Survey on Hand Pose Estimation based on Single Depth Map

Tanay Deshmukh[1] and Jameel Malik[2]

[1] tdeshmuk@rhrk.uni-kl.de
[2] Jameel.Malik@dfki.de

Abstract. Hand Pose Estimation plays a significant role in the field of Human-Computer Interaction. Specifically for activity recognition with numerous applications in Augmented Reality and Virtual Reality. Hand Pose Estimation has been a topic of research for decades. The invention of deep learning and development of low-cost depth cameras, along with the huge investments of big Tech companies in this field, has given a boost to this research and broadened its application. It can be classified into two methods: Depth Based methods and Image-Based methods. Image-based methods use RGB images while depth-based methods use depth images. In this report, we focus on the approaches using a single depth image as input. We compare different approaches based on their architecture and state the positives and negatives of each approach. We further discuss some of the most widely used datasets for this problem. This is not a comprehensive review of the hand pose estimation techniques but a subset of the recent state-of-the-art methods.

Keywords: Hand Pose Estimation, Human Computer Interaction, Depth Input

1 Introduction

Accurate 3D hand pose estimation is one of the core technologies for human computer interaction in augmented reality and virtual reality applications. This problem as attracted considerable attraction in the computer vision community since the invention of deep learning and low cost depth cameras. Depth based 3D hand pose estimation can be categorized into generative, discriminative and hybrid methods. Generative models try to fit a predefined deformable hand model to depth images by minimizing hand crafted cost functions like Iterative Closest Point(ICP) [6, 8, 9, 14] Discriminative methods directly localize hand joints from input depth image. Previous discriminative approaches apply random forest and their variants. However, these approaches are limited by the hand crafted features and the recent developments of CNN based methods outperform them by far. Most of the existing methods employ a similar framework which takes a 2D depth map as input and directly regress the 3D co-ordinates of key-points, i.e. hand joints, via 2D CNNs. These method do not fully utilize the intrinsic 3D information given by the depth map thus distorting the actual shape of the hand. This forces the networks to perform perspective distortion-invariant estimation. Also, the process of directly regressing 3D key-points from a 2D image is highly non-linear making the learning process difficult.

To better utilize the depth information Ge et al.[2] propose to first project the input depth map onto three orthogonal planes and use the multi view projections to regress 2D heat maps which are further used to estimate the key-point positions on each plane. These heat maps are later fused to obtain the final 3D hand pose. Regressing the 3D joint locations accurately is the main difficulty that CNN based methods face. Direct mapping of 3D joint locations from input image is highly non-linear and makes the learning complex with low generalization ability of the network [15]. To overcome this difficulty, Ge et al.[2] use an alternative way, to map input image to a set of heat-maps which represent the probability distribution of joint positions thus making the learning process easier.

Multi-view CNNs can not fully exploit the 3d spatial information of the depth image since there is information loss while projection from 3D to 2D. Although, increasing the number of projection

can improve the accuracy, it will also increase the computational complexity. Ge et al. [3] propose a simple and effective approach to better utilize the depth information using 3D CNNs. The proposed architecture takes 3D volumetric representation of the hand depth image as input which captures the 3D spatial structure, and accurately regress full 3D hand pose in a single pass.

This approach how ever directly regresses the 3D joint locations making the learning process difficult. Ge et al.[2] used heat maps instead of direct;y regressing the 3D joint locations to make the learning process easy. However, the heat-maps in this method only provide 2D information and the depth information is not fully utilized. To overcome the problems of perspective distortion and non-linear mapping, Moon et al.[6] propose a novel voxel-to-voxel prediction network. They cast the 3D hand pose estimation problem from a single depth image to a voxel-to-voxel prediction that uses a 3D voxelized grid and estimates the per voxel likelihood for each key-point thus generating a 3D heat-map for every key-point as opposed to 2D heat-maps proposed by Ge et al.[2]

Below, We will first explain the Hand Pose Estimation Problem and discuss some of the existing challenges. Next, we will explain the above introduced methods in detail. At the end of the paper we will briefly investigate some of the most widely used data sets in this field.

2 Hand Pose Estimation Problem

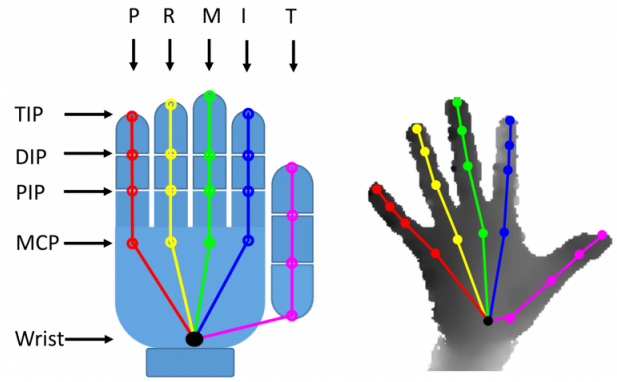

Fig. 1. Graphic representation of the hand joints and their naming convention [18].

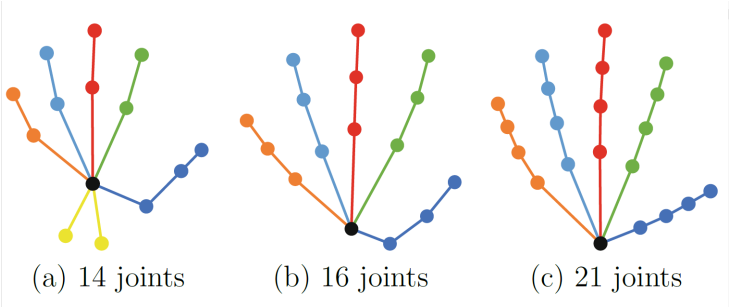

Fig. 2. Visualization of hand pose using 14, 16 and 21 joints as used in NYU [16], ICVL [13] and MSRA [8] datasets respectively. Image is taken from [1]

Hand Pose Estimation is the process of modelling human hand as set of some predefined number of key-points. These key-points usually represent the locations of the joints in the hand. The task of Hand Pose Estimation is to estimate the 3D location of these joints. The number of these predefined joints vary between different datasets. There is no global agreement on the number of joints to be used.

Figure 1 shows the actual mapping of the hand pose, superimposed on a depth image and a corresponding skeleton explaining the naming convention for each of the 21 joints

Figure 2 shows the the different number of joints in three well known hand datasets. The studies that want to compare the results for different number of joints have to change the architecture for every data-set. However, nowadays the model with 21 joints is the most widely used one.

3 Approaches

Regressing 3D landmarks directly from 2D depth map is a non-linear problem and does not utilize the intrinsic depth information present in the depth map. To better utilize the depth information Ge et al. [2] propose a novel regression method using multi view CNN. The goal behind using multi view CNNs is to learn the relation between the projected depth maps and the heat maps of each of the views. Firstly, the input depth image is converted to a set of 3D points in world-coordinate-system by using the depth cameras intrinsic parameters. Multi-view projections are obtained by projecting the 3D points onto three orthogonal planes.

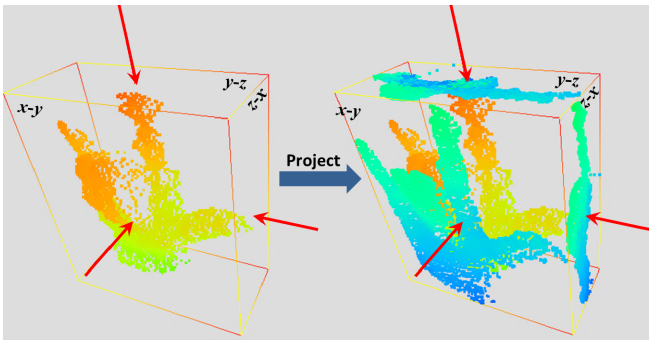

Fig. 3. 3D points are projected onto three orthogonal vertices of the OBB coordinate system [2]

Figure 4 shows the projection of 3d points onto three orthogonal planes of the Oriented Bounding Box(OBB). To obtain an OBB, first Principal Component Analysis (PCA) is performed on the of 3D points. Center of the bounding box is set to the origin of OBB co-ordinate system and the original axes are aligned with the principal components. Multi view projections are thus taken in OBB co-ordinate system. The distance between the plane and 3D point is normalized between 0 and 1, and is stored as the pixel value in the projected image. For each of the projections, a convolutional network is constructed having the same architecture. Inspired by the work of Tompson et al. [16], a multi-resolution CNN is employed for each view.

The projected image is resized to 96x96 pixels and the down-sampled to 48x4d and 24x24 pixels. These three images are then propagated through three different CNNs consisting to two convolutional stages ($Convolution + Relu + MaxPool$). Two stages of fully connected layer follow the convolutional layers to produce 21 heat-maps ($for 21 joints$) of 18x18 pixels indicating the confidence of a joint location for that specific view. The hand joint locations are estimated by applying MAP (Maximum a posterior) estimator based on the projections. Under the assumption that the

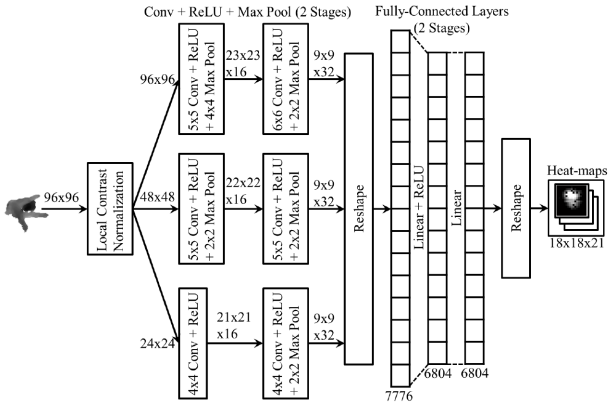

Fig. 4. 3D points are projected onto three orthogonal vertices of the OBB coordinate system [2]

three projections I_{xy}, I_{yz}, I_{xy} are independent and equal priori probability $P(\Phi)$, the posterior probability of joint locations can be calculated as the product of individual estimations from all the three views. The problem to estimate optimal joint locations Φ^* can be formulated as,

$$\begin{aligned}\Phi^* &= \underset{\Phi}{argmax} P(\Phi|I_{xy}, I_{yz}, I_{xy}) \\ &= \underset{\Phi}{argmax} P(\Phi|I_{xy})P(\Phi|I_{yz})P(\Phi|I_{zx})\end{aligned}$$

The posterior probabilities can be estimated from the heat-maps generated by each view. The final 3D joint locations are obtained by heat-map fusion.The network was trained using stochastic gradient descent as the optimization algorithm and Mean Squared Error as the loss function. The results are evaluated using two evaluation metrics, first is the standard, mean error distance for joints across the dataset. The second is the percentage of good frames. Good frames are those images where the estimated joint locations are below a predefined threshold value as compared to the ground truth.

Multi-view CNNs, however, do not optimally utilize the depth information. Some information is lost when the 3D points are projected onto three orthogonal planes. Although increasing the number of projections would yield better results, it will also increase the computational complexity. Also, the multi-view fusion acts as post processing step thus the hand pose can not be obtained in a single pass of the architecture.

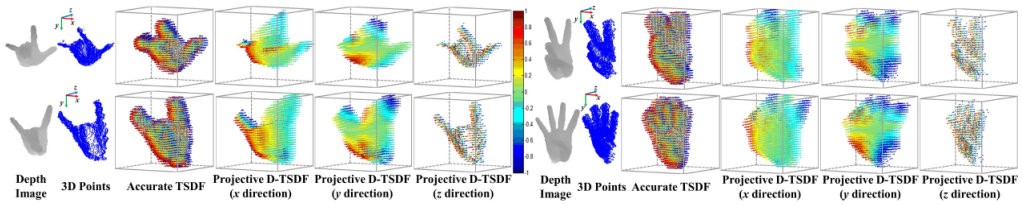

Fig. 5. Representation of the TSDF volumes [3]

To overcome these shortcomings, Ge et al. [3] propose to use a novel architecture using 3D CNN that can capture the 3D spatial structure of the hand. Similar to [2], the input depth map is first

converted to a set of 3D points. The obtained 3D points are encoded using 3D volumes storing the protective Truncated Signed Distance Function(TSDF) values. Figure 5 shows a visual comparison between accurate TSDF and projective D-TSDF. In accurate TSDF, each voxel stores the signed distance between the voxel center and the closet point point on the surface. This is ,however, computationally expensive as it requires the search of closest point among all surface points for each voxel. Projective TSDF is a better method for real-time considerations as it finds the closest point only in the line of sight in the camera frame. To better encode the volumetric representation, Ge et al. [3] apply projective Directional TSDF (D-TSDF) as proposed in [10]. Here, the euclidean distance is replaced by a 3D vector [*dx, dy, dz*] representing the distances in three directions according to the camera's co-ordinate system. The input 3D volume containing $M \times M \times M$ voxels is constructed using an axis-aligned bounding box(AABB), which is minimum bounding box with all the axes aligned to the axes of camera co-ordinate system. Edge length of the voxel is set as,

$$l_{voxel} = max l_x, l_y, l_z / M$$

where I_x, I_y and I_z are AABB's three edge lengths and M is the volume resolution. Truncation distance is set as $3 \times I_{voxel}$. Volume resolution is balanced with computational cost at 32.

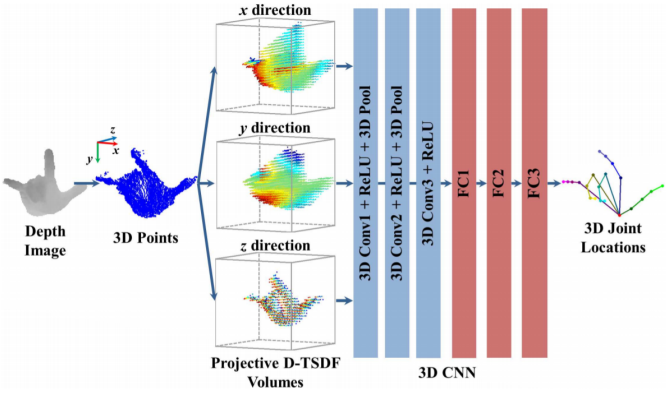

Fig. 6. Overview of the proposed method using 3D CNN [3]

Figure 6 gives an overview of the proposed method. A 3D CNN is trained in an end-to-end manner to map the 3D volumetric representation of the depth map to the relative 3D hand joint locations. The first three convolutional layers use kernels of size $5^3, 3^3$ and 3^3 respectively with stride 1 and no padding. The first two convolutional layers are followed by a 3D max-pooling layer with kernel size 2^3, stride 2 and no padding. The output of the third convolutional layer, i.e. feature extraction, is resized and passed to three fully-connected layers to map extracted features to 3D hand joint locations. Stochastic gradient descent is used to minimize the following objective function,

$$w^* = \underset{w}{argmin} \sum_{n=1}^{N} ||Y_n - F(V_n, w)||^2,$$

where w denotes the network weights, Y_n denotes the ground truth transformed into volumes co-ordinate system and normalized between 0 and 1. F represents the 3D CNN regressor and V_n

represents the volumetric representation of the depth image. In order to make the 3D CNN robust to the large variation in global orientations and hand sizes, 3D data augmentation is performed on the volumetric block which includes scaling and 3D rotations.

3D CNN's capture the spatial information of the input depth map. However, the major drawback of this approach is that it directly tries to regress the hand joint locations. It makes the learning process difficult. Ge et al.[2] use 2D heat maps to avoid this. However, this approach need to be followed by a fusion step which can again constitute to information loss along with increase in computation time and complexity. To overcome these issues, Moon et al. [6] propose another approach which uses a 3D voxelized depth map as input and estimates 3D heat-maps for every joint locations. Hand pose estimation problem is cast into a voxel to voxel prediction problem that estimates the per voxel likelihood of the hand joint locations.

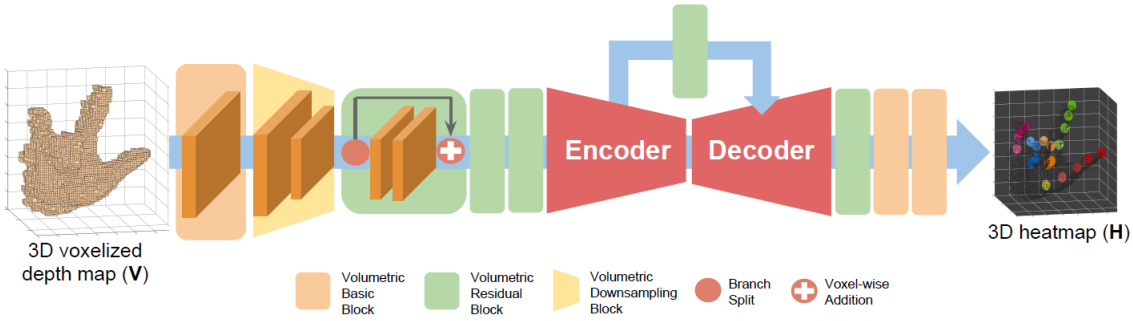

Fig. 7. Overview of the proposed V2V method [6]

The V2V PoseNet pipelines uses four kinds of building blocks. The first one is volumetric basic block located at the beginning and end of the network consisting of Volumetric convolution, volumetric batch normalization [5], and RELU as the activation function. Second is the volumetric residual block which is extended from the 2D residual block in [4]. Third is the volumetric downsampling block, similar to volumetric max pooling. Fourth is the volumetric upsampling block consisting of volumetric deconvolution layer, volumetric batch normalization, vloumetric batch normalization, and activation function(RELU). Batch normalization and activation function over the deconvolution layer ease the learning process. The complete architecture is based on the hour-glass model [7] and has been modified for better results. From figure 7, in encoder, the volumetric downsampling block reduces the spatial size of the obtained feature map and the volumetric residual block increase the number of channels. This increase in the number of channels increases the performance. In decoder, volumetric upsampling block enlarges the spatial size of the feature map while decreasing the number of channels. This helps the network to densly localize the keypoints. For the decoder to stably upsample the feature map, encoder and decoder are connected by a voxel wise addition on each scale. The upsampled feature maps are further passed through two $1 \times 1 \times 1$ volumetric basic blocks and one $1 \times 1 \times 1$ volumetric convolution layer. To supervise the per-voxel likelihood estimation, 3D heatmaps are generated for each of the keypoint with mean of the Gaussian peak positioned at the ground-truth location.

$$H_n^*(i,j,k) = exp\left(-\frac{(i-i_n)^2 + (j-j_n)^2 + (k-k_n)^2}{2\sigma^2}\right),$$

where H_n^* is the ground truth 3D heatmap of nth keypoint. For training, mean square error loss function L as follows is used,

$$L = \sum_{n=1}^{n} \sum_{i,j,k} ||H_n^*(i,j,k - H_n(i,j,k))||^2,$$

where H_n^* and H_n are the ground-truth and estimated heatmaps respectively. Evaluation metrics used are 3D distance error and the percentage of good frames. V2V-posenet has also been applied to Human Pose estimation.

4 Datastes

ICVL : The Imperial College Vision Lab (ICVL) dataset [13] contains 330K training and 1.6K testing annotated depth frames. The annotations follow a 16 joints model.
NYU : The New York University (NYU) dataset [16] contains 72K training and 8.2K testing depth images. The annotations follow a 36 joints model.
MSRA : Microsoft Research Asia dataset [11] contains a total of 76K annotated depth frames and it follows a 21 joints model.
BigHand2.2M : BigHand2.2M [19], as the name suggests contains 2.2M anotated depth frames and follows a 21 joints model.

5 Conclusion

In this survey we defined the Hand Pose Estimation problem and explained some of the best working methods in the past three years. Multi-View CNN by Ge et al. [2] published in 2016, using 3D CNN by Ge et al. [3] published in 2017 and the state-of-the-art method V2V-PoseNet by Moon et al. [6] in 2018. We explore the drawbacks of traditional methods as well as the early approaches using deep learning. We also discuss the drawbacks of each of the mentioned methods and the reason why [6] has been the stat-of-the-art method. Although the discussed methods show significant increase in performance, a previous [17, 12] study shows that simple nearest-neighbour baseline outperforms most existing systems. Even-though voxelizing the depth map shows good performance, the current existing methods perform well only on single hand pose estimation. Given the interest of research and vast applicability of the problem, experiencing a true VR/AR environment with seamless interaction of objects will not be out of reach.

References

1. Bardia Doosti. Hand pose estimation: A survey. *arXiv preprint arXiv:1903.01013*, 2019.
2. 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.
3. 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*, pages 1991–2000, 2017.
4. 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.
5. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
6. 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5088, 2018.
7. Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.
8. 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.
9. 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.
10. Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.
11. 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.
12. James Steven Supančič, Gregory Rogez, Yi Yang, Jamie Shotton, and Deva Ramanan. Depth-based hand pose estimation: methods, data, and challenges. *International Journal of Computer Vision*, 126(11):1180–1198, 2018.
13. 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.
14. 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.
15. Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.
16. 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.
17. Shanxin Yuan, Guillermo Garcia-Hernando, Björn Stenger, Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, Pavlo Molchanov, Jan Kautz, Sina Honari, Liuhao Ge, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2018.
18. 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.
19. Shanxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4866–4874, 2017.