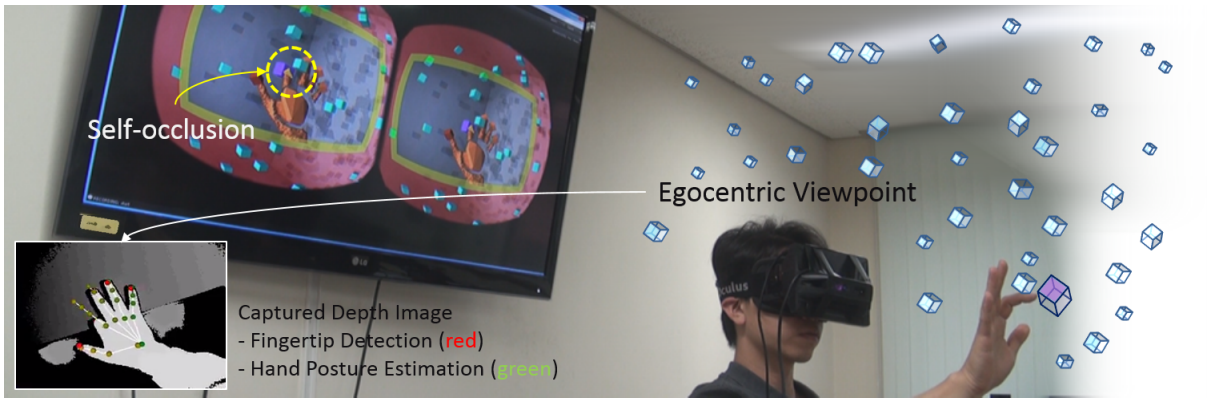


Figure 1: Application of hand pose estimation in egocentric viewpoint in AR/VR headset to control objects shown in the display. Originally used in [16].

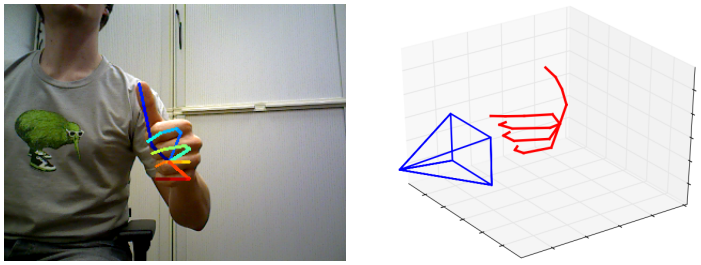
user's hands. Figure 1 shows one of the applications of hand pose estimation used in egocentric viewpoint in AR/VR headset [16].

Nevetheless the applications of hand pose estimation are not limited to AR/VR technologies. Sridhar *et al.* [40] built a system working with a number of finger actions. Markussen *et al.* [21] also proposed a mid-air keyboard to type on air. Yin *et al.* [54] used hand pose estimation to design a system that understands sign language. In a similar study, Chang *et al.* [4] used finger tip detection and tracking to read alphabet written by finger in the air. Outside HCI field, Shlizerman *et al.* and Rohrbach *et al.* applied body and hand pose estimation systems for predicting body movements of a piano player [34] and to detect activities [31].

In recent years, the interest in systems controlled by fingers, made researchers more ambitious to the extent that they discarded 2.5D depth map images and tried to estimate hand pose by a single RGB image. This method is a harder task and needs a considerable larger data to train. Below, we will first explain the hand pose estimation problem and discuss its variations and next we will discuss different methods in solving this problem. At the end of the paper we will briefly investigate new datasets in this field and will see how the size of datasets have changed dramatically through time.

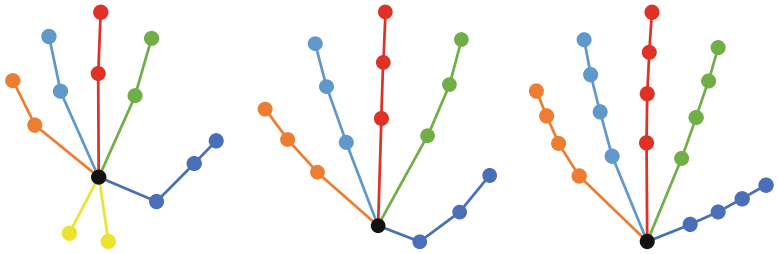
2 Hand Pose Estimation Problem

Hand pose estimation is the process of modeling human hand as a set of some parts (*e.g.* palm and fingers) and finding their positions in a hand image (2D estimation) or the simulation of hand parts positions in a 3D space Although it is also used to estimate hand with the phalanges (like [49] which is discussed in [55] as *strawberryfg* method), in almost all the recent papers hands are modeled as a number of joints and the task is equivalent to finding the position of these joints. We can then *estimate* the real hand pose using those joints. Figure 2 shows an image with its 2D and 3D estimation of hand pose using joints model connected with lines.



(a) (b)

Figure 2: (a) Image with 2D estimation of hand joints (b) 3D estimation of hand joints. Originally used in [59].



(a) 14 joints (b) 16 joints (c) 21 joints

Figure 3: Visualization of modeling hand with 14, 16 and 21 joints as used in NYU [47], ICVL [44], and MSRA [29] datasets. Originally used in [15].

There is no global agreement on the number of joints used to model hands in different datasets. Those studies which want to compare their results for different datasets, have to change their model for each individual dataset. Albeit, 21 joints is now the most popular model and most of the datasets and pre-trained networks use this model. Figure 3 shows different number of joints in three popular hand datasets.

3 Approaches

Before deep learning revolution, people used to apply traditional machine learning and computer vision techniques for hand pose estimation. Wang *et al.* [50] used a color glove for user and then by using nearest neighbor they found the position of each color in the image and therefore the position of each specific part of the hand (Figure 4). However, among all traditional approaches, random forest and its variations was the most popular one [19, 43–46, 53]. At that time, this method was the most successful which made its way in the commercial products as well. Using depth camera, in Kinect, Microsoft applied random forest as a classifier for human body pose estimation [35]. They first normalized depth map data using their neighbors value to be invariant to rotation. Then, they labeled each part of the body (similarly for hands) with a label and tried to classify each pixel (with its neighbor pixels) as one of these labels.

Figure 4: Hand tracking using color glove. Originally used in [50].

Publishing a parallel algorithm to run decision tree and random forest quickly [32], they used random forest (which comprises a number of decision trees) to classify each point in the map.

In what follows, we will explain all the major approaches in solving hand pose estimation problem using deep learning. Although we can categorize these methodologies into estimating 2D or 3D skeleton, detection-based and regression-based algorithms, using 2D or 3D CNN, we divided them into methods using depth maps and methods using RGB image or both. We first define two different types of networks used in this problem which we will use in explaining algorithms.

Detection-based Methods *vs.* Regression-based Methods In the detection-based method, the model produces a probability density map for each joint. So for example if a network uses 21-joints model for hands, for each image it will produce 21 different probability density maps as heatmaps. The exact location of each joint can be found by applying an *argmax* function on corresponding heatmap. In contrast, regression-based method tries to directly estimate the position of each joint. That is, if it uses 21-joints model, it should have 3×21 neurons in the last layer to predict (x, y, z) coordinates of each joint. Due to the high non-linearity, training a regression-based network requires more data and training iterations. But since producing a 3D probability density function for each joint is a heavy task for a network, regression-based networks is used in 3D hand pose estimation tasks. Below we will discuss papers from both classes.

3.1 Depth-based Methods

Traditionally, depth map image based methods were the main method in hand and body pose estimation. Sinha *et al.* [38] used a regression-based method to find 21 joints in the hand, based on a depth map. They tried to find the location of joints in each finger independently. To this end, they trained a separate network for each finger to regress three joints on that finger. Note that although they used depth map to regress the coordinates, they also used RGB to isolate the hand and to remove all the other pixels included in the hand's cropped frame. Unlike the next papers that we will discuss, Sinha *et al.* did not use a separate deep network for hand segmentation,

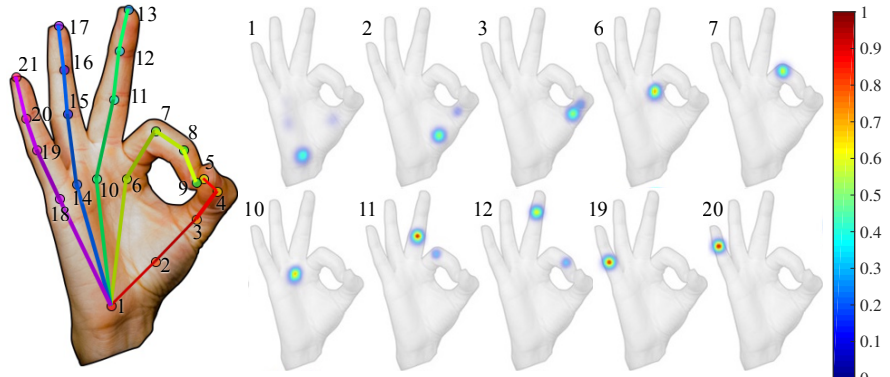


Figure 5: The output of a detection-based algorithm. For each joint in the hand, one probability density function will be generated which is depicted as heatmaps. Originally used in [36].

probably because of computation limitations. Instead they used RGB pixel color values to remove pixels which are not in the skin range.

In [2] Baek *et al.* used Generative Adversarial Network (GAN) [12] to estimate hand pose by making a one to one relation between depth disparity maps and 3D hand pose models. GAN is a specific CNN to generate new samples based on the previous learned samples. It consists of a discriminator and a generator network which are competing with each other to win the game. The discriminator network is a classifier trained to detect real and fake images. Generator is also a convolutional neural network which generates fake images based on a random initialization. These fake images should be good enough to deceive the discriminator as real ones. Conditioned GAN [26] is a special GAN which gets a real image and it is conditioned to generate an image similar to that one; *i.e.* the generator does not start from a random initialization.

Baek *et al.* used a CyclicGAN [58] which is a GAN for transferring one image from one domain to another. In this work one domain is the depth map of the hand and the other is the 3D representation of the hand joints. Baek *et al.* used a Hand Pose Generator (HPG) and a Hand Pose Discriminator (HPD) in their model. As you can guess from the above explanation, the HPG's job is to generate a hand, based on the 3D representation of the joints. In contrast, they used a Hand Pose Estimator (HPE) whose job is to generate the 3D hand pose, based on the input depth map. So in the training step HPG, HPD and HPE are optimized to reduce the error of HPE (which is the final goal of this algorithm) and to increase the consistency of the HPE-HPG combination $f^E(f^G)) : Y \rightarrow Y$ and the HPG-HPE combination $f^G(f^E)) : X \rightarrow X$. In the testing phase the algorithm refines the 3D model which is guided by the HPG to generate the best 3D model whose corresponding depth map is very similar to the input depth map. Figure 7 shows the schema of Baek *et al.* 's algorithm.

Ge *et al.* [9] created a new technique in the depth-based methods which was used in

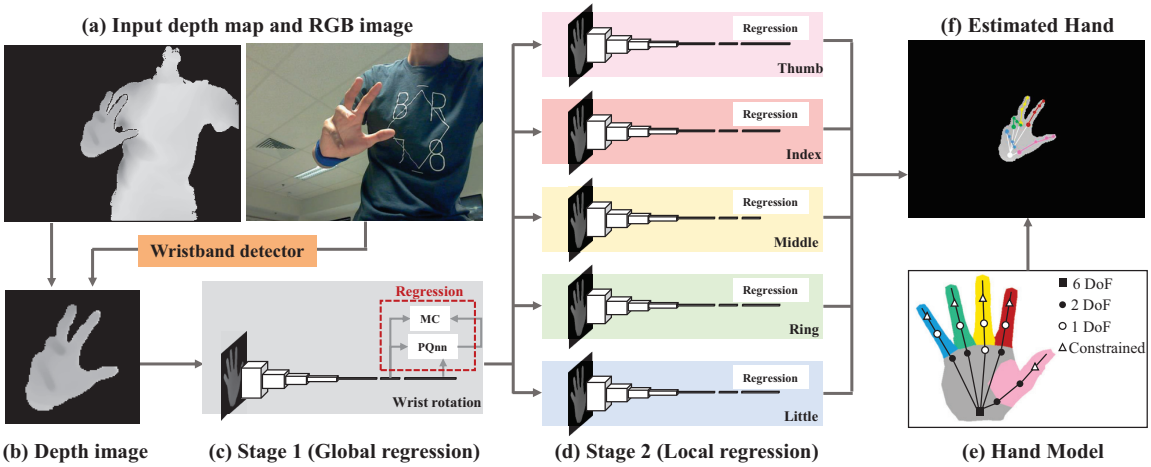


Figure 6: Sinha *et al.* 's multi network hand pose estimation. Originally used in [38].

their next studies and by the other teams multiple times. The main idea of this paper (and their next paper [10]) is to *estimate* 3D image from the 2.5D image and then estimate the hand pose from this new viewpoint. Although they did not mention how exactly this 3D model can be generated from depth-map, it can be understood from the cited papers, that they did not perform this part with machine learning algorithms.

As we know, depth maps only give us a surface of a hand, not a 3D shape. To estimate the 3D shape, they fix the camera in a 3D space and fix a surface as the farthest point in the camera's sight. So for every pixel in the depth map, proportional to the distance number, they should put voxels from that surface to the camera. With this method, the depth map produced from these voxels and the original depth map are the same. In the next step, they have to render these 3D volume from three perpendicular views; front, top and the side. To this end, they applied a Principle Component Analysis (PCA) on voxel's coordinates and picked the top three principal components and rendered 3D shape on those planes. They passed these depth maps to the network and got a probability for each joint in each xy, xz and yz planes. In the next step which they call *fusion* step, they mix the probabilities by multiplying them together.

In their next paper [10], Ge *et al.* performed the similar approach and used three different CNNs. But instead of 2D renders, they generated three 3D shapes with Truncated Signed Distance Function (TSDF). In TSDF shapes, signed distance of the voxel to the closest surface will be stored in each voxel. Also for the fusion step, instead of multiplying the probabilities, they concatenated the output vectors and used three fully connected layers to get the final result. They tested their method on MSRA [29] and NYU [47] datasets and got the best results with a good margin comparing to other methods. Figure 8 shows schema of Ge *et al.* 's [9] and [10] papers models.

Inspired by Qi *et al.* 's PointNet model in [28], Ge *et al.* [8] applied PointNet on their 3D shape estimate (HandPointNet). First they sampled from points in the 3D

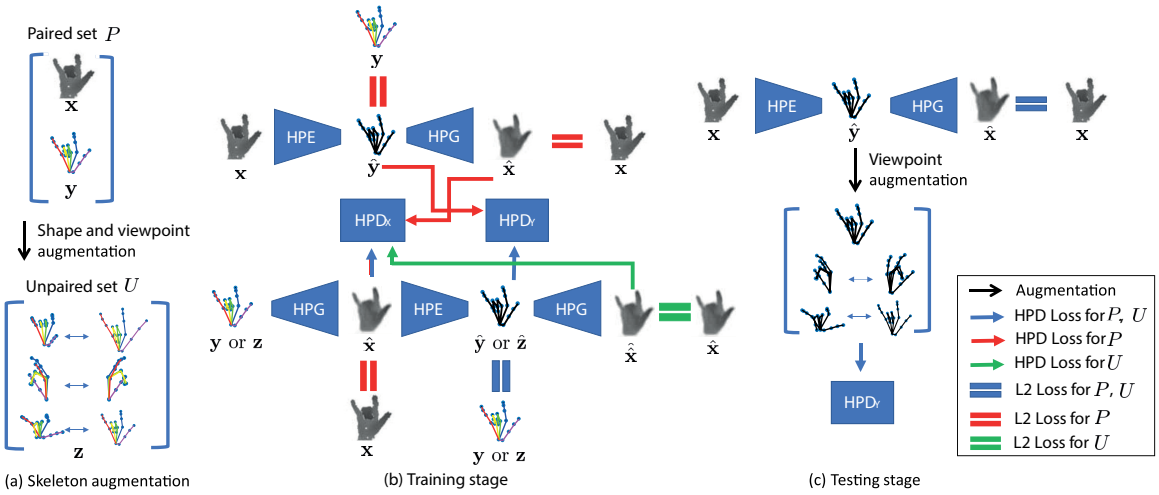


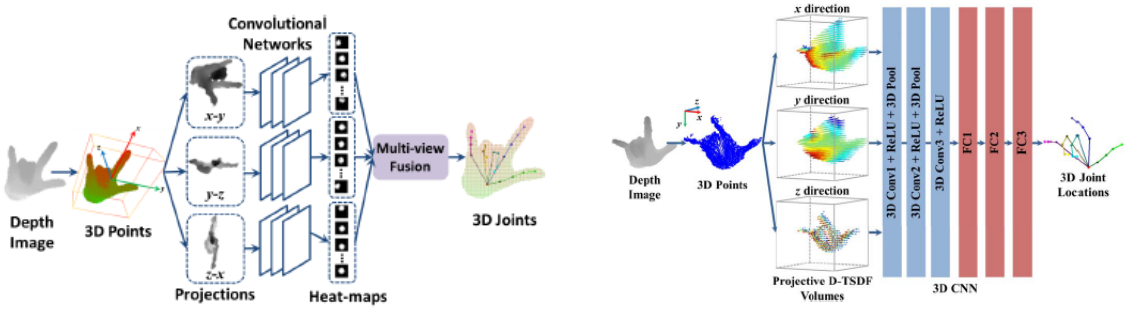
Figure 7: Baek *et al.* 's GAN based network architecture. In the diagram interaction with the paired set P and unpaired set U are represented by Red and Green respectively and the Blue lines is for interaction with both U and P. Originally used in [2].

shape to have a fixed and smaller number of points as the input. Also they applied their previous PCA analysis to rotate the 3D shape of the hand toward the principal components. Then they ran the hierarchical PointNet For 3D hand pose regression. In each step, this network downsampled the points. At last they used fully connected layers to regress the exact positions of hand joints in the space.

Ge *et al.* continued the same approach in [11] in which they changed the structure of their network. Therefore, instead of using multiple layers for downsampling the points, they used an architecture similar to encoder-decoder architecture. This structure first learns a global features and then using these global features it generates the desirable number of points used for estimating the position of joints.

Using a similar 2.5D depth map to 3D voxel-based hand shape convertor, in [22] Moon *et al.* designed a detection-based, voxel-to-voxel network (V2V) to directly estimate the position of each hand part based on the estimated 3D hand shape. Since the input and output are in 3D, they used 3D CNN which does all the convolution and deconvolution operations in 3D domain. The idea behind this paper is that depth maps taken from different angels of a single hand have the same 3D pose. To estimate hand pose via depth maps, it is needed to train the model to produce the same pose for different depth map inputs. On the other hand, a 3D point cloud has exactly one 3D hand pose and therefore their relation is one to one. So instead of having a huge dataset to cover all the shapes of a hand, we can train the model on the 3D point cloud of that hand and directly generate 3D pose via 3D encoder and decoders.

After the success of residual blocks of ResNet [14] in object classification, Moon *et al.* also used residual block with a deeper network. They applied their algorithm to



(a) (b)

Figure 8: Ge *et al.* 's work in [9] (a) and [10] (b). Originally used in [9] and [10] respectively

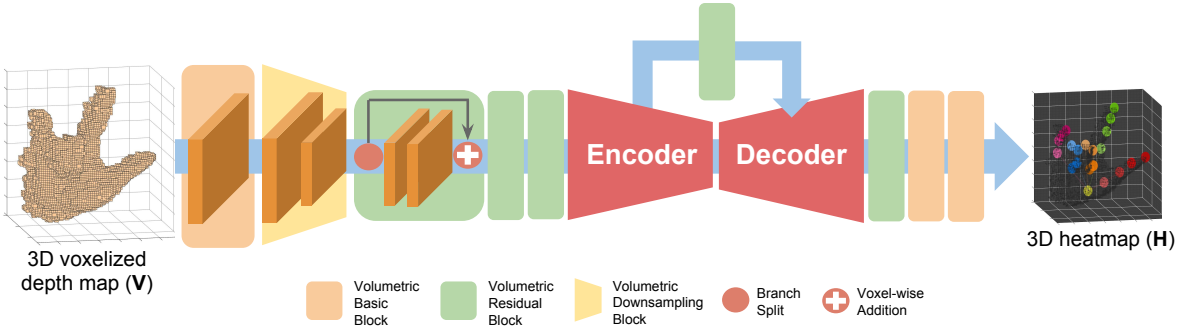


Figure 9: Architecture of V2V-PoseNet network using 3D CNN as encoder and decoder. Originally used in [22].

almost all the famous depth-based hand datasets and compared their method with a vast majority of algorithms and got the best result with a high margin compared to the others. One of the interesting facts about this algorithm is that, it can be easily applied to body pose estimation problem as well. They tested their algorithm on ITOP dataset [13] (both top-view and front-view) and reported their results. Figure 9 shows the architecture of V2V-PoseNet.

3.2 Image-based Methods

In spite of the fact that using a simple RGB image as an input gives the model a very good generalization power to be used everywhere, reducing the dimension of the input from 2.5D to 2D will make the task drastically harder. The data needed to train a network using RGB images is much bigger than the data needed to train a similar network using depth maps. Below we will first discuss the general approach used in the RGB-based networks, then we will discuss how the top tier image-based algorithms solved the high cost of annotating data and making a bigger datasets. Therefore most of the influential studies in this section came up with their own dataset.

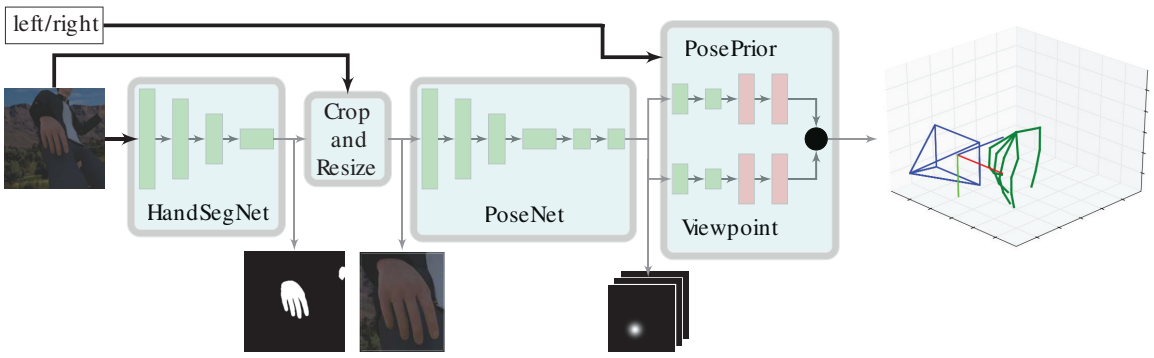


Figure 10: Architecture of PoseNet network using detection-based network *PoseNet* estimate 2D pose and regression-based network PosePrior to estimate the hand pose in 3D. Originally used in [59].

Note that image-based methods need to first isolate the hand (crop and resize) and then pass the cropped image to the network to estimate the pose. To do this, most of the image-based networks use a variations of *SegNet* [1] which is designed to segment a general picture (*e.g.* a street picture). Because of the binary classification (hand or background) and lack of variety in the input images, the hand segmentation is a relatively easier task than general segmentation problem. Therefore segmentation networks used in hand segmentation are more light weight than general SegNet and consequently faster to perform. In the following papers unless it is mentioned, one variation of SegNet is used for isolating the hands as a pre-processing step.

One of the important works in the RGB-based methods is Zimmermann *et al.* 's paper and dataset [59]. In this paper they used four different deep learning streams to make a 3D estimation of hand joints using a single RGB image. They first used a CNN called *HandSegNet* which is a light weight version of Wei *et al.* 's [51] human body detector trained on hand datasets.

As it is depicted in Figure 10, the output of *HandSegNet* is a mask picture showing the hand pixels. Based on this image, the hand is cropped and resized and is passed to the *PoseNet* network. PoseNet is a detection-based network which produces one probability density function (as a heatmap) for each hand joint. These predictions are in 2D and on the same coordinates of the input image. To get a 3D estimation, Zimmerman *et al.* used a network called *PosePrior* to convert these 2D predictions to 3D hand estimation. This network is a regression-based network and predicts the coordinates of the joints followed by a normalization. In this step, they normalize the distances of the joints considering the farthest distance as 1 and dividing the other distances to that number. Finally, they find a 3D rotation matrix such that a certain rotated keypoints is aligned with y-axis of the canonical frame.

As mentioned earlier, since the RGB image contains less information than depth disparity maps, the RGB-based networka are harder to train and requires a larger dataset. The most important issue in image-based method is the occlusion case, that

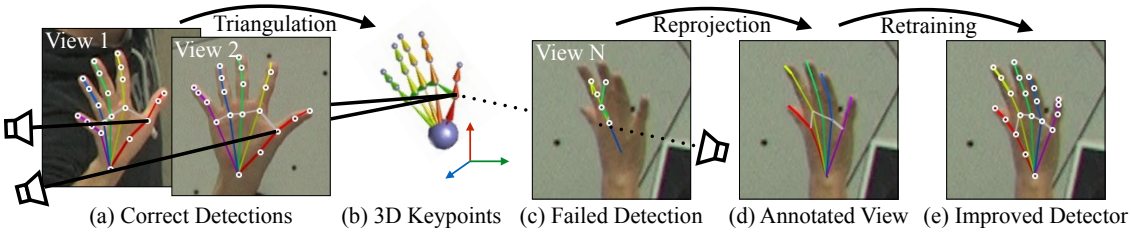


Figure 11: the triangulation, projection and retraining steps in the Simon *et al.* 's paper. Originally used in [36].

is when an object or the hand itself occludes some parts of the hand. The following two papers used two different innovative solutions to overcome this problem.

Inspired by Wei *et al.* 's approach to estimate body pose, in [36], Simon *et al.* used a multicamera approach to estimate hand pose. They used Carnegie Mellon University's Panoptic Studio [17] which contains more than 500 cameras (480 VGA and 30+ HD cameras) in a spherical space. They first trained a weak hand pose estimator using a synthesized dataset of hands. In the next step, they put a person in the center of the panoptic and applied the hand pose estimator on all the cameras recording video. The algorithm produces a (not very accurate) pose estimation for all of these views. It works in most of the views, but in the views in which the hand is occluded it does not work very well.

In the next part, which Simon *et al.* call triangulation step, they converted their 2D estimation to 3D estimation to evaluate their results, with knowing all the camera's intrinsic parameters and their relative physical position. To estimate the correct 3D pose, for each joint, they used RANSAC [6] algorithm to randomly select 2D views and convert them to 3D view. Then they keep the model with which most of the views agree. Finally, in a reverse operation, they projected the 3D view to the pictures and annotate that frame. In fact, instead of annotating data manually, they used multiple views to annotate their data. Then with this annotated dataset which is from multiple views, they trained the network again and made it more accurate. They repeated this process of annotation and training the model for three times and therefore they ended up with a very good and accurate model and dataset of annotated hand poses. For the feature extraction, they used a pre-trained VGG-19 network [37] (up to conv4_4) which produces a 128-channel feature. Figure 11 shows the triangulation, projection and retraining steps in the Simon *et al.* 's [36] paper.

To overcome the occlusion problem and especially the size of the dataset and the high cost of annotating hands frame by frame, Mueller *et al.* [23] used a synthesized dataset which is annotated automatically. They used kinematic sensors which has multiple electromagnetic sensors (usually 6 6D sensors) connected to a hand. These sensors are connected to a receiver and a transmitter which generates the 3D hand pose automatically.

Although using synthesized dataset is easy to generate and annotate, they lack

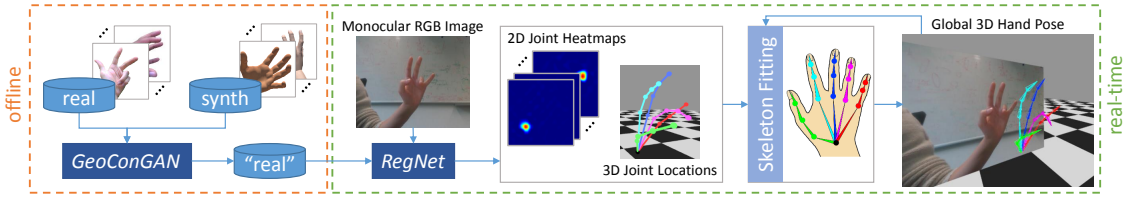


Figure 12: different steps of dataset production and hand pose estimation of Mueller *et al.* 's paper. Originally used in [23].

generalization power. As the image generated by this devices are computer generated, it will not work very well on real-world hand images. To overcome this issue they used a conditioned GAN [26] called *GeoConGAN* to transfer the computer generated images to real images. Also, to reach a better one-to-one relation between real and computer generated images, they applied a CyclicGAN [58] which has two parts of Real to Synthesized GAN (called *real2synth*) and Synthesized to Real GAN (called *synth2real*). Each of this GANs has its own generator and discriminator. Mueller *et al.* controlled the process with two losses; first converting synthesized image to real and calculating synth2real loss and again converting the result to synthesized image and calculating real2synth loss. They also randomly put some backgrounds behind the hands to make the images more realistic. Moreover, to create occlusion on the hands, they artificially put some objects in front of the hands to have some occluded frames in the dataset as well. They used ResNet [14] architecture for their feature extraction network to take advantage of residual blocks. Figure 12 shows the different steps of dataset production and hand pose estimation of Mueller *et al.* 's paper [23].

Spurr *et al.* [39] also used this cyclic concept for making a one-to-one relation between RGB image to 3D hand joints pose. They used GAN and Variational Autoencoder (VAE) to transfer the images to a latent space and then transfer it to the other domain. With that, they tried to map every RGB hand image to a 3D pose and use this map in the hand pose estimation task.

So far we only discussed algorithms which have using depth disparity maps or single RGB image. But there are studies in which RGBD images have been used (*i.e.* using both RGB image and its corresponding depth disparity map). Moreover some researchers used RGBD images during the training and used RGB while testing.

Dibra *et al.* in [5] designed a network which uses both RGB images and depth maps to estimate the hand pose. They used a specific network called *SynthNet* to estimate the hand pose which will be explained shortly. Next, with the generated 3D shape (3D shape of the whole hand not just the skeleton) they generate a depth map. They did the same process for depth map input as well. With the method explained above they generated a 3D model from the 2.5D depth map and then they calculated the loss of the algorithm from difference of two depth maps generated from these 3D models.

Unlike almost all the papers discussed so far, Dibra *et al.* did not use a joint model. In this paper they came up with a new method for hand pose estimation which unlike

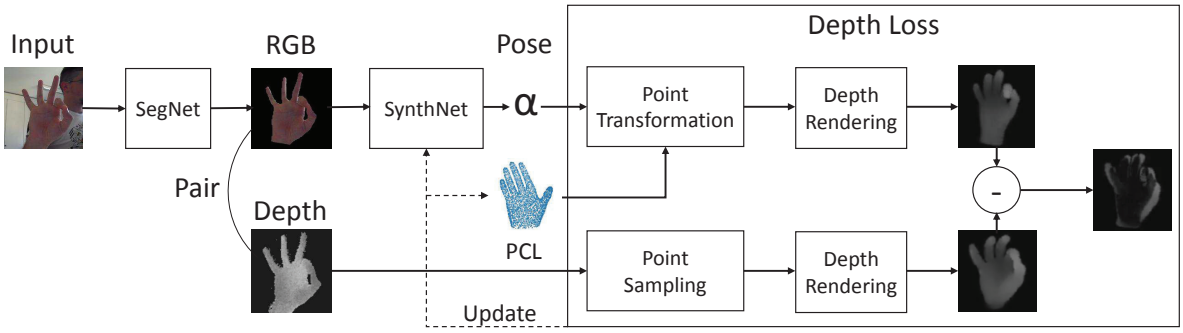


Figure 13: Different stages in Dibra *et al.* 's 3D hand pose estimation algorithm. Originally used in [5].

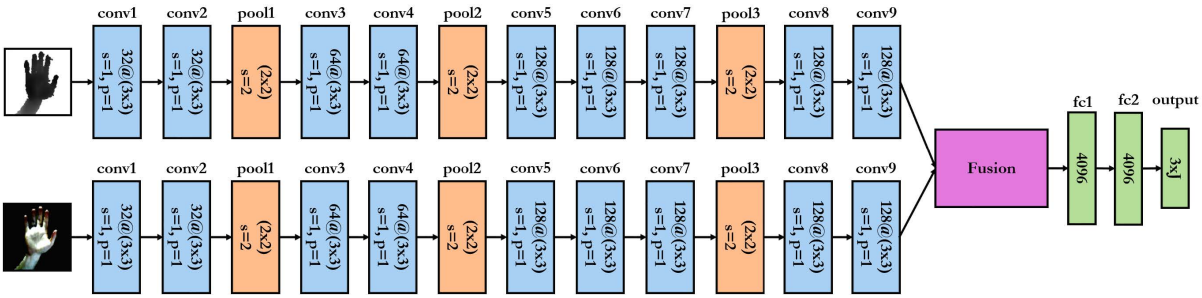


Figure 14: The two different streams in the architecture of *FuseNet*. Originally used in [18].

the other methods is not data-driven but uses information about human hand anatomy. So in their model they produce a hand shape which is physically possible (given all the possible cases of hand pose in prior). Figure 13 shows different stages in Dibra *et al.* 's 3D hand pose estimation algorithm.

Also there are models which used both depth disparity maps and RGB images. To make estimation more accurate, Kazakos *et al.* in [18] used two different deep learning streams for RGB and depth disparity maps called *FuseNet*. They used two identical convolutional neural networks for feature extraction of RGB and depth map images. The final prediction is generated from two fully connected layer which regresses (x, y, z) coordinates of each joint. Despite using two different streams, they results was not as good as the other approaches using one of these inputs. Figure 14 shows the two different streams in the architecture of *FuseNet*.

4 Datasets

In this section, we explain some of the most important datasets used in hand pose estimation and discuss their properties in detail. Also you can find the list of 20 hand

datasets in Table 5 which, to the best of our knowledge, is the most complete list of all the datasets in the hand pose estimation field.

4.1 ICVL Hand Dataset

The Imperial College Vision Lab (ICVL) dataset [44] is one of the oldest datasets in hand pose estimation field. It contains 180K annotated depth frames. They used 10 subjects to take 26 different poses. 16-joints model was used in annotating this dataset.

4.2 NYU Hand Dataset

New York University (NYU) dataset [47] contains 72,757 frames from a single actor in the train set and 8,252 frames from two different actors in test set. It is an RGBD dataset captured from side view (3rd person view). 36-joints model was used model to annotate this dataset.

4.3 HandNet Dataset

HandNet dataset [52] is one of the biggest depth datasets. It is generated using kinematic sensors with 10 different subjects, half male and half female to have different hand sizes in the dataset. It contains 202K frames in the trainning set and 10K frames in the test set. 6-joints model was used to annotate data.

4.4 CMU Panoptic Hand Dataset

Carnegie Mellon University (CMU) Panoptic [36] is RGB images which are recorded and annotated in the CMU's Panoptic studio. It contains both the synthesized and real images with 14,817 frames in 3rd person view which are annotated by 21-joints model in 2D.

4.5 BigHand 2.2M Benchmark Hand Dataset

The BigHand 2.2M Benchmark [56] is the biggest hand dataset so far. As we can see from its name it has, 2.2M annotated depth frames which are generated from 10 different subjects using kinematic six 6D electromagnetic sensors. Like all the recent datasets, it uses 21-joints model annotated in 3D. As the whole dataset was annotated with kinematic sensors, no object was held in the hands.

4.6 First-Person Hand Action dataset

The First-Person Hand Action dataset (FHAD) [7] is a new dataset introduced by The Imperial College. It contains 105,459 frames of egocentric view of 6 subjects doing 45

different types of activities in the kitchen, in the office or social activities. This dataset was also annotated in 3D, using 21-joints model.

4.7 GANerated Hand Dataset

GANerated is a new big dataset for the RGB-based dataset which has interaction with objects that can be helpful in estimating the hand pose under occlusion. It contains 330K frames synthesized hand shapes annotated in 3D using 21 joints model. In this dataset kinematic electromagnetic sensors was used to capture the hand pose. Also a CycleGAN was utilized to convert these computer generated images to look like real images. Different backgrounds was randomly put behind the hands to make them more similar to real photos. Also artificial objects was put on the hand to produce a hand occlusion.

The complete list of hand datasets with all their major properties is listed in Table 5.

5 Conclusion

In this report we defined the hand pose estimation problem and explained the major methods of solving this problem in detail. We also reviewed some of the recent applications of this field. Since every data driven method needs sufficient data in the first place, we talked about major datasets and listed all the datasets in this field with their most important properties. We showed how this field have grown in just a few years, from completely controlled situations with color gloves to 3D hand pose estimation using a single RGB image. Although the papers discussed here show good results on these dataset they do not get satisfactory results in the real world problems. Most importantly, the result of most of these systems is worst than a simple nearest-neighbor baseline [55]. However, because of the interests of big technology companies in this field, perhaps in the near future we see much bigger and more generalized datasets and therefore very well performing models even on a single RGB image. If we reach this technology, using an AR/VR device as our new PC, typing on the air and control objects in the display with our fingers will not be out of reach to .

Table 1: List of hand datasets with their properties

Dataset	Year	Synth./Real	RGB/D	Objects	#Joints	View	Ann.	#Subjects	#Frames (train/test)
GANerated [23]	2018	Synth.	RGB	Yes	21	Ego	2D+3D	-	330,000
FHAD [7]	2018	Real	RGB+D	Yes	21	Ego	3D	6	105,459
BigHand2.2M [56]	2017	Real	D	No	21	3rd	3D	10	2.2M
EgoDexter [24]	2017	Real	RGB+D	Yes	5	Ego	3D	4	1,485
SynthHands [24]	2017	Synth.	RGB+D	Both	21	Ego	3D	2	63,530
RHD [59]	2017	Synth.	RGB+D	No	21	3rd	3D	20	41k/2.7k
STB [57]	2017	Real	RGB+D	No	21	3rd	3D	-	18,000
CMU Panoptic [36]	2017	Both	RGB	No	21	3rd	2D	-	14,817
Graz16 [25]	2016	Real	RGB+D	Yes	21	Ego	2D+3D	6	2,166
Dexter+Object [41]	2016	Real	RGB+D	Yes	5	3rd	3D	2	3,014
EgoHands [3]	2015	Real	RGB	Yes	-	Ego	-	48	15,053
MSRA15 [43]	2015	Real	D	No	21	3rd	3D	9	76,375
MSRC [33]	2015	Synth.	D	No	-	-	3D	-	-
HandNet [52]	2015	Real	D	No	6	3rd	3D	10	202k/10k
NYU [47]	2014	Real	D	No	36	3rd	3D	2	72k/8k
ICVL [44]	2014	Real	D	No	16	3rd	3D	10	331k/1.5k
UCI-EGO [30]	2014	Real	RGB+D	No	26	Ego	3D	2	400
Hands in Action [48]	2014	Real	D	Yes	-	3rd	3D	-	-
MSRA14 [29]	2014	Real	D	No	21	3rd	3D	6	2,400
Dexter1 [42]	2013	Real	RGB+D	No	6	3rd	3D	1	2137
ASTAR [53]	2013	Real	D	No	20	3rd	3D	30	-

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[4] Hyung Jin Chang, Guillermo Garcia-Hernando, Danhang Tang, and Tae-Kyun Kim. Spatio-temporal hough forest for efficient detection–localisation–recognition of fingerwriting in egocentric camera. *Computer Vision and Image Understanding*, 148:87–96, 2016.

[5] Endri Dibra, Silvan Melchior, Ali Balkis, Thomas Wolf, Cengiz Oztireli, and Markus Gross. Monocular rgb hand pose inference from unsupervised refinable nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1075–1085, 2018.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2018.

[8] Liuhao Ge, Yujun Cai, Junwu Weng, and Junsong Yuan. Hand pointnet: 3d hand pose estimation using point sets. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

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

[10] 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.

[11] Liuhao Ge, Zhou Ren, and Junsong Yuan. Point-to-point regression pointnet for 3d hand pose estimation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Albert Haque, Boya Peng, Zelun Luo, Alexandre Alahi, Serena Yeung, and Li Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*, pages 160–177. Springer, 2016.

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

[15] Tianping Hu, Wenhai Wang, and Tong Lu. Hand pose estimation with attention-and-sequence network. In *Pacific Rim Conference on Multimedia*, pages 556–566. Springer, 2018.

[16] Youngkyoon Jang, Seung-Tak Noh, Hyung Jin Chang, Tae-Kyun Kim, and Woontack Woo. 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint. *IEEE Transactions on Visualization and Computer Graphics*, 21(4):501–510, 2015.

[17] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[18] Evangelos Kazakos, Christophoros Nikou, and Ioannis A Kakadiaris. On the fusion of rgb and depth information for hand pose estimation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 868–872. IEEE, 2018.

[19] 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.

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

[21] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 1073–1082. ACM, 2014.

[22] 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. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[23] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[24] 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)*, volume 10, 2017.

[25] Markus Oberweger, Gernot Riegler, Paul Wohlhart, and Vincent Lepetit. Efficiently creating 3d training data for fine hand pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4957–4965, 2016.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[27] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. User-defined gestures for augmented reality. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 955–960, New York, NY, USA, 2013. ACM.

[28] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[29] 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.

[30] Grégory Rogez, Maryam Khademi, JS Supančič III, Jose Maria Martinez Montiel, and Deva Ramanan. 3d hand pose detection in egocentric rgb-d images. In *Workshop at the European conference on computer vision*, pages 356–371. Springer, 2014.

[31] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Computer*

Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 1194–1201. IEEE, 2012.

[32] Toby Sharp. Implementing decision trees and forests on a gpu. In *European conference on computer vision*, pages 595–608. Springer, 2008.

[33] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3633–3642. ACM, 2015.

[34] Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[35] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1297–1304. Ieee, 2011.

[36] Tomas Simon, Hanbyul Joo, Iain A Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, volume 1, page 2, 2017.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014.

[38] Ayan Sinha, Chiho Choi, and Karthik Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4150–4158, 2016.

[39] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. Cross-modal deep variational hand pose estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. Investigating the dexterity of multi-finger input for mid-air text entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 3643–3652. ACM, 2015.

[41] Srinath Sridhar, Franziska Mueller, Michael Zollhöfer, Dan Casas, Antti Oulasvirta, and Christian Theobalt. Real-time joint tracking of a hand manipulating an object from rgb-d input. In *European Conference on Computer Vision*, pages 294–310. Springer, 2016.

[42] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE international conference on computer vision*, pages 2456–2463, 2013.

[43] 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.

[44] 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.

[45] Danhang Tang, Jonathan Taylor, Pushmeet Kohli, Cem Keskin, Tae-Kyun Kim, and Jamie Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *Proceedings of the IEEE international conference on computer vision*, pages 3325–3333, 2015.

[46] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proceedings of the IEEE international conference on computer vision*, pages 3224–3231, 2013.

[47] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*, 33(5):169, 2014.

[48] 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.

[49] Q. Wan. 2017 hand challenge fudan university team. In *Hands 2017*, 2017.

[50] Robert Y Wang and Jovan Popović. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3):63, 2009.

[51] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.

[52] Aaron Wetzler, Ron Slossberg, and Ron Kimmel. Rule of thumb: Deep derotation for improved fingertip detection. *arXiv preprint arXiv:1507.05726*, 2015.

[53] 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.

[54] Fang Yin, Xiujuan Chai, and Xilin Chen. Iterative reference driven metric learning for signer independent isolated sign language recognition. In *European Conference on Computer Vision*, pages 434–450. Springer, 2016.

[55] Shanxin Yuan, Guillermo Garcia-Hernando, Bjrn 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, and Tae-Kyun Kim. Depth-based 3d hand pose estimation: From current achievements to future goals. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[56] Shanxin Yuan, Qi Ye, Björn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2.2m benchmark: Hand pose dataset and state of the art analysis. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2605–2613. IEEE, 2017.

[57] Jiawei Zhang, Jianbo Jiao, Mingliang Chen, Liangqiong Qu, Xiaobin Xu, and Qingxiong Yang. A hand pose tracking benchmark from stereo matching. In *Image Processing (ICIP), 2017 IEEE International Conference on*, pages 982–986. IEEE, 2017.

[58] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, Oct 2017.

[59] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of International Conference on Computer Vision (ICCV)*, volume 10, 2017.