

V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map

Gyeongsik Moon
ASRI, Seoul National University
mks0601@snu.ac.kr

Ju Yong Chang
Kwangwoon University
juyong.chang@gmail.com

Kyoung Mu Lee
ASRI, Seoul National University
kyoungmu@snu.ac.kr

Abstract

Most of the existing deep learning-based methods for 3D hand and human pose estimation from a single depth map are based on a common framework that takes a 2D depth map and directly regresses the 3D coordinates of keypoints, such as hand or human body joints, via 2D convolutional neural networks (CNNs). The first weakness of this approach is the presence of perspective distortion in the 2D depth map. While the depth map is intrinsically 3D data, many previous methods treat depth maps as 2D images that can distort the shape of the actual object through projection from 3D to 2D space. This compels the network to perform perspective distortion-invariant estimation. The second weakness of the conventional approach is that directly regressing 3D coordinates from a 2D image is a highly non-linear mapping, which causes difficulty in the learning procedure. To overcome these weaknesses, we firstly cast the 3D hand and human pose estimation problem from a single depth map into a voxel-to-voxel prediction that uses a 3D voxelized grid and estimates the per-voxel likelihood for each keypoint. We design our model as a 3D CNN that provides accurate estimates while running in real-time. Our system outperforms previous methods in almost all publicly available 3D hand and human pose estimation datasets and placed first in the HANDS 2017 frame-based 3D hand pose estimation challenge. The code is available in ¹.

1. Introduction

Accurate 3D hand and human pose estimation is an important requirement for activity recognition with diverse applications, such as human-computer interaction or augmented reality [33]. It has been studied for decades in computer vision community and has attracted considerable research interest again due to the introduction of low-cost depth cameras.

Recently, powerful discriminative approaches based on

¹https://github.com/mks0601/V2V-PoseNet_RELEASE

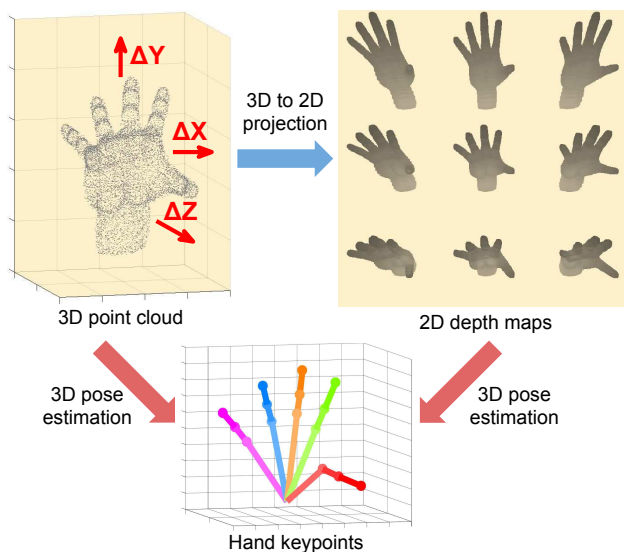


Figure 1: Visualization of perspective distortion in 2D depth image. The 3D point cloud has one-to-one relation with a 3D pose, but the 2D depth image has many-to-one relation because of perspective distortion. Thus, the network is compelled to perform perspective distortion-invariant estimation. The 2D depth maps are generated by translating the 3D point cloud by $\Delta X = -300, 0, 300$ mm (from left to right) and $\Delta Y = -300, 0, 300$ mm (from bottom to top). In all cases, ΔZ is set to 0 mm. Similar values to the real human hand size and camera projection parameters in the MSRA dataset were used for our visualization.

convolutional neural networks (CNNs) are outperforming existing methods in various computer vision tasks including 3D hand and human pose estimation from a single depth map [2, 10, 13, 15, 28]. Although these approaches achieved significant advancement in 3D hand and human pose estimation, they still suffer from inaccurate estimation because of severe self-occlusions, highly articulated shapes of target objects, and low quality of depth images. Analyzing previous deep learning-based methods for 3D hand and human pose estimation from a single depth image, most of these

methods [1, 2, 6, 13–15, 23, 28–30, 46] are based on a common framework that takes a 2D depth image and directly regresses the 3D coordinates of keypoints, such as hand or human body joints. However, we argue that this approach has two serious drawbacks. The first one is perspective distortion in 2D depth image. As the pixel values of a 2D depth map represent the physical distances of object points from the depth camera, the depth map is intrinsically 3D data. However, most previous methods simply take depth maps as a 2D image form, which can distort the shape of an actual object in the 3D space by projecting it to the 2D image space. Hence, the network *sees* a distorted object and is burdened to perform distortion-invariant estimation. We visualize the perspective distortions of the 2D depth image in Figure 1. The second weakness is the highly non-linear mapping between the depth map and 3D coordinates. This highly non-linear mapping hampers the learning procedure and prevents the network from precisely estimating the coordinates of keypoints as argued in [45]. This high nonlinearity is attributed to the fact that only one 3D coordinate for each keypoint has to be regressed from the input.

To cope with these limitations, we propose the *voxel-to-voxel prediction network for pose estimation (V2V-PoseNet)*. In contrast to most of the previous methods, the V2V-PoseNet takes a voxelized grid as input and estimates the per-voxel likelihood for each keypoint as in Figure 2.

By converting the 2D depth image into a 3D voxelized form as input, our network can *see* the actual appearance of objects without perspective distortion. Also, estimating the per-voxel likelihood of each keypoint enables the network to learn the desired task more easily than the highly non-linear mapping that estimates 3D coordinates directly from the input. We perform a thorough experiment to demonstrate the usefulness of the proposed volumetric representation of input and output in 3D hand and human pose estimation from a single depth map. The performance of the four combinations of input (i.e., 2D depth map and voxelized grid) and output (i.e., 3D coordinates and per-voxel likelihood) types are compared.

The experimental results show that the proposed voxel-to-voxel prediction allows our method to achieve the state-of-the-art performance in almost all of the publicly available datasets (i.e., three 3D hand [38, 40, 44] and one 3D human [15] pose estimation datasets) while it runs in real-time. We also placed first in the HANDS 2017 frame-based 3D hand pose estimation challenge [54]. We hope that the proposed system to become a milestone of 3D hand and human pose estimation problems from a single depth map. Now, we assume that the term “3D pose estimation” refers to the localization of the hand or human body keypoints in 3D space.

Our contributions can be summarized as follows.

- We firstly cast the problem of estimating 3D pose from

a single depth map into a voxel-to-voxel prediction. Unlike most of the previous methods that regress 3D coordinates directly from the 2D depth image, our proposed V2V-PoseNet estimates the per-voxel likelihood from a voxelized grid input.

- We empirically validate the usefulness of the volumetric input and output representations by comparing the performance of each input type (i.e., 2D depth map and voxelized grid) and output type (i.e., 3D coordinates and per-voxel likelihood).
- We conduct extensive experiments using almost all of the existing 3D pose estimation datasets including three 3D hand and one 3D human pose estimation datasets. We show that the proposed method produces significantly more accurate results than the state-of-the-art methods. The proposed method also placed first in the HANDS 2017 frame-based 3D hand pose estimation challenge.

2. Related works

Depth-based 3D hand pose estimation. Hand pose estimation methods can be categorized into generative, discriminative, and hybrid methods. Generative methods assume a pre-defined hand model and fit it to the input depth image by minimizing hand-crafted cost functions [34, 41]. Particle swarm optimization (PSO) [34], iterative closest point (ICP) [39], and their combination [32] are the common algorithms used to obtain optimal hand pose results.

Discriminative methods directly localize hand joints from an input depth map. Random forest-based methods [20, 22, 38, 40–42, 47] provide fast and accurate performance. However, they utilize hand-crafted features and are overcome by recent CNN-based approaches [1–3, 5, 6, 9, 10, 13, 14, 23, 28, 29, 36, 44, 49, 50] that can learn useful features by themselves. Tompson *et al.* [44] firstly utilized CNN to localize hand keypoints by estimating 2D heatmaps for each hand joint. Ge *et al.* [9] extended this method by exploiting multi-view CNN to estimate 2D heatmaps for each view. Ge *et al.* [10] transformed the 2D input depth map to the 3D form and estimated 3D coordinates directly via 3D CNN. Guo *et al.* [13, 14] proposed a region ensemble network to accurately estimate the 3D coordinates of hand keypoints and Chen *et al.* [2] improved this network by iteratively refining the estimated pose. Oberweger *et al.* [28] improved their preceding work [29] by utilizing recent network architecture, data augmentation, and better initial localization.

Hybrid methods are proposed to combine the generative and discriminative approach. Oberweger *et al.* [30] trained discriminative and generative CNNs by a feedback loop. Zhou *et al.* [57] pre-defined a hand model and estimated the parameter of the model instead of regressing 3D coordinates

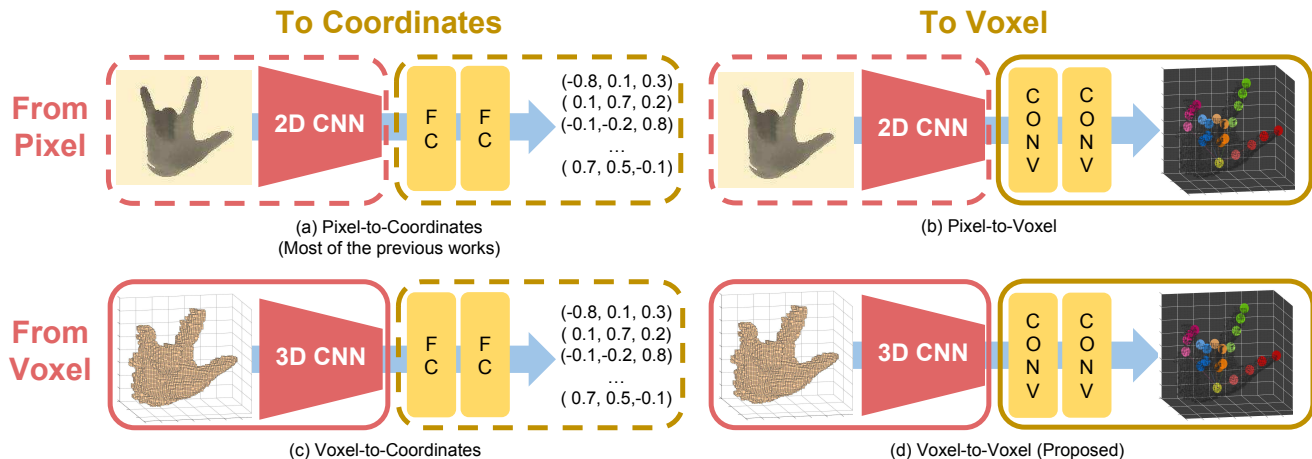


Figure 2: Various combinations of inputs and outputs for 3D pose estimation from a single depth image. Most of the previous works take a 2D depth image as input and estimate the 3D coordinates of keypoints as in (a). In contrast, the proposed system takes a 3D voxelized grid and estimates the per-voxel likelihood of each keypoint as in (d). Note that (b) and (d) are solely composed of the convolutional layers that become the fully convolutional architecture.

directly. Ye *et al.* [52] used spatial attention mechanism and hierarchical PSO. Wan *et al.* [46] used two deep generative models with a shared latent space and trained discriminator to estimate the posterior of the latent pose.

Depth-based 3D human pose estimation. Depth-based 3D human pose estimation methods also rely on generative and discriminative models. The generative models estimate the pose by finding the correspondences between the pre-defined body model and the input 3D point cloud. The ICP algorithm is commonly used for 3D body tracking [7, 12, 17, 21]. Another method such as template fitting with Gaussian mixture models [51] was also proposed. By contrast, the discriminative models do not require body templates and they directly estimate the positions of body joints. Conventional discriminative methods are mostly based on random forests. Shotton *et al.* [35] classified each pixel into one of the body parts, while Girchick *et al.* [11] and Jung *et al.* [19] directly regressed the coordinates of body joints. Jung *et al.* [56] used a random tree walk algorithm (RTW), which reduced the running time significantly. Recently, Haque *et al.* [15] proposed the viewpoint-invariant pose estimation method using CNN and multiple rounds of a recurrent neural network. Their model learns viewpoint-invariant features, which makes the model robust to viewpoint variations.

Volumetric representation using depth information. Wu *et al.* [48] introduced the volumetric representation of a depth image and surpassed the existing hand-crafted descriptor-based methods in 3D shape classification and retrieval. They represented each voxel as a binary random variable and used a convolutional deep belief network to learn the probability distribution for each voxel. Several recent works [25, 37] also represented 3D input data as a

volumetric form for 3D object classification and detection. Our work follows the strategy from [25], wherein several types of volumetric representation (i.e., occupancy grid models) were proposed to fully utilize the rich source of 3D information and efficiently deal with large amounts of point cloud data. Their proposed CNN architecture and occupancy grids outperform those of Wu *et al.* [48].

Input and output representation in 3D pose estimation. Most of the existing methods for 3D pose estimation from a single depth map [1, 2, 6, 13–15, 23, 28–30, 46] are based on the model in Figure 2(a) that takes a 2D depth image and directly regresses 3D coordinates. Recently, Ge *et al.* [10] and Deng *et al.* [5] converted a 2D depth image to a truncated signed distance function-based 3D volumetric form and directly regressed 3D coordinates as shown in Figure 2(c). In 3D human pose estimation from a RGB image, Pavlakos *et al.* [31] estimated the per-voxel likelihood for each body keypoint via 2D CNN as in the Figure 2(b). To estimate the per-voxel likelihood from an RGB image, they treated the discretized depth value as a channel of the feature map, which resulted in different kernels for each depth value. In contrast to all the above approaches, our proposed system estimates the per-voxel likelihood of each keypoint via the 3D fully convolutional network from the voxelized input as in Figure 2(d). To the best of our knowledge, our network is the first model to generate voxelized output from voxelized input using 3D CNN for 3D pose estimation.

3. Overview of the proposed model

The goal of our model is to estimate the 3D coordinates of all keypoints. First, we convert 2D depth images to 3D volumetric forms by reprojecting the points in the 3D space and discretizing the continuous space. After voxelizing the

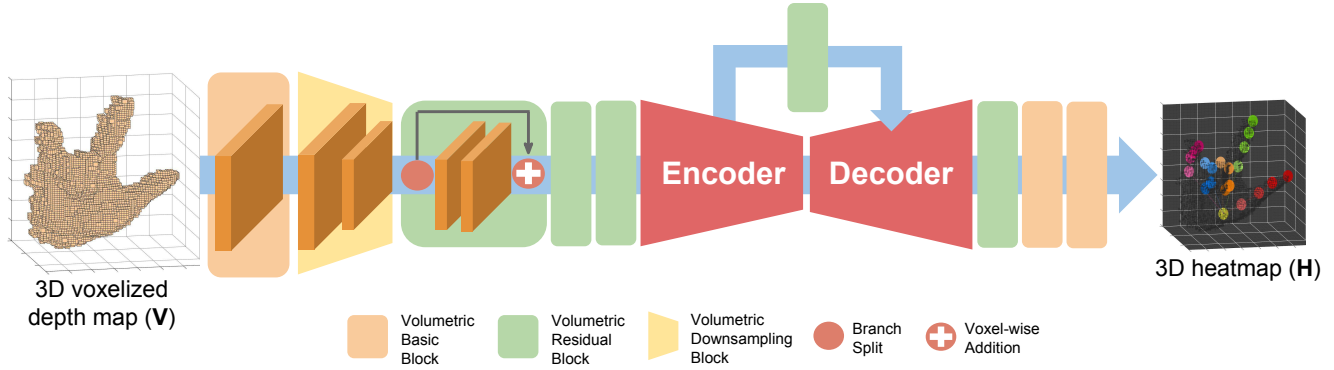


Figure 3: Overall architecture of the V2V-PoseNet. V2V-PoseNet takes voxelized input and estimates the per-voxel likelihood for each keypoint through encoder and decoder. To simplify the figure, we plotted each feature map without Z-axis and combined the 3D heatmaps of all keypoints in a single volume. Each color in the 3D heatmap indicates keypoints in the same finger.

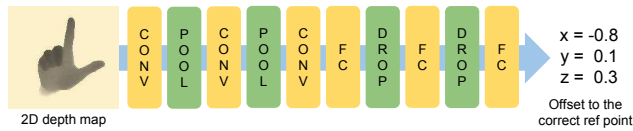


Figure 4: Reference point refining network. This network takes cropped depth image and outputs the 3D offset from the current reference point to the center of ground-truth joint locations.

2D depth image, the V2V-PoseNet takes the 3D voxelized data as an input and estimates the per-voxel likelihood for each keypoint. The position of the highest likelihood response for each keypoint is identified and warped to the real world coordinate, which becomes the final result of our model. Figure 3 shows the overall architecture of the proposed V2V-PoseNet. We now describe the target object localization refinement strategy, the process of generating the input of the proposed model, V2V-PoseNet, and some related issues of the proposed approach in the following sections.

4. Refining target object localization

To localize keypoints, such as hand or human body joints, a cubic box that contains the hand or human body in 3D space is a prerequisite. This cubic box is usually placed around the reference point, which is obtained using ground-truth joint position [29, 30, 57] or the center-of-mass after simple depth thresholding around the hand region [2, 13, 14]. However, utilizing the ground-truth joint position is infeasible in real-world applications. Also, in general, using the center-of-mass calculated by simple depth thresholding does not guarantee that the object is correctly contained in the acquired cubic box due to the error in the center-of-mass calculations in cluttered scenes. For example, if other objects are near the target object, then the simple depth thresholding method cannot properly filter the other objects be-

cause it applies the same threshold value to all input data. Hence, the computed center-of-mass becomes erroneous, which results in a cubic box that contains only some part of the target object. To overcome these limitations, we train a simple 2D CNN following Oberweger *et al.* [28] to obtain an accurate reference point as shown in Figure 4. This network takes a depth image, whose reference point is calculated by the simple depth thresholding around the hand region, and outputs 3D offset from the calculated reference point to the center of ground-truth joint locations. The refined reference point can be obtained by adding the output offset value of the network to the calculated reference point.

5. Generating input of the proposed system

To create the input of the proposed system, the 2D depth map should be converted to voxelized form. To voxelize the 2D depth map, we first reproject each pixel of the depth map to the 3D space. After reprojecting all depth pixels, the 3D space is discretized based on the pre-defined voxel size. Then, the target object is extracted by drawing the cubic box around the reference point obtained in Section 4. We set the voxel value of the network’s input $V(i, j, k)$ as 1 if the voxel is occupied by any depth point and 0 otherwise.

6. V2V-PoseNet

6.1. Building block design

We use four kinds of building blocks in designing the V2V-PoseNet. The first one is the *volumetric basic block* that consists of a volumetric convolution, volumetric batch normalization [18], and the activation function (i.e., ReLU). This block is located in the first and last parts of the network. The second one is the *volumetric residual block* extended from the 2D residual block of option B in [16]. The third one is the *volumetric downsampling block* that is iden-

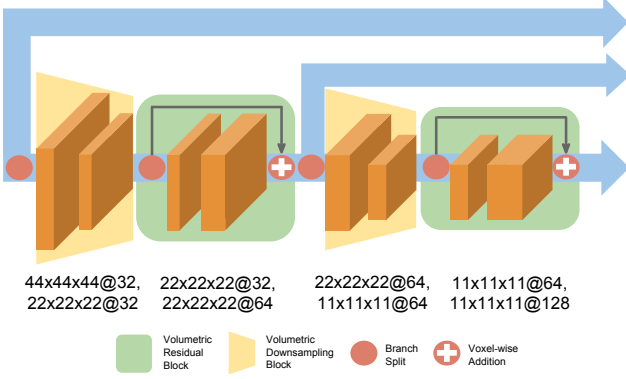


Figure 5: Encoder of the V2V-PoseNet. The numbers below each block indicate the spatial size and number of channels of each feature map. We plotted each feature map without Z-axis to simplify the figure.

tical to a volumetric max pooling layer. The last one is the *volumetric upsampling block*, which consists of a volumetric deconvolution layer, volumetric batch normalization layer, and the activation function (i.e., ReLU). Adding the batch normalization layer and the activation function to the deconvolution layer helps to ease the learning procedure. The kernel size of the residual blocks is $3 \times 3 \times 3$ and that of the downsampling and upsampling layers is $2 \times 2 \times 2$ with stride 2.

6.2. Network design

The V2V-PoseNet performs voxel-to-voxel prediction. Thus, it is based on the 3D CNN architecture that treats the Z-axis as an additional spatial axis so that the kernel shape is $w \times h \times d$. Our network architecture is based on the hourglass model [27], which was slightly modified for more accurate estimation. As the Figure 3 shows, the network starts from the $7 \times 7 \times 7$ volumetric basic block and the volumetric downsampling block. After downsampling the feature map, three consecutive residual blocks extract useful local features. The output of the residual blocks goes through the encoder and decoder described in Figures 5 and 6, respectively.

In the encoder, the volumetric downsampling block reduces the spatial size of the feature map while the volumetric residual block increases the number of channels. It is empirically confirmed that this increase in the number of channels helps improve the performance in our experiments. On the other hand, in the decoder, the volumetric upsampling block enlarges the spatial size of the feature map. When upsampling, the network decreases the number of channels to compress the extracted features. The enlargement of the volumetric size in the decoder helps the network to densely localize keypoints because it reduces the stride between voxels in the feature map. The encoder and decoder are connected with the voxel-wise addition for each

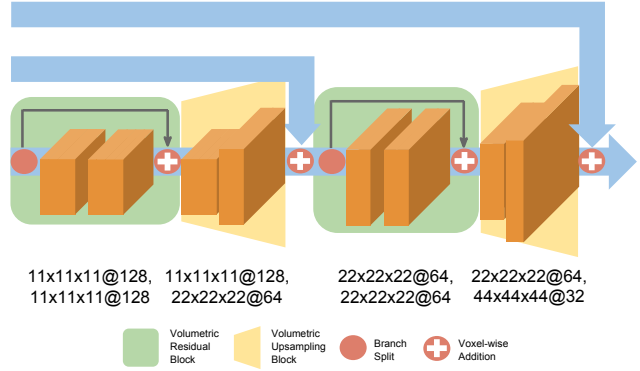


Figure 6: Decoder of the V2V-PoseNet. The numbers below each block indicate the spatial size and number of channels of each feature map. We plotted feature map without Z-axis to simplify the figure.

scale so that the decoder can upsample the feature map more stably. After passing the input through the encoder and decoder, the network predicts the per-voxel likelihood for each keypoint through two $1 \times 1 \times 1$ volumetric basic blocks and one $1 \times 1 \times 1$ volumetric convolutional layer.

6.3. Network training

To supervise the per-voxel likelihood for each keypoint, we generate 3D heatmap, wherein the mean of Gaussian peak is positioned at the ground-truth joint location as follows:

$$H_n^*(i, j, k) = \exp\left(-\frac{(i - i_n)^2 + (j - j_n)^2 + (k - k_n)^2}{2\sigma^2}\right), \quad (1)$$

where H_n^* is the ground-truth 3D heatmap of n th keypoint, (i_n, j_n, k_n) is the ground-truth voxel coordinate of n th keypoint, and $\sigma = 1.7$ is the standard deviation of the Gaussian peak.

Also, we adopt the mean square error as a loss function L as follows:

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

where H_n^* and H_n are the ground-truth and estimated heatmaps for n th keypoint, respectively, and N denotes the number of keypoints.

7. Implementation details

The proposed V2V-PoseNet is trained in an end-to-end manner from scratch. All weights are initialized from the zero-mean Gaussian distribution with $\sigma = 0.001$. Gradient vectors are calculated from the loss function and the weights are updated by the RMSProp [43] with a mini-batch size of 8. The learning rate is set to 2.5×10^{-4} . The size of the

input to the proposed system is $88 \times 88 \times 88$. We perform data augmentation including rotation ($[-40, 40]$ degrees in XY space), scaling ($[0.8, 1.2]$ in 3D space), and translation ($[-8, 8]$ voxels in 3D space). Our model is implemented by Torch7 [4] and the NVIDIA Titan X GPU is used for training and testing. We trained our model for 10 epochs.

8. Experiment

8.1. Datasets

ICVL Hand Posture Dataset. The ICVL dataset [40] consists of 330K training and 1.6K testing depth images. The frames are collected from 10 different subjects using Intel’s Creative Interactive Gesture Camera [26]. The annotation of hand pose contains 16 joints, which include three joints for each finger and one joint for the palm.

NYU Hand Pose Dataset. The NYU dataset [44] consists of 72K training and 8.2K testing depth images. The training set is collected from subject A, whereas the testing set is collected from subjects A and B by three Kinects from different views. The annotations of hand pose contain 36 joints. Most of the previous works only used frames from the frontal view and 14 out of 36 joints in the evaluation, and we also followed them.

MSRA Hand Pose Dataset. The MSRA dataset [38] contains 9 subjects with 17 gestures for each subject. Intel’s Creative Interactive Gesture Camera [26] captured 76K depth images with 21 annotated joints. For evaluation, the leave-one-subject-out cross-validation strategy is utilized.

HANDS 2017 Frame-based 3D Hand Pose Estimation Challenge Dataset. The HANDS 2017 frame-based 3D hand pose estimation challenge dataset [54] consists of 957K training and 295K testing depth images that are sampled from BigHand2.2M [55] and First-Person Hand Action [8] datasets. There are five subjects in the training set and ten subjects in the testing stage, including five unseen subjects. The ground-truth of this dataset is the 3D coordinates of 21 hand joints.

ITOP Human Pose Dataset. The ITOP dataset [15] consists of 40K training and 10K testing depth images for each of the front-view and top-view tracks. This dataset contains depth images with 20 actors who perform 15 sequences each and is recorded by two Asus Xtion Pro cameras. The ground-truth of this dataset is the 3D coordinates of 15 body joints.

8.2. Evaluation metrics

We used 3D distance error and percentage of success frame metrics for 3D hand pose estimation following [38, 40]. For 3D human pose estimation, we used mean average precision (mAP) that is defined as the detected ratio of all human body joints based on 10 cm rule following [15, 56].

Input \ Output	3D Coordinates	Per-voxel likelihood
2D depth map	18.85 (21.1 M)	13.01 (4.6 M)
3D voxelized grid	16.78 (457.5 M)	10.37 (3.4 M)

Table 1: Average 3D distance error (mm) and number of parameter comparison of the input and output types in the NYU dataset. The number in the parenthesis denotes the number of parameters. The visualized model for each input and output type is shown in Figure 2.

Methods	Average 3D distance error
Baseline	11.14 mm
+ Localization refinement	9.22 mm
+ Epoch ensemble	8.42 mm

Table 2: Effect of localization refinement and epoch ensemble. The average 3D distance error is calculated in the NYU dataset.

8.3. Ablation study

We used NYU hand pose dataset [44] to analyze each component of our model because this dataset is challenging and far from saturated.

3D representation and per-voxel likelihood estimation. To demonstrate the validity of the 3D representation of the input and per-voxel likelihood estimation, we compared the performances of the four different combinations of the input and output forms in Table 1. As the table shows, converting the input representation type from the 2D depth map to 3D voxelized form (also converting the model from 2D CNN to 3D CNN) substantially improves performance, regardless of output representation. This justifies the effectiveness of the proposed 3D input representation that is free from perspective distortion. The results also show that converting the output representation from the 3D coordinates to the per-voxel likelihood increases the performance significantly, regardless of the input type. Among the four combinations, *voxel-to-voxel* gives the best performance even with the smallest number of parameters. Hence, the superiority of the *voxel-to-voxel* prediction scheme compared with other input and output combinations is clearly justified.

To fairly compare four combinations, we used the same network building blocks and design, which were introduced in Section 6. The only difference is that the model for the per-voxel likelihood estimation is fully convolutional, whereas for the coordinate regression, we used fully connected layers at the end of the network. Simply converting *voxel-to-voxel* to *pixel-to-voxel* decreases the number of parameters because the model is changed from the 3D CNN to the 2D CNN. To compensate for this change, we doubled the number of channels of each feature map in the *pixel-to-voxel* model. If the number of channels is not doubled, then the performance was degraded. For all four models,

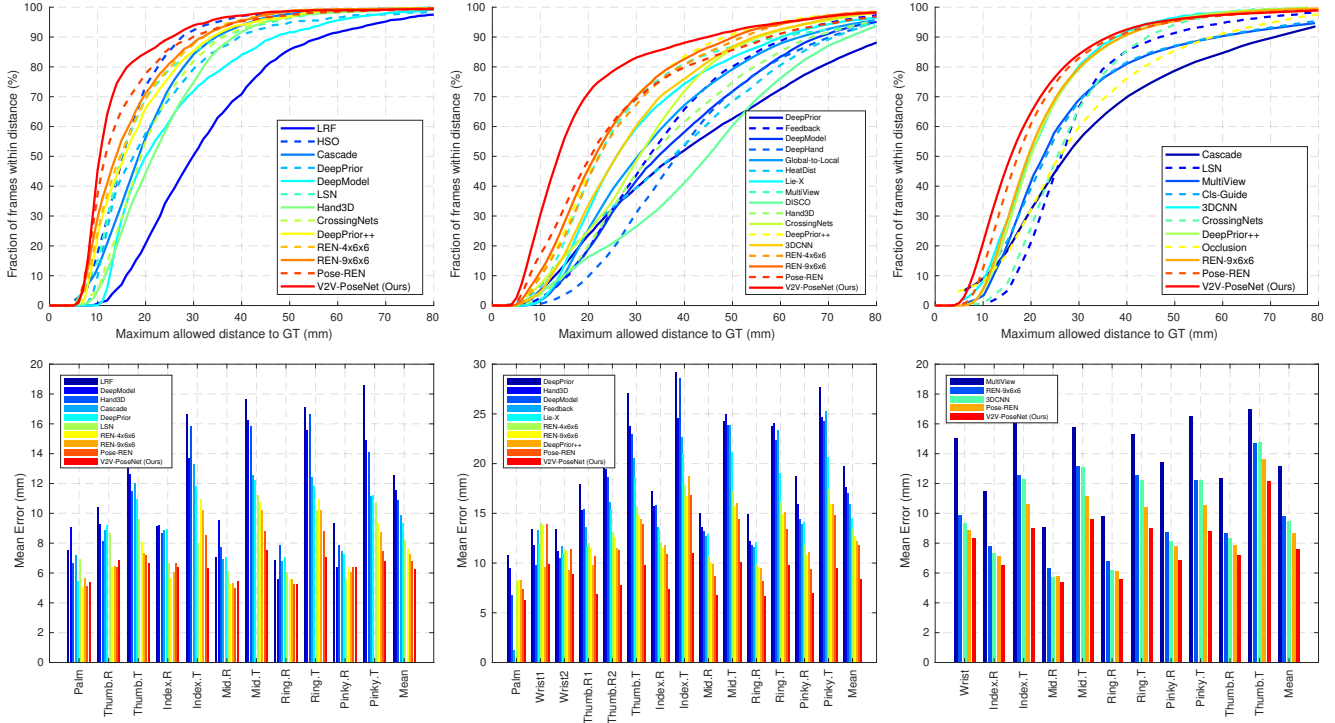


Figure 7: Comparison of the proposed method (V2V-PoseNet) with state-of-the-art methods. Top row: the percentage of success frames over different error thresholds. Bottom row: 3D distance errors per hand keypoints. Left: ICVL dataset, middle: NYU dataset, right: MSRA dataset.

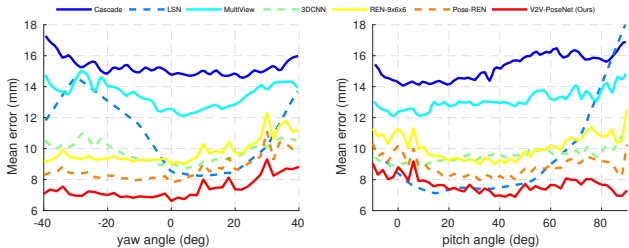


Figure 8: Comparison of average 3D distance error over different yaw (left) and pitch (right) angles on the MSRA dataset.

we used 48×48 depth map or $48 \times 48 \times 48$ voxelized grid as input because the original size ($88 \times 88 \times 88$) does not fit into GPU memory in the case of *voxel-to-coordinates*.

Refining localization of the target object. To demonstrate the importance of the localization refining procedure in Section 4, we compared the performance of two with and without the localization refinement step. As shown in Table 2, the refined reference points significantly boost the accuracy of our model, which shows that the reference point refining procedure has a crucial influence on the performance.

Epoch ensemble. To obtain more accurate and robust estimation, we applied a simple ensemble technique that we call *epoch ensemble*. The epoch ensemble averages the estimations from several epochs. Specifically, we save the

trained model for each epoch in the training stage and then in the testing stage, we average all the estimated 3D coordinates from the trained models. As we trained our model by 10 epochs, we used 10 models to obtain the final estimation. Epoch ensemble has no influence in running time when each model is running in different GPUs. However, in a single-GPU environment, epoch ensemble linearly increases running time. The effect of epoch ensemble is shown in Table 2.

8.4. Comparison with state-of-the-art methods

We compared the performance of the V2V-PoseNet on the three 3D hand pose estimation datasets (ICVL [40], NYU [44], and MSRA [38]) with most of the state-of-the-art methods, which include latent random forest (LRF) [40], cascaded hand pose regression (Cascade) [38], DeepPrior with refinement (DeepPrior) [29], feedback loop training method (Feedback) [30], hand model based method (DeepModel) [57], hierarchical sampling optimization (HSO) [41], local surface normals (LSN) [47], multi-view CNN (MultiView) [9], DISCO [1], Hand3D [5], DeepHand [36], lie-x group based method (Lie-X) [49], improved DeepPrior (DeepPrior++) [28], region ensemble network (REN- $4 \times 6 \times 6$ [14], REN- $9 \times 6 \times 6$ [13]), CrossingNets [46], pose-guided REN (Pose-REN) [2], global-to-local prediction method (Global-to-Local) [23], classification-guided approach (Cls-Guide) [50], 3DCNN

Team name	Average 3D distance error
V2V-PoseNet (Ours)	9.95 mm
NVResearch and UMontreal	10.18 mm
NTU	11.30 mm
THU VCLab	11.70 mm
NAIST RVLab	11.90 mm

Table 3: The top-5 results of the HANDS 2017 frame-based 3D hand pose estimation challenge.

based method (3DCNN) [10], occlusion aware based method (Occlusion) [24], and hallucinating heat distribution method (HeatDist) [3]. Some reported results of previous works [2, 13, 14, 28–30, 40, 49, 57] are calculated by prediction labels available online. Other results [1, 3, 5, 9, 10, 23, 24, 36, 38, 41, 46, 47, 50] are calculated from the figures and tables of their papers.

As shown in Figure 7, our method outperforms all existing methods on the three 3D hand pose estimation datasets in standard evaluation metrics. This shows the superiority of *voxel-to-voxel* prediction, which is firstly used in 3D hand pose estimation. The performance gap between ours and the previous works is largest on the NYU dataset that is very challenging and far from saturated. We additionally measured the average 3D distance error distribution over various yaw and pitch angles on the MSRA dataset following the protocol of previous works [38] as in Figure 8. As it demonstrates, our method provides superior results in almost all of yaw and pitch angles.

Our method also placed first in the HANDS 2017 frame-based 3D hand pose estimation challenge [54]. The top-5 results comparisons are shown in Table 3. As shown in the table, the proposed V2V-PoseNet outperforms other participants. A more detailed analysis of the challenge results is covered in [53].

We also evaluated the performance of the proposed system on the ITOP 3D human pose estimation dataset [15]. We compared the system with state-of-the-art works, which include RTW [56], viewpoint-invariant feature-based method [15], and REN-9x6x6 [13]. The score of each method is obtained from [13, 15]. As shown in Table 4, the proposed system outperforms all the existing methods by a large margin in both of views, which indicates that our model can be applied to not only 3D hand pose estimation, but also other challenging problems such as 3D human pose estimation from the front- and top-views.

8.5. Computational complexity

We investigated the computational complexity of the proposed method. The training time of the V2V-PoseNet is two days for ICVL dataset, 12 hours for NYU and MSRA datasets, six days for HANDS 2017 challenge dataset, and three hours for ITOP dataset. The testing time is 3.5 fps when using 10 models for epoch ensemble, but can acceler-

Body part	mAP (front-view)				mAP (top-view)			
	[56]	[15]	[13]	V2V-PoseNet (Ours)	[56]	[15]	[13]	V2V-PoseNet (Ours)
Head	97.8	98.1	98.7	98.29	98.4	98.1	98.2	98.4
Neck	95.8	97.5	99.4	99.07	82.2	97.6	98.9	98.91
Shoulders	94.1	96.5	96.1	97.18	91.8	96.1	96.6	96.87
Elbows	77.9	73.3	74.7	80.42	80.1	86.2	74.4	79.16
Hands	70.5	68.7	55.2	67.26	76.9	85.5	50.7	62.44
Torso	93.8	85.6	98.7	98.73	68.2	72.9	98.1	97.78
Hip	80.3	72	91.8	93.23	55.7	61.2	85.5	86.91
Knees	68.8	69	89	91.80	53.9	51.6	70	83.28
Feet	68.4	60.8	81.1	87.6	28.7	51.5	41.6	69.62
Mean	80.5	77.4	84.9	88.74	68.2	75.5	75.5	83.44

Table 4: mAP comparison of state-of-the-art methods on front and top views of the ITOP dataset.

ate to 35 fps in a multi-GPU environment, which shows the applicability of the proposed method to real-time applications. The most time-consuming step is the input generation that includes reference point refinement and voxelizing the depth map. This step takes 23 ms and most of the time is spent on voxelizing. The next step is network forwarding, which takes 5 ms and takes 0.5 ms to extract 3D coordinates from the 3D heatmap. Note that our model outperforms previous works by a large margin without epoch ensemble on the ICVL, NYU, MSRA, and ITOP datasets while running in real-time using a single GPU.

9. Conclusion

We proposed a novel and powerful network, V2V-PoseNet, for 3D hand and human pose estimation from a single depth map. To overcome the drawbacks of previous works, we converted 2D depth map into the 3D voxel representation and processed it using our 3D CNN model. Also, instead of directly regressing 3D coordinates of keypoints, we estimated the per-voxel likelihood for each keypoint. Those two conversions boost the performance significantly and make the proposed V2V-PoseNet outperform previous works on the three 3D hand and one 3D human pose estimation datasets by a large margin. It also allows us to achieve the highest accuracy on the 3D hand pose estimation challenge dataset. As *voxel-to-voxel* prediction is firstly tried in 3D hand and human pose estimation from a single depth map, we hope this work to provide a new way of accurate 3D pose estimation.

Acknowledgments

This work was supported by the Visual Turing Test project (IITP-2017-0-01780) from the Ministry of Science and ICT of Korea.

References

- [1] D. Bouchacourt, P. K. Mudigonda, and S. Nowozin. Disco nets: Dissimilarity coefficients networks. In *Advances in Neural Information Processing Systems*, pages 352–360, 2016.
- [2] X. Chen, G. Wang, H. Guo, and C. Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *arXiv preprint arXiv:1708.03416*, 2017.
- [3] C. Choi, S. Kim, and K. Ramani. Learning hand articulations by hallucinating heat distribution. In *IEEE International Conference on Computer Vision*, Oct 2017.
- [4] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, Neural Information Processing Systems Workshop*, number EPFL-CONF-192376, 2011.
- [5] X. Deng, S. Yang, Y. Zhang, P. Tan, L. Chang, and H. Wang. Hand3d: Hand pose estimation using 3d neural network. *arXiv preprint arXiv:1704.02224*, 2017.
- [6] D. Fourure, R. Emonet, E. Fromont, D. Muselet, N. Neverova, A. Trémeau, and C. Wolf. Multi-task, multi-domain learning: application to semantic segmentation and pose regression. *Neurocomputing*, 251:68–80, 2017.
- [7] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time human pose tracking from range data. In *European Conference on Computer Vision*, pages 738–751. Springer, 2012.
- [8] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. *arXiv preprint arXiv:1704.02463*, 2017.
- [9] L. Ge, H. Liang, J. Yuan, and D. Thalmann. Robust 3d hand pose estimation in single depth images: from single-view cnn to multi-view cnns. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3593–3601, 2016.
- [10] L. Ge, H. Liang, J. Yuan, and D. Thalmann. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [11] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *IEEE International Conference on Computer Vision*, pages 415–422. IEEE, 2011.
- [12] D. Grest, J. Woetzel, and R. Koch. Nonlinear body pose estimation from depth images. In *Joint Pattern Recognition Symposium*, pages 285–292. Springer, 2005.
- [13] H. Guo, G. Wang, X. Chen, and C. Zhang. Towards good practices for deep 3d hand pose estimation. *arXiv preprint arXiv:1707.07248*, 2017.
- [14] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yand. Region ensemble network: Improving convolutional network for hand pose estimation. *IEEE International Conference on Image Processing*, 2017.
- [15] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards viewpoint invariant 3d human pose estimation. In *European Conference on Computer Vision*, pages 160–177. Springer, 2016.
- [16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [17] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt. Personalization and evaluation of a real-time depth-based full body tracker. In *IEEE International Conference on 3D Vision*, pages 279–286. IEEE, 2013.
- [18] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.
- [19] H. Y. Jung, Y. Suh, G. Moon, and K. M. Lee. A sequential approach to 3d human pose estimation: Separation of localization and identification of body joints. In *European Conference on Computer Vision*, pages 747–761. Springer, 2016.
- [20] C. Keskin, F. Kırac, Y. E. Kara, and L. 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.
- [21] S. Knoop, S. Vacek, and R. Dillmann. Sensor fusion for 3d human body tracking with an articulated 3d body model. In *IEEE International Conference on Robotics and Automation*, pages 1686–1691. IEEE, 2006.
- [22] H. Liang, J. Yuan, and D. Thalmann. Parsing the hand in depth images. *IEEE Transactions on Multimedia*, 16(5):1241–1253, 2014.
- [23] M. Madadi, S. Escalera, X. Baro, and J. Gonzalez. End-to-end global to local cnn learning for hand pose recovery in depth data. *arXiv preprint arXiv:1705.09606*, 2017.
- [24] M. Madadi, S. Escalera, A. Carruesco, C. Andujar, X. Baró, and J. González. Occlusion aware hand pose recovery from sequences of depth images. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 230–237. IEEE, 2017.
- [25] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE International Conference on Intelligent Robots and Systems*, pages 922–928. IEEE, 2015.
- [26] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013.
- [27] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.
- [28] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *IEEE International Conference on Computer Vision Workshop*, Oct 2017.
- [29] M. Oberweger, P. Wohlhart, and V. Lepetit. Hands deep in deep learning for hand pose estimation. *Computer Vision Winter Workshop*, pages 21–30, 2015.
- [30] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *IEEE International Conference on Computer Vision*, pages 3316–3324, 2015.
- [31] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d hu-

- man pose. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.
- [32] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1106–1113, 2014.
- [33] J. Romero, H. Kjellström, and D. Kragic. Monocular real-time 3d articulated hand pose estimation. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 87–92. IEEE, 2009.
- [34] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. 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.
- [35] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.
- [36] A. Sinha, C. Choi, and K. Ramani. Deephand: Robust hand pose estimation by completing a matrix imputed with deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4150–4158, 2016.
- [37] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.
- [38] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015.
- [39] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly. Robust articulated-icp for real-time hand tracking. In *Computer Graphics Forum*, volume 34, pages 101–114. Wiley Online Library, 2015.
- [40] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3786–3793, 2014.
- [41] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *IEEE International Conference on Computer Vision*, pages 3325–3333, 2015.
- [42] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *IEEE International Conference on Computer Vision*, pages 3224–3231, 2013.
- [43] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.
- [44] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169, 2014.
- [45] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *Advances in Neural Information Processing Systems*, pages 1799–1807, 2014.
- [46] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.
- [47] C. Wan, A. Yao, and L. Van Gool. Hand pose estimation from local surface normals. In *European Conference on Computer Vision*, pages 554–569. Springer, 2016.
- [48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.
- [49] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng. Lie-x: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision*, pages 1–25, 2017.
- [50] H. Yang and J. Zhang. Hand pose regression via a classification-guided approach. In *Asian Conference on Computer Vision*, pages 452–466. Springer, 2016.
- [51] M. Ye and R. Yang. Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2345–2352, 2014.
- [52] Q. Ye, S. Yuan, and T.-K. 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.
- [53] S. Yuan, G. Garcia-Hernando, B. Stenger, T.-K. Kim, et al. Depth-based 3d hand pose estimation: From current achievements to future goals. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.
- [54] S. Yuan, Q. Ye, G. Garcia-Hernando, and T.-K. Kim. The 2017 hands in the million challenge on 3d hand pose estimation. *arXiv preprint arXiv:1707.02237*, 2017.
- [55] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. Big-hand2.2m benchmark: Hand pose dataset and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017.
- [56] H. Yub Jung, S. Lee, Y. Seok Heo, and I. Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2015.
- [57] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. *International Joint Conference on Artificial Intelligence*, pages 2421–2427, 2016.