Weakly-supervised hand part segmentation from depth images

Mohammad Rezaei
The University of Texas at Arlington
Arlington, Texas, USA
mohammad.rezaei@mavs.uta.edu

Farnaz Farahanipad
The University of Texas at Arlington
Arlington, Texas, USA
farnaz.farahanipad@mavs.uta.edu

Alex Dillhoff
The University of Texas at Arlington
Arlington, Texas, USA
alex.dillhoff@uta.edu

Ramez Elmasri
The University of Texas at Arlington
Arlington, Texas, USA
elmasri@cse.uta.edu

Vassilis Athitsos
The University of Texas at Arlington
Arlington, Texas, USA
athitsos@uta.edu

ABSTRACT

Existing learning-based methods require a large number of labeled data to produce accurate part segmentation labels. However, acquiring ground truth labels is costly, giving rise to a need for methods that either require fewer labels or can utilize other currently available labels as a form of weak supervision for training. In this paper, in order to mitigate the burden of labeled-data acquisition, we propose a data-driven method for hand part segmentation on depth maps without any need for extra effort to obtain segmentation labels. The proposed method uses the labels already provided by public datasets in terms of major 3D hand joint locations to learn to estimate the hand shape and pose given a depth map. Given the pose and shape of a hand, the corresponding 3D hand mesh is generated using a deformable hand model and then rendered to a color image using a texture based on Linear Blend Skinning (LBS) weights of the hand model. The segmentation labels are then computed from the rendered color image. Since segmentation labels are not provided with current public datasets, we manually annotate a subset of the NYU dataset to perform quantitative evaluation of our method and show that a mIoU of 42% can be achieved with a model trained without using segmentation-based labels. Both qualitative and quantitative results confirm the effectiveness of our method. The code is publicly available for research purposes at: https://git.io/JmCBS.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; • **Hand pose estimation → hand joint localization**; • **Hand shape estimation → hand part segmentation**.

KEYWORDS

3D hand pose estimation, 3D hand shape estimation, semantic segmentation, hand part segmentation, human-computer interaction, Deep Learning, Computer Vision

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
PETRA 2021, June 29-July 2, 2021, Corfu, Greece
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8792-7/21/06…$15.00
https://doi.org/10.1145/3453892.3453902

ACM Reference Format:
Mohammad Rezaei, Farnaz Farahanipad, Alex Dillhoff, Ramez Elmasri, and Vassilis Athitsos. 2021. Weakly-supervised hand part segmentation from depth images. In *The 14th PErvasive Technologies Related to Assistive Environments Conference (PETRA 2021), June 29-July 2, 2021, Corfu, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3453892.3453902

1 INTRODUCTION

The hand is a crucial human body part that enables numerous daily activities. As such, hand part segmentation using vision systems is necessary for interaction between people and digital devices and thus is crucial in many applications relating to computer vision and human computer interaction (HCI), such as augmented reality (AR), virtual reality (VR) and gesture recognition, which could have numerous downstream applications including assisting people with disabilities [14, 39]. Hand part segmentation is a very challenging task due to the large degree of variation in hand appearance, heavy self-occlusion, large variability in global orientation and self-similarity between hand parts.

As depth cameras become more accurate, more affordable, and more widely used, significant advancements have been made in depth-based hand pose estimation [4, 11, 12, 18, 31, 50, 51, 53] and segmentation [8, 47]. Nonetheless, hand part segmentation has received little attention. In this paper, we propose the first data-driven method to perform hand part segmentation. Our method differs from existing depth-based hand segmentation methods in that they consider the whole hand as one semantic entity and attempt to segment out the hand from the background [8, 47]. In contrast, we divide the hand into six semantic parts, namely five fingers as well as the palm and attempt to assign pixel-wise labels to the input depth image.

It is widely recognized that deep learning-based methods are data intensive and thus require a large amount of annotated data to learn to carry out their respective tasks. However, acquiring segmentation labels for depth images is costly and labor intensive. To mitigate the burden of labeled data acquisition, our method uses the 3D hand pose labels, already provided with most public datasets, as a form of weak supervision. More specifically, a deep model is first trained to perform both 3D hand pose and shape estimation similar to [9]. The hand shape is represented as a triangular mesh parameterized by pose coefficients of a deformable hand model [40]. As a preprocessing step, we use LBS weights to assign each triangular face of the mesh a semantic label that determines which hand part it belongs to. Each hand part is given a pre-defined color.

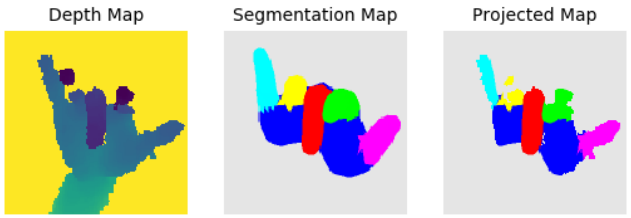

Figure 1: The outline of the the proposed method

Based on this color map, we create a texture by which the mesh is rendered to a color image. We can simply obtain each pixel's semantic label according to their color. Finally, the color image is refined to ensure that the segmentation labels are aligned with the input depth map (see Fig 1).

Despite some similarities, our method differs from [9] in that their goal is to estimate hand shape and pose while our method is aimed at performing hand part segmentation. Our method also estimates hand shape and pose as a by-product. We conduct extensive experiments to evaluate the proposed method both qualitatively and quantitatively. Since there is no public depth dataset that provides both hand part segmentation labels and 3D joint locations as ground-truth, we manually label a subset of the NYU dataset to perform quantitative evaluation.

To the best of our knowledge, this is the first data-driven method to perform hand part segmentation on depth images. Both quantitative and qualitative results confirm that our method achieves a good performance despite the fact that it has not been trained with a single segmentation-labeled data. This paper is organized as follows: Section 2 provides a literature review of works related to the proposed method, Section 3 lays out in detail how the method works, and finally the quantitative and qualitative results are presented in Section 4.

2 RELATED WORK

2.1 3D Hand Pose Estimation

Hand pose estimation remains a challenging problem that has brought about novel advancements in both Computer Vision and Machine Learning. Before the widespread adoption of deep learning techniques, many approaches relied on hand crafted features, optimization methods, and distance metrics. Athitsos et al. used edge maps and Chamfer matching to perform 3D hand pose estimation [3]. Other works leveraged pose estimation for tracking using optimization methods such as Particle Swarm Optimization (PSO) [36, 41]. After the rise of deep learning and with the advent of low cost consumer depth cameras, several methods have been proposed to perform 3D hand pose estimation based on depth maps [4, 11, 12, 18, 31, 50, 51]. Zimmermann in [57] proposed the first data-driven method to estimate 3D hand pose using a single RGB image. Several methods have since been introduced to perform monocular hand pose estimation [5, 6, 9, 17, 33, 52, 56].

In recent years, data efficient methods have gained popularity as models have grown larger and more complicated, resulting in the significantly increased need for labeled training data. Authors in [10, 43] use different data modalities to compensate for the lack of available annotated data. In [28], 2D annotations are used as weak supervision to train a 3D pose estimator. Wan in [49] proposed a data-driven self-supervised method for the task of depth-based 3D hand pose estimation to eliminate the need for any real data label. Our method is similar in spirit to these methods as it is aimed at mitigating the need for explicit labeled data which could be hard to acquire.

2.2 Hand Segmentation

Most existing methods cast hand segmentation as a dense prediction problem for every pixel in the image, where the task is to assign every pixel a label to determine whether it belongs to hand or not (binary classification). Hand segmentation from color images can be broadly categorized into two groups: 1). methods that take their visual clue from and are based on skin [2, 7, 20, 48]. One can refer to [22] for more details. 2). methods that are based on motion [1, 15, 27, 29, 42].

However, it is notoriously challenging to accurately detect skin in unconstrained settings due to severe light condition variations and complex effects like subsurface scattering, making it difficult to develop segmentation methods that could work well on images in the wild. Unlike color images, hand segmentation from depth images does not suffer from these problems. This line of research was pioneered by [47]. [8] provided a dataset for hand segmentation on depth images featuring multiple hands. In [23], they proposed a method to perform hand segmentation for hand-object interaction.

In contrast to these methods, we formulate our problem as a semantic segmentation task where the goal is to assign one label from a predefined set of class labels (one label per part) to each pixel [37]. In other words, we are interested in determining for every pixel what hand part they belong to.

In [41, 45], they perform hand part segmentation as part of their experiment and use its performance as a proxy for the accuracy of 3D pose estimation. [21] reports hand part segmentation performance as a proxy for detailed surface registration. However, our method is fundamentally different from them in that their approach is not data-driven, meaning that these methods perform optimization on each test data individually, while our method is data-driven and accumulates knowledge over the course of training. Another clear advantage of our data-driven approach is that unlike these methods, it does not require to do computationally expensive optimization at inference time and the inference can be done by a single forward pass of the network, which only takes around 100 *ms* on average on a single Nvidia GTX 1080 Ti GPU.

2.3 Hand Models

In order to represent the hand, many hand models have been proposed in recent years. Some early works modeled the hand using geometric primitives [36]. Subsequent researches used various methods such as sphere meshes [46], sum of Gaussians [44], or loop subdivision of a control mesh [26]. In this work, we use the hand model proposed in [40] referred to as MANO. The MANO hand model has a high representation power and has made many improvements on previous hand models including learning pose dependent corrective blend shapes, first proposed in [30], to correct

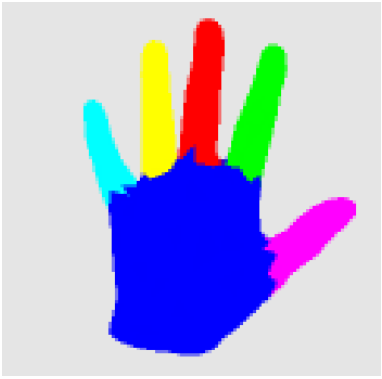

Figure 2: The hand divided into six semantic parts, which are five fingers as well as the palm

some limitations of the standard Linear Blend Skinning that lead to unnatural results. It is used as a fully differentiable layer in our network to allow for end-to-end training of the whole pipeline.

3 METHOD OVERVIEW

As illustrated in Fig 3, our method includes two general stages. The first stage essentially follows the standard training paradigm of 3D pose estimation methods [9]. The 3D pose and shape estimator network is trained using weak labels in terms of 3D joint locations of the hand. The network takes as input a depth map of size 128×128 and regresses the hand shape $\vec{\beta}$ and pose $\vec{\theta}$ parameters, scale S and the translation vector T. The hand parameters are passed on to the differentiable articulated mesh deformation hand model that generates a triangulated 3D mesh. The 3D mesh is scaled by S and then translated by T, which is in turn fed to the regressor to compute the underlying 3D skeleton. The supervision is applied to the predicted 3D skeleton to minimize its deviation from the ground-truth 3D skeleton. The regressor is a single linear layer trained prior to this stage and kept fixed over the course of training at this stage. The hand model is also kept fixed during training. At inference time, the network takes a depth image as input and estimates the hand model parameters to generate its corresponding 3D mesh. The 3D mesh is then rendered to a color image using Neural Renderer [24]. The color of each mesh triangular face is chosen according to the hand part it belongs to, which is derived from LBS skinning weights provided by the MANO hand model. Thus the semantic label for each pixel can simply be computed based on its color in the rendered color image. Finally, the rendered color image is aligned with the input depth map to ensure that no background pixel is assigned a label as a hand part.

3.1 Hand Model

Our model attempts to fit the MANO hand model [40] to the input depth image. The MANO hand model parametrizes a hand using pose parameters $\vec{\theta}$, which represent the relative rotation between pre-defined joints and their parent joints in the kinematic tree, and hand shape parameters $\vec{\beta}$, which denote the linear shape coefficients that represent offsets from the template mesh $\bar{T}$. Given hand shape $\vec{\beta}$ and pose $\vec{\theta}$ vectors, the template mesh $\bar{T}$ is first sculpted as follows [40]:

$$T_P(\vec{\beta},\vec{\theta}) = \bar{T} + \sum_{n=1}^{|\vec{\beta}|} \beta_n S_n + \sum_{n=1}^{9K} (R_n(\vec{\theta}) - R_n(\vec{\theta^*}))P_n \quad (1)$$

where S_n is the n-th principal component of shape displacement, $|\vec{\beta}|$ is the number of linear shape coefficients, R_n denotes the part relative rotation matrix for the n-th joint in the kinematic tree, $\vec{\theta^*}$ represents the rest pose, and P_n is the n-th element in the matrix of pose blend shapes. The sculpted mesh is then deformed using Linear Blend Skinning [25] to generate the 3D mesh as follows [40]:

$$M(\vec{\beta},\vec{\theta}) = W(T_P(\vec{\beta},\vec{\theta}), J(\vec{\beta}), \vec{\theta}, \tilde{W}) \quad (2)$$

where W is a linear blend skinning [25] function applied to sculpted mesh T rigged with a kinematic tree of 16 joints. J is a joint regressor that takes the template mesh sculpted only by shape blend shapes (before applying pose blend shapes) and regresses the 3D joint locations, and $\tilde{W}$ is the matrix of the LBS weights. The MANO hand model parameters $\bar{T}$,S,P,J and $\tilde{W}$ are learned using registered hand scans by the training procedure detailed in [40]. These parameters are kept fixed during our training process.

In order to reduce the space of pose parameters and thus the possibility of generating unnatural meshes, instead of directly using pose parameters that represent angles between joints and their parents, we use coefficients of Principal Component Analysis (PCA), as in [40], which are computed on angle-axis representation of the respective joints in the data collected to build the model [40]. We use 26 PCA coefficients to represent the hand pose concatenated by a vector of size 3 representing the hand global orientation in axis-angle representation to form the pose vector $\vec{\theta} \in \mathbb{R}^{29}$. We use 10 coefficients for the shape $\vec{\beta} \in \mathbb{R}^{10}$.

Given the shape $\vec{\beta}$ and pose $\vec{\theta}$ parameters, the MANO layer generates a hand mesh through the function $M(\vec{\beta},\vec{\theta})$ of $N = 778$ vertices and 1538 faces.

3.2 Regressor

In order to extract joints that are compatible with the 14 standard joints in the NYU dataset, we pretrain a single-layer feed forward network without activation layer $\bar{R}$ that takes as input a 3D hand mesh and outputs 14 3D joint locations of the hand. We train the regressor $\bar{R}$ using manually annotated randomly generated meshes by sampling from the pose $\vec{\theta} \in [-1.3, +1.3]^{29}$ and shape $\vec{\beta} \in [-0.01, +0.01]^{10}$ of the MANO hand model. The regression is essentially a matrix multiplication and is therefore fully differentiable and can be integrated into our end-to-end trainable pipeline. After pre-training, the parameters of the regressor are kept fixed for the subsequent training of our pose and shape estimator network.

3.3 Pose and Shape Estimator Network

The pose estimator takes as input the depth image and estimates hand pose $\vec{\theta}$ and shape $\vec{\beta}$ parameters as well as translation $T = (T_x,T_y,T_z)$ and scale S. The backbone is a ResNet-50 network [19]. Its last fully connected layer is replaced with a fully connected layer of size 256 to encode the hand features into a latent space, followed

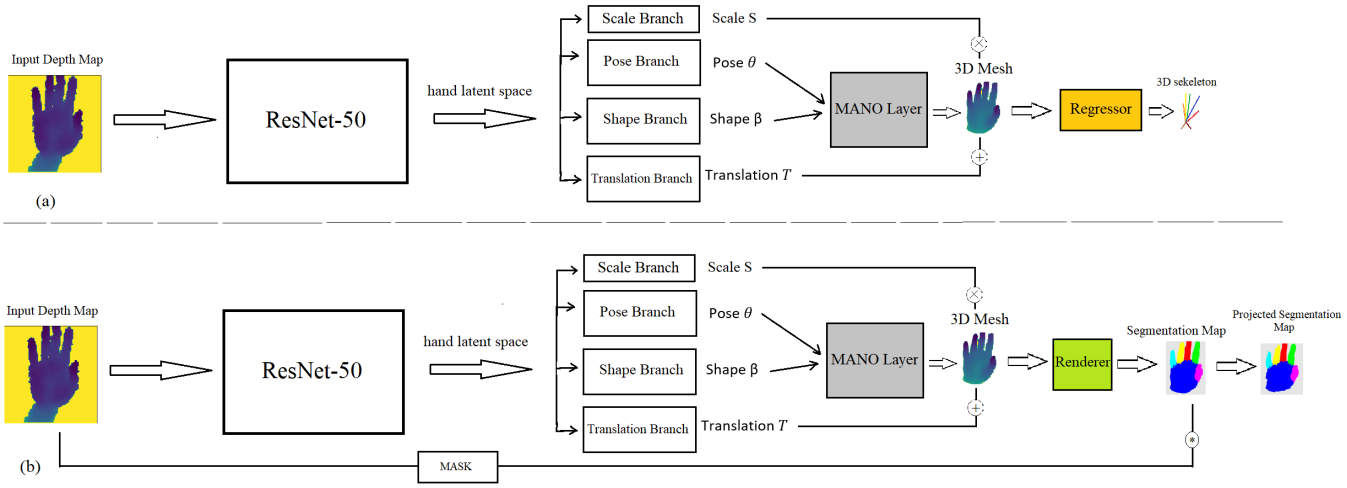

Figure 3: The general overview of the method. (a) At training time, the pose and shape estimator network takes a depth image as input and estimates the hand pose, shape, scale and translation, which are used to generate, scale and then translate the corresponding 3D mesh. The 3D mesh serves as input to the Regressor to compute the 3D hand joints. The weak supervision is applied to the estimated 3D joints using the ground-truth provided by the dataset. (b) At inference time, given a depth map, the model estimates the corresponding hand mesh and uses a renderer to obtain a color image. The segmentation labels are then computed based on the color of pixels in the rendered image.

by four separate branches to estimate the hand pose $\vec{\theta} \in \mathbb{R}^{29}$, hand shape $\vec{\beta} \in \mathbb{R}^{10}$, translation $T \in \mathbb{R}^3$ and scale $S \in \mathbb{R}$ respectively (see Fig 3).

3.4 Renderer

At inference time, we use the estimated mesh generated by the model to render it to a color image I' using the renderer proposed in [24]. We use a simple texture for rendering which is computed as follows. We determine for each vertex in the mesh what hand part it belongs to based on LBS skinning weights $\tilde{W}$ provided by the MANO hand model. Using this information, we determine for each triangular face of the mesh what hand part it belongs to by doing majority voting among its three vertices. Finally we assign each face a pre-defined color based on what hand part it belongs to (see Fig 2). This process is done offline and needs to be done only once. The segmentation label for each pixel is then easily computed based on the color of each pixel in the rendered color image. Orthographic projection is used for rendering the mesh.

3.5 Alignment

When the mesh is rendered to a color image, it may not be fit to the input depth map due to inaccuracies in the estimation of the pose and shape estimator network. This may result in some false positives. For example, some background pixels in the input depth image may be assigned a label as a hand part. In order to prevent this issue, we first compute the foreground mask for both input depth map I and the rendered color image I' as follows:

$$I'_{Mask}(P) = \begin{cases} 1, & \text{if } P_c \neq BC \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

$$I_{Mask}(P) = \begin{cases} 1, & \text{if } P_c \neq BD \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$MASK_{ref}(P) = I_{Mask}(P) \wedge I'_{Mask}(P) \tag{5}$$

where P denotes pixel, P_c is the value of the pixel P, BC denotes the background color in I' and BD denotes the background depth in the input depth map I. $\wedge$ denotes the logical AND operation. Finally, the rendered color image I' is aligned by multiplying I' by $MASK_{ref}$ to exclude the pixels in I' that correspond to the background pixels in the input depth map:

$$I'_{aligned}(P) = I'(P) \wedge MASK_{ref}(P) \tag{6}$$

3.6 Training

Since there is no segmentation label at training time, our method uses the 3D joint locations of the hand already provided with most depth-based public datasets to learn the task of hand pose and shape estimation. The pose and shape estimator network is trained by minimizing the following loss:

$$L = \alpha_{joint} L_{joint} + \alpha_{pose} L_{pose} + \alpha_{shape} L_{shape} \tag{7}$$

L_{joint} is aimed at minimizing the difference between the estimated joints and the ground-truth joints and is computed as follows:

$$L_{joint} = |J - J'|^2 \tag{8}$$

where $J, J' \in \mathbb{R}^{14\times3}$ are the ground-truth and estimated 3D locations of the standard 14 joints respectively. The estimated joint J' is computed as follows:

$$J' = \bar{R}(SM(\vec{\beta}, \vec{\theta}) + T) \tag{9}$$

where $\overrightarrow{\beta}, \overrightarrow{\theta}$, T and S are shape, pose, translation vector and scale respectively, which are estimated by the pose and shape estimator. L_{pose} and L_{shape} are defined as follows:

$$L_{pose} = \|\overrightarrow{\theta}\| \tag{10}$$

$$L_{shape} = \|\overrightarrow{\beta}\| \tag{11}$$

where $\|.\|$ denotes Frobenius Norm. L_{pose} and L_{shape} are used to regularize the space of pose and shape parameters to push them to be close to the mean shape and pose, and in doing so encourage generating more physically plausible meshes. To apply sutiable balance between the loss terms, we set α_{joint} = 10, α_{pose} = 1 and α_{shape} = 1000.

4 EXPERIMENTS AND DISCUSSION

In this section, we present both quantitative and qualitative results of the proposed method. To evaluate the proposed method, we need a dataset that provides 3D joint locations (for training) and segmentation labels (for testing). However, to the best of our knowledge, there is no such dataset. The only public depth dataset that provides segmentation labels for hand parts is the FingerPaint dataset [41]. However, it does not provide 3D joint locations required for training our method.

Because of the above-mentioned reasons, we chose to train our model on the NYU pose dataset [47], which is one of the most commonly used public benchmarks for hand pose estimation methods. This dataset, captured by 3 calibrated and synchronized PrimeSense depth cameras, consists of 72757 depth images for training and 8252 depth images for testing. NYU is a challenging dataset featuring hands that cover a wide range of hand poses. It provides the labels for depth images in terms of 3D joint locations of the hand, which are used by the proposed method to train the pose and shape estimator. Our network is implemented by PyTorch [38] and the Nvidia GTX 1080 Ti GPU is used for training. The model is trained end-to-end for 40 epochs using Adam optimizer with a learning rate of 10^{-4} and a learning decay of 10^{-1} every 20 epochs.

The NYU dataset provides about 7k annotations for hand segmentation. However, they are not suitable for our evaluation since they provide binary labels (hand or none-hand), whereas our method needs part-based segmentation labels. Thus, we manually label a subset of size 500 from the NYU test set for quantitative evaluation.

Table 1: Performance in terms of 2D Keypoint localization on the NYU dataset (Finger joints only). Mask R-CNN keypoint refers to the case where joint positions are localized by finding the positions of joint confidence maps with maximum probabilities. Mask R-CNN keypoint and mask restricts keypoints lying on estimated masks

Methods	Mean Keypoint error (Pixels)
Ours	**10.24**
Duan-KNN[13]	10.32
Mask RCNN(kpt and mask)[13]	15.7
Mask RCNN(kpt only)[13]	20.97

Table 2: Hand part segmentation performance

Hand Part	mIoU
Pinky Finger	0.38
Ring Finger	0.41
Middle Finger	0.41
Index Finger	0.37
Thumb	0.39
Palm	0.53
Average	0.42

4.1 Quantitative Evaluation

We begin the evaluation by reporting the performance of the proposed method in terms of 2D keypoint localization and 3D hand pose estimation. In order to generate accurate segmentation maps, it is crucial for the model to detect hand parts accurately. Thus, the ability of the proposed method to accurately localize 2D hand joints is strongly correlated with the performance of the method in terms of hand part segmentation. As can be seen in Table 1, despite the fact that our method's original goal is not to perform 2D keypoint localization, our method outperforms the state-of-the-art methods that were originally used to do 2D joint localization. Since [13] reports the results for only finger joints, in order to have a fair comparison, we select only finger joints out of the standard 14 joints estimated by our method. The numbers reported in Table 1 are computed by taking average across the selected joints. The localization accuracy for individual joints can be seen in Fig 6.

The performance of our method in terms of 3D hand pose estimation is reported in Table 3. As can be seen, our method compares with state-of-the-art methods in 3D pose estimation despite the fact that it has not been specialized for this task. A commonly used metric for reporting 3D results is mean 3D error, which is the average distance between the predicted joint location and its corresponding ground-truth in 3D space. Table 3 reports the average across all 14 joints for each method.

Next, we perform evaluation of the segmentation performance of the proposed method. Since we cast our problem as semantic segmentation with 6 classes (five fingers and the palm), we use two commonly used metrics for evaluating semantic segmentation methods. It is worth noting that unlike many RGB-based semantic segmentation methods that consider the background pixels as a separate semantic entity and assign a separate label to them, we do not consider the background pixel label assignment as it is trivial in depth images to segment out the background pixels using simple pre-processing steps such as thresholding. The first metric used for evaluation is Pixel Accuracy, which represents the proportion of pixels in the image that are labeled correctly. The second metric is Intersection over Union (IoU), which is calculated separately for each class, defined follows:

$$IoU = \frac{|TP|}{|TP| + |FP| + |FN|} \tag{12}$$

Where TP, FP and FN denote true positive, false positive and false negative respectively. As in [54], to account for class imbalance, we report class-wise average among classes, that is, mean IoU denoted by (mIoU). The results in terms of mIoU can be seen in Table 2. It

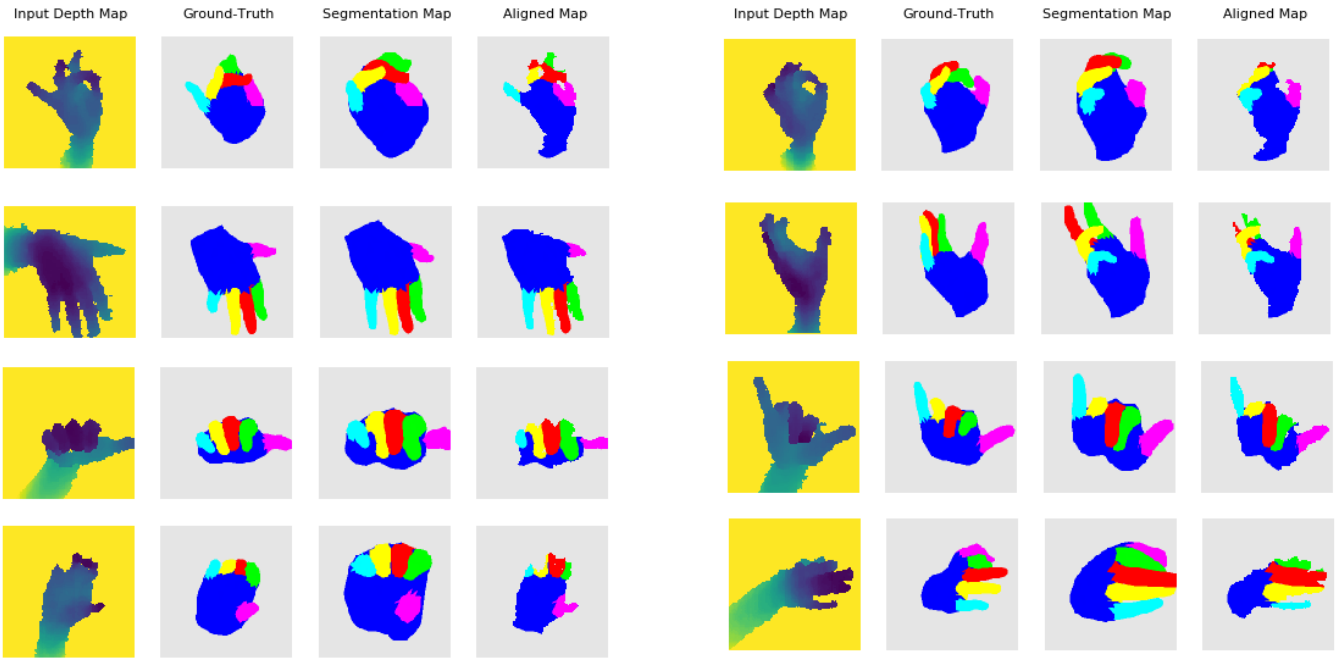

Figure 4: Qualitative results of the method given depth images of hands in various poses

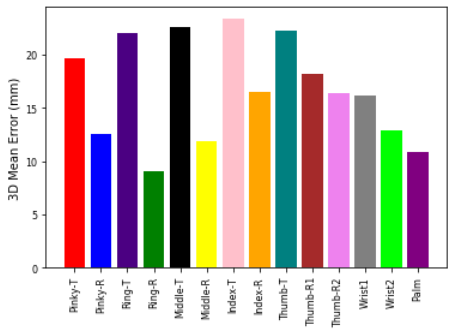

Figure 5: Per-joint mean 3D error. T and R denote tip and root respectively (e.g. Index-T denotes the tip of the index finger)

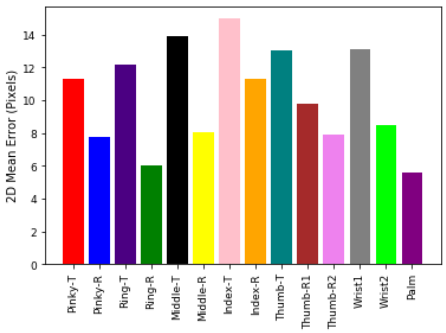

Figure 6: Per-joint mean 2D error. T and R denote tip and root respectively (e.g. Index-T denotes the tip of the index finger)

should be mentioned that since this is the first data-driven method proposed to perform depth-based hand part segmentation, there is no other work to compare against.

The hand palm is arguably easier to segment than other parts owing to the fact that fingers are more likely to be occluded, and if they do, the model is likely to mistake one finger for another since they look similar in many cases, which makes the task of segmenting fingers more challenging. Fingers also have higher levels of articulation and motion which leads to 3D joint labels of fingers being less accurate in comparison to the palm. As can be seen in Fig 5 and Fig 6, the accuracy of the method to localize the palm keypoints is higher than that for all the finger joints (except for Ring-R in the 3D case). As can be seen in Fig 5 and Fig 6, fingertips are the most difficult keypoints for the model to predict because they tend to get occluded more frequently than other keypoints. Yet, our method achieves a mIoU of 0.39 for fingers. It should be kept in mind that the proposed method achieves a good performance despite the fact that it has not been trained using segmentation labels. We also report the IoU averages across all classes and the Pixel Accuracy to be 0.42 and 92% respectively.

Table 3: Performance comparison with some of the state-of-the-art methods in terms of 3D hand pose estimation on NYU dataset [47]

Methods	Mean 3D error (mm)
Ours	17.66
DeepPrior [34]	20.75
DeepPrior-Refine [34]	19.72
DeepModel [55]	17.03
Feedback [35]	15.97
DeepHPS [32]	14.41
3DCNN [16]	14.11

4.2 Qualitative Evaluation

In order to verify the quality of the generated segmentation maps and the robustness of the proposed method in various cases, we draw some relatively hard samples from the NYU testing set and show the result of the proposed method on them as illustrated in Fig 4. Experiments demonstrate that our method is capable of generating high-quality hand meshes and as a result accurate part segmentation , which has many potential applications including in animation. Furthermore, as can be seen in Fig 4, the proposed method is robust in estimating hand shape and pose and as a result the segmentation map accurately even in hard scenarios such as self-occlusion and exaggerated articulation.

4.3 Conclusion and Future work

In this paper, we presented the first data-driven method to perform hand part segmentation on depth images. We investigated the possibility of taking advantage of weak labels (in this case 3D joint locations) to learn the task of hand part segmentation. Thus, our method does not impose any additional burden in terms of requiring extra effort to manually label data, which could be both expensive and labor intensive. Both quantitative and qualitative results demonstrate the effectiveness of our method. The proposed method could have many potential applications that have not been investigated in this paper, including but not limited to shape estimation which is used in animation, gesture recognition and Augmented/Virtual reality. This work opens new lines for future research, including extending the proposed method to RGB images, which are more widely used in real-world scenarios. The proposed method also opens up some possibilities in terms of improving the performance of 3D hand pose estimation methods by incorporating part segmentation labels.

REFERENCES

[1] Edoardo Ardizzone, Marco La Cascia, and Davide Molinelli. 1996. Motion and color-based video indexing and retrieval. In *Proceedings of 13th International Conference on Pattern Recognition*, Vol. 3. IEEE, 135–139.

[2] Antonis A Argyros and Manolis IA Lourakis. 2004. Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision*. Springer, 368–379.

[3] Vassilis Athitsos and Stan Sclaroff. 2003. Estimating 3D hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Vol. 2. IEEE, II–432.

[4] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2018. Augmented skeleton space transfer for depth-based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8330–8339.

[5] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2019. Pushing the envelope for rgb-based dense 3d hand pose estimation via neural rendering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1067–1076.

[6] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2020. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6121–6131.

[7] Sven Bambach, Stefan Lee, David J Crandall, and Chen Yu. 2015. Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions. In *Proceedings of the IEEE International Conference on Computer Vision*. 1949–1957.

[8] Abhishake Kumar Bojja, Franziska Mueller, Sri Raghu Malireddi, Markus Oberweger, Vincent Lepetit, Christian Theobalt, Kwang Moo Yi, and Andrea Tagliasacchi. 2019. Handseg: An automatically labeled dataset for hand segmentation from depth images. In *2019 16th Conference on Computer and Robot Vision (CRV)*. IEEE, 151–158.

[9] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 2019. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10843–10852.

[10] Yujun Cai, Liuhao Ge, Jianfei Cai, and Junsong Yuan. 2018. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 666–682.

[11] Yujin Chen, Zhigang Tu, Liuhao Ge, Dejun Zhang, Ruizhi Chen, and Junsong Yuan. 2019. So-handnet: Self-organizing network for 3d hand pose estimation with semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 6961–6970.

[12] Kuo Du, Xiangbo Lin, Yi Sun, and Xiaohong Ma. 2019. Crossinfonet: Multi-task information sharing based hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9896–9905.

[13] Le Duan, Minmin Shen, Song Cui, Zhexiao Guo, and Oliver Deussen. 2018. Estimating 2d multi-hand poses from single depth images. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[14] Farnaz Farahanipad, Harish Ram Nambiappan, Ashish Jaiswal, Maria Kyrarini, and Fillia Makedon. 2020. HAND-REHA: dynamic hand gesture recognition for game-based wrist rehabilitation. In *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 1–9.

[15] Alireza Fathi, Xiaofeng Ren, and James M Rehg. 2011. Learning to recognize objects in egocentric activities. In *CVPR 2011*. IEEE, 3281–3288.

[16] Liuhao Ge, Hui Liang, Junsong Yuan, and Daniel Thalmann. 2017. 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*. 1991–2000.

[17] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 10833–10842.

[18] Liuhao Ge, Zhou Ren, and Junsong Yuan. 2018. Point-to-point regression pointnet for 3d hand pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*. 475–491.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*. Springer, 630–645.

[20] Michael J Jones and James M Rehg. 2002. Statistical color models with application to skin detection. *International Journal of Computer Vision* 46, 1 (2002), 81–96.

[21] David Joseph Tan, Thomas Cashman, Jonathan Taylor, Andrew Fitzgibbon, Daniel Tarlow, Sameh Khamis, Shahram Izadi, and Jamie Shotton. 2016. Fits like a glove: Rapid and reliable hand shape personalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5610–5619.

[22] Praveen Kakumanu, Sokratis Makrogiannis, and Nikolaos Bourbakis. 2007. A survey of skin-color modeling and detection methods. *Pattern recognition* 40, 3 (2007), 1106–1122.

[23] Byeongkeun Kang, Kar-Han Tan, Nan Jiang, Hung-Shuo Tai, Daniel Tretter, and Truong Nguyen. 2017. Hand segmentation for hand-object interaction from depth map. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 259–263.

[24] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3907–3916.

[25] Ladislav Kavan and Jiří Žára. 2005. Spherical blend skinning: a real-time deformation of articulated models. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games*. 9–16.

[26] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. 2015. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2540–2548.

[27] Sangpil Kim, Hyung-gun Chi, Xiao Hu, Anirudh Vegesana, and Karthik Ramani. [n.d.]. First-Person View Hand Segmentation of Multi-Modal Hand Activity Video Dataset. ([n. d.]).

[28] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. 2020. Weakly-Supervised Mesh-Convolutional Hand Reconstruction in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4990–5000.

[29] Cheng Li and Kris M Kitani. 2013. Pixel-level hand detection in ego-centric videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3570–3577.

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)* 34, 6 (2015), 1–16.

[31] Jameel Malik, Ibrahim Abdelaziz, Ahmed Elhayek, Soshi Shimada, Sk Aziz Ali, Vladislav Golyanik, Christian Theobalt, and Didier Stricker. 2020. HandVoxNet: Deep Voxel-Based Network for 3D Hand Shape and Pose Estimation from a Single Depth Map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7113–7122.

[32] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Heloir, and Didier Stricker. 2018. Deephps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *2018 International Conference on 3D Vision (3DV)*. IEEE, 110–119.

[33] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

[34] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. 2015. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807* (2015).

[35] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. 2015. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*. 3316–3324.

[36] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect.. In *BmVC*, Vol. 1. 3.

[37] Gabriel L Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. 2016. Deep learning for human part discovery in images. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1634–1641.

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[39] Shishir G Patil, Don Kurian Dennis, Chirag Pabbaraju, Nadeem Shaheer, Harsha Vardhan Simhadri, Vivek Seshadri, Manik Varma, and Prateek Jain. 2019. Gesturepod: Enabling on-device gesture-based interaction for white cane users. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 403–415.

[40] Javier Romero, Dimitrios Tzionas, and Michael J Black. 2017. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)* 36, 6 (2017), 245.

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

[42] Yaser Sheikh, Omar Javed, and Takeo Kanade. 2009. Background subtraction for freely moving cameras. In *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 1219–1225.

[43] Adrian Spurr, Jie Song, Seonwook Park, and Otmar Hilliges. 2018. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 89–98.

[44] Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. 2014. Real-time hand tracking using a sum of anisotropic gaussians model. In *2014 2nd International Conference on 3D Vision*, Vol. 1. IEEE, 319–326.

[45] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, et al. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

[46] Anastasia Tkach, Mark Pauly, and Andrea Tagliasacchi. 2016. Sphere-meshes for real-time hand modeling and tracking. *ACM Transactions on Graphics (ToG)* 35, 6 (2016), 1–11.

[47] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)* 33, 5 (2014), 1–10.

[48] Aisha Urooj and Ali Borji. 2018. Analysis of hand segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4710–4719.

[49] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2019. Self-supervised 3d hand pose estimation through training by fitting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10853–10862.

[50] Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. 2018. Dense 3d regression for hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5147–5156.

[51] Fu Xiong, Boshen Zhang, Yang Xiao, Zhiguo Cao, Taidong Yu, Joey Tianyi Zhou, and Junsong Yuan. 2019. A2j: Anchor-to-joint regression network for 3d articulated pose estimation from a single depth image. In *Proceedings of the IEEE International Conference on Computer Vision*. 793–802.

[52] Linlin Yang, Shile Li, Dongheui Lee, and Angela Yao. 2019. Aligning latent spaces for 3d hand pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2335–2343.

[53] 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. 2018. Depth-based 3d hand pose estimation: From current achievements to future goals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2636–2645.

[54] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 633–641.

[55] Xingyi Zhou, Qingfu Wan, Wei Zhang, Xiangyang Xue, and Yichen Wei. 2016. Model-based deep hand pose estimation. *arXiv preprint arXiv:1606.06854* (2016).

[56] Yuxiao Zhou, Marc Habermann, Weipeng Xu, Ikhsanul Habibie, Christian Theobalt, and Feng Xu. 2020. Monocular Real-time Hand Shape and Motion Capture using Multi-modal Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5346–5355.

[57] Christian Zimmermann and Thomas Brox. 2017. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE international conference on computer vision*. 4903–4911.