Part Segmentation for Highly Accurate Deformable Tracking in Occlusions via Fully Convolutional Neural Networks

Weilin Wan¹, Aaron Walsman¹, and Dieter Fox¹,²

Abstract—Successfully tracking the human body is an important perceptual challenge for robots that must work around people. Existing methods fall into two broad categories: geometric tracking and direct pose estimation using machine learning. While recent work has shown direct estimation techniques can be quite powerful, geometric tracking methods using point clouds can provide a very high level of 3D accuracy which is necessary for many robotic applications. However these approaches can have difficulty in clutter when large portions of the subject are occluded. To overcome this limitation, we propose a solution based on fully convolutional neural networks (FCN). We develop an optimized Fast-FCN network architecture for our application which allows us to filter observed point clouds and improve tracking accuracy while maintaining interactive frame rates. We also show that this model can be trained with a limited number of examples and almost no manual labelling by using an existing geometric tracker and data augmentation to automatically generate segmentation maps. We demonstrate the accuracy of our full system by comparing it against an existing geometric tracker, and show significant improvement in these challenging scenarios.

I. INTRODUCTION

Human pose tracking in 3D is required for many robotic applications, including robotic medical care and personal assistance. Recent work in pose estimation and human body tracking has achieved accurate results in uncluttered scenarios, especially when depth information is available [1], [2], [3]. However, in most practical applications, humans often interact closely with objects and are often partially occluded from the view of the camera. In this case, the RGBD information generated by the objects can interfere with the geometric computation and cause errors in tracking.

In this paper, we focus on improving the robustness and reliability of pose tracking with occlusions using RGBD data. In particular, we propose a method based on FCN, which is developed from deep convolutional neural networks by adding deconvolutional layers, allowing the network to output pixel-wise classifications. Rather than individual labels, the output of FCNs can be the same resolution as the input images. We propose a lightweight Fast-FCN, and show that it can be incorporated into a geometric model-based tracker to drastically reduce errors caused by clutter and occlusion and to initialize the model pose. Figure 1 illustrates the structure of our system. FCNs have been applied by researchers in semantic segmentation analysis and pose estimation with RGB data, but we focus on optimization-based 3D tracking in this paper, in order to achieve the extremely high accuracy required in many practical robotics applications. Our model can achieve accurate pixel-wise predictions with a simplified network structure. The Fast-FCN architecture that we propose is able to achieve a much better run time performance with a minor accuracy cost in our task, which means we can run it along with geometric optimization and keep the overall system running at interactive frame rates.

Fully Convolutional Networks often require a large amount of training data which can be time-consuming and expensive to label by hand. In order to address this, we built a new RGBD dataset by using an existing geometric tracker to label human poses in unoccluded scenarios. We then randomly added artificial occluding objects to these videos and trained our network to label human body parts in the presence of occlusion. This data augmentation technique was extremely effective and allowed us to train these networks with almost no manual labelling. The geometric tracker alone frequently fails on this augmented dataset, but our new tracker that incorporates the Fast-FCN model is able to remove these occlusions and track the human poses successfully, both in this augmented dataset and in naturally generated sequences with occlusion.

The paper is organized as follows. In Section II, we discuss...
the related work. Section III describes the methodology of our tracking system. Section IV introduces the dataset we used for this task. The experiments and evaluations are documented in Section V. Section VI concludes.

II. RELATED WORK

Prior work in human pose estimation can be divided into two basic approaches: geometric tracking and discriminative prediction using machine learning. Geometric tracking had many early successes [4], [5], [1]. The core idea is to optimize the pose of a kinematic human model by reducing an error function designed to describe the distance between the current pose and an observation. In recent years this has been extended to deformable models [2], [3]. The downside of these approaches is that they often require some pose initialization, and can be sensitive to occlusion and interference from nearby objects. In our approach, we avoid these problems by incorporating a fast and effective discriminative model to initialize the pose when the model is out of place and filter points that do not belong to the body.

Discriminative methods also have a rich history [6], [7]. In these approaches, the position of joint locations are predicted directly from observations. The skeleton tracking in the ground-breaking Microsoft Kinect used this approach to estimate the pose and gestures of humans playing video games [8]. While the Kinect introduced cheap depth-sensing to the wider robotics community, there have also been several attempts to estimate 2D and 3D human pose directly from RGB images without the use of depth information [9], [10], [11]. More recently, discriminative methods using deep learning have shown remarkable success in this domain [12], [13], [14]. Recent methods have also been able to generate fine-grained part locations[15] and even 3D reconstructions of human pose [16], [17] from RGB images, although these methods have not been shown to produce the 3D accuracy necessary for safe physical human-robot interaction that is available using commercial depth sensors. In contrast, our approach uses an efficient discriminative model in conjunction with a 3D pose tracker designed to produce highly accurate spatial information. This plays to the strengths of both approaches: the discriminative component is able to locate subject anywhere in the frame and make important semantic decisions about which points belong to the body and which do not, which enables the tracking component to produce highly accurate 3D pose information even in the presence of clutter and occlusion.

There have also been many hybrid approaches to this problem [18], [19], [20]. A common approach here is to use a discriminative component to construct one or more additional loss terms for the tracker’s optimization to drive the pose toward detected body parts. In our approach, we use a fully convolutional network to segment the observed image into regions corresponding to various body parts and the background. We then use this segmentation to inform the association between our model and the point-cloud. We also augment our loss function to drive joints toward detected regions, but this is only used when the tracker is being initialized or has lost track of the subject.

Fully convolutional networks (FCN) have become quite popular for semantic segmentation [21], [22], [23]. These methods produce pixel-level predictions by first downsampling an image using convolutional layers to predict semantic features and then upsampling these features using deconvolutional layers to generate a high-resolution output. These approaches have been successfully applied to 2D human body part segmentation and pose estimation in many recent works [24], [25], [26]. While these techniques are able to attain high accuracy in terms of 2D joint locations and/or high IoU for part segmentation, we found that when used in conjunction with a good articulated tracker these performance metrics were not strongly indicative of final 3D tracking performance. We therefore opted to use a lightweight and fast network that allowed us to maintain both high accuracy and frame rate.

It is also worth noting that discriminative methods typically require very large datasets to train effectively. Here we again use the synergy between our two components to side-step this issue. While our learned model also requires significant training data, we were able to generate this easily by using our tracker to label sequences, and then augmenting these sequences with artificial occlusions and distractor objects. This allowed us to train our model in these difficult scenarios without resorting to manual data labelling.

III. METHODOLOGY

Our system consists of an FCN based classifier and an optimization-based tracker similar to [3]. The Fast-FCN model we propose is able to provide pixel-wise semantic segmentation of body parts. When tracking, instead of computing data association to the entire point cloud, we restrict ourselves to associations over the points labelled by the FCN. We also use these segmentation masks to provide a coarse error signal for the pose when initializing the model and when the optimizer loses track of the subject. Section III-A explains the architecture of our FCN. Section III-B introduces how we use the semantic segmentation in pose optimization and Section III-C discusses how the error signal for the pose is computed when initializing the model and recovering from tracking failures.

A. Fast-FCN Architecture

In this section, we focus on producing pixel-wise semantic segmentation of body parts for pose initialization and optimization via a Fully Convolutional Network (FCN). We use a second generation Microsoft Kinect that produces RGBD data with a resolution of $512 \times 424$. For performance reasons, we re-scale the input data to be $128 \times 106$ with four channels (RGB-D). The architecture of our network is illustrated in Figure 2. The first half of network architecture down-samples the input data from $128 \times 106 \times 4$ to $16 \times 13 \times 512$. In each convolutional layer, we use the rectified linear unit (ReLU) activation function. The second half up-samples the data symmetrically and produces a semantic logistic output for
each pixel. In order to keep the system running in real-time, we must balance accuracy and complexity. This tradeoff is discussed further in Section V.

When estimating segmentations we consider a label set $Y = \{\text{background, head, torso, right arm, left arm, right leg, left leg}\}$. The output is a single channel potential map with the same size as the input image, where $p_{iy} \in Y$ denotes the label of the $i^{th}$ pixel in the image. In our application, the most important task is to filter out objects that may interfere with the tracking by marking them as background. Inspired by [27], in which a modification of standard cross-entropy loss is proposed for imbalanced labels, we define our loss $L$ on frame $f$ by using cross-entropy along with a class imbalance modifier for all the pixels in $f$. First, for the $i^{th}$ pixel in $f$, the cross-entropy is calculated as:

$$H_i = -\sum_{y=1}^{|Y|} p_{iy} \cdot \log(p_{iy}')$$

where $p_{iy}'$ is the probability of pixel $i$ labeled as $y$, which is calculated using the softmax function given the network output label score $\{x_{iy}\}_{y=1}^{Y}$:

$$p_{iy}' = \frac{\exp(x_{iy})}{\sum_{y'=1}^{Y} \exp(x_{iy'})}$$

We add a class imbalance modifier $M$ to address the difference between background and body parts. At pixel $i$, $M$ is defined as:

$$M_i = \begin{cases} -\log(p_{iy}'=\text{background}), & \text{if } p_{iy} \text{ is background} \\ -\log(1 - p_{iy}'=\text{background}), & \text{otherwise} \end{cases}$$

Hence, for the frame $f$ with $n_p$ pixels in total, our loss $L$ is calculated as:

$$L = \frac{1}{n_p} \sum_{i=1}^{n_p} (H_i + \lambda M_i)$$

where $\lambda$ is a scalar factor for adjusting the effect of $M$. The Fast-FCN architecture was implemented in the Tensorflow [28] machine learning framework using the Adam optimization algorithm.

B. Optimization with Semantic Filtering

To achieve high accuracy in tracking, we combined our Fast-FCN with the 3D articulated human model based generative tracking method proposed in [3]. We modified the process of calculating the offsets between the model and the observations.

Our approach follows the geometric tracking paradigm in which we have an articulated kinematic model that we fit to point cloud observations. Each vertex $v$ on the model is attached to a kinematic skeleton via smooth-skinning [29] and has a semantic label $v_y \in Y$ Each step of the tracker consists of two phases. In the data association phase, a residual is computed based on associations between model vertices and nearby point cloud observations. Once we have this residual, the derivative of the model pose with respect to the combined error is computed and an optimization step is taken to minimize this error. There are many options and techniques for data association. Schmidt et al. [1] use a model made up of signed distance fields in order to quickly compute distances between the point cloud observations and model links, associating each point to the closest link. Ye and Yang [2] use a Gaussian Mixture Model of a random sample of their observations and model vertices. Walsman et. al [3] use a window-based search technique to match model vertices with the point cloud. Despite this variety, all of these techniques have one thing in common: they use physical distance to compute association.

In contrast, our technique first uses the Fast-FCN model described above to construct semantic labels for each pixel. Once these labels have been computed, we use the window-based search from [3], but only allow associations between vertices and observations that have the same label. This prevents two common failure modes. First, observation points that are not part of the subject are labelled background, which prevents occluding objects and clutter from attracting the model away from its correct pose. Second, this prevents different body parts from interfering with each other and helps the tracker resolve ambiguity when they get too close to each other. Once these associations have been made, the rest of our tracking procedure follows [3] with some minor
C. Initialization and Tracking Recovery

One issue common to many model-based trackers is the requirement of a good initialization on each frame. This means that if the model ever strays too far from the subject’s actual pose, the tracker may not be able to follow and could get stuck in a local bad optimum. This is especially problematic at the beginning of a sequence where the subject’s initial position is completely unknown. Many discriminative and hybrid techniques such as [18], [8] attempt to remedy this by detecting body part positions directly from the observations. In hybrid model-based approaches the distance between the model joint locations and these detected positions can then be used to construct an additional term in the optimizer’s loss function. We take a similar approach and use our body-part segmentation to generate approximate centroids for each body part. In practice, we found that these detected centroids do not need to be highly accurate, and if they are able to drive the model to within 20cm of the true position, the vertex-based tracking mechanism is able to take over and accurately recover the pose.

To generate centroids, we calculate one target position $t_y \in \mathbb{R}^3$ for each body part label in $Y$ except the background based on the semantic segmentation map. Rather than use an expensive detection mechanism to accurately locate joint positions, we use a binning approach to pick the point with the highest approximate density as shown in Figure 3. We then add a loss to the tracking error term of the form:

$$\sum_{y \in Y} \left| |t_y - p_y|^2 \right|_2$$

where $p_y$ is the position of a joint in the model that corresponds to the center of the body part (the elbow joints in the arms, knee joints in the legs, chest joint in the torso, and base of the skull for the head). Note that while the point with the highest density does not always correspond to these features, this method is only used to get the model into a position close enough for the 3D tracker to take over. With that in mind, we set a low weight on this error term so that it does not interfere with the tracker’s normal operation except during initialization and tracking failure.

IV. DATA GENERATION

For training and evaluation, we made an RGB-D indoor human video dataset by recording eight individuals performing a variety of motions. We applied the tracking system in [3] to these videos and used the output to generate ground truth for each video frame. We then augmented the dataset by inserting RGB-D images of different objects into each video sequence as clutter and occlusion. This approach allowed us to simultaneously generate ground truth information without expensive manual labelling while also producing difficult scenarios that the tracker could not handle alone.

The video sequences were made using a second generation Microsoft Kinect [30]. Each video contains 300 frames and the resolution of the depth data is $512 \times 424$. There are thirty-two video sequences of human body motion without occlusions, as well as seven video sequences of human interacting with everyday objects including a table, chair, suitcase and guitar. The thirty-two sequences without occlusions form the raw material for our augmented data, while the remaining seven sequences are used as an additional test set to ensure that our trained model works on natural data.

In our dataset, we mainly focus on natural poses, such as standing with arms swinging, walking around pulling a suitcase and sitting on stools. Since the goal of Fast-FCN classifier is to help the internal generative tracking method to maintain its tracking accuracy in occlusions, we only include poses that internal tracking methods are capable of recovering when no occlusions are present.

A. Generating Labels

In order to generate labels for the augmented data, we first generated a pixel-wise ground truth body part map for each frame in the unoccluded sequences. The ground-truth maps are $128 \times 106$ pixels and have seven labels: background, head, left arm, torso, right arm, left leg, right leg. As we mentioned above, we applied a model based tracking system to each sequence. We assigned each vertex of the model one of the corresponding labels, and then during tracking we used the geometric data association computed by tracker to transfer these labels to the point cloud. After running each sequence, we manually discarded any frames where the tracker failed, which was the only manual process in this procedure.

B. Data Augmentation

To simulate the inferences caused by occluding objects and clutter, we deliberately inserted different objects into the non-occluded videos. We cropped both RGB and depth images of different objects, including a sofa, lamp, bedside table, and bench from the SUN v2 RGB-D dataset [31]. Also, we collected our own RGB-D object images, including a suitcase, desk-chair, guitar, paper box, and stool using the Kinect. We augmented the dataset by overlaying these different objects onto each video sequence with different scaling, rotation, and depth offsets. In general we control the objects to occlude around one-third to a half of the person.

V. EXPERIMENTS

Our primary task is accurate 3D tracking using RGB-D data, so we test our system using a metric which measures the 3D distance from ground truth joint positions. We first
slightly to the left or right, occluding around one third of the interference, we inserted an object in front of the person proposed in [3]. To simulate the scenarios with object augmentation objects in this holdout set were not featured in any of the training data in order to guarantee that we did not overfit to a particular individual or occluding object. Additionally, in Figure 6, we provide test results of comparing our method against a geometric tracker [3] and a state-of-art RGB based regression approach [33]. The tests are performed using both object-inserted sequences and human-object interaction sequences from our dataset.

All tests were performed on a machine running Ubuntu 16.04 with a 4GHz Intel i7 and an Nvidia GTX 1070 graphic card. We trained our Fast-FCN network using twenty-eight sequences that were augmented with additional objects. A holdout set containing four video sequences with both segmentation and joint position ground truth, and 2 object images were used for testing. The human subjects and augmentation objects in this holdout set were not featured in any of the training data in order to guarantee that we did not overfit to a particular individual or occluding object.

Fig. 4: A comparison of tracking performance on the augmented object-insertion data compared against [3]. The x axis represents distance thresholds from the ground truth joint positions. Each curve represents the percentage of joint positions (y-axis) that are within the distance indicated from the ground truth locations (x-axis).

Fig. 5: Comparing our results with training by no occlusion dataset.

A. Tracking accuracy

We first compare our method with the tracking system proposed in [3]. To simulate the scenarios with object interference, we inserted an object in front of the person slightly to the left or right, occluding around one third of the body in each video. Also, to test the system with different levels of interference, for the same sequence we separately inserted the object with the furthest point of the object at either 30cm, 45cm, 60cm away from the person. Figure 4 shows the accuracy of each method as a function of how close a joint must be to the ground-truth position in order to be considered accurate. As the object is inserted closer to the person, the baseline method increasingly loses track and has a lower accuracy. However, with semantic segmentation, our system is able to perform with stability even in the most difficult scenarios.

B. The Effects of Data Augmentation

In order to evaluate the effect of data augmentation, we also trained a model on the original videos without adding occluding objects. The results are shown in Figure 5. Training on augmented data is clearly beneficial. It is also worth noting that the model trained without occlusions is actually worse than running the tracker alone (compare to Figure 4). This shows that while good semantic segmentation can help tracking performance, poor segmentation masks can be detrimental, because they can force data association to bad observations.

C. Architecture comparison

We chose two state-of-art approaches in semantic segmentation with deconvolutional neural networks for comparison. The first is U-Net [23], which also inspired our Fast-FCN architecture. In [26] [24] [25] [34], researchers developed FCN architectures by adding deconvolve layers based on the VGG-16 [35] classification network. Thus, we chose the VGG-FCN structure proposed in [25] for body part labeling as our second comparison. We rebuilt the structure of these networks, and only modified the sizes to fit our dataset. We trained the networks on our dataset, and ran until convergence. A summary of the structure of these three networks is shown in Table I along with the average running time for a single frame input.
Fig. 6: Our results compared against [3] and [33]. Part (A) and (B) are tested with object-inserted video sequences. Part (C) and (D) are tested with human-object interaction video sequences. In each column the first row is the input video frame, the second row is failure cases using the geometric tracker alone [3], the third row is the results using direct regression approach [33], and the forth row is the results from our technique.

To illustrate the accuracy of each network on our dataset, we use pixel-wise accuracy and intersection over union (IoU) as evaluation metrics. The pixel-wise accuracy is calculated as the ratio of correctly labeled pixels over the total number of pixels. However, in our case, merely comparing pixel-wise accuracy is misleading because the background always occupies most of the area in the image, dominating the accuracy calculation. Thus, we also computed pixel-wise IoU for comparison, which is the ratio of the sum of correctly labeled pixels over the sum of the union of similarly labeled pixels in the prediction and ground truth.

First, we test the networks with single frame images with objects inserted. The results in Table I and Table II show the accuracy of the semantic segmentation output. Then we tested tracking accuracy with the same video sequences and data augmentation, and show the results in Figure 7. We can see that our Fast-FCN model performs comparably to the U-Net, but at a third of the cost.

VI. CONCLUSION

In this paper, we proposed a real-time solution for human pose tracking with occlusions based on fully convolutional neural networks and 3D articulated human model. The Fast-
FCN we trained is able to accurately label the semantic segmentations of the person being tracked. By using the semantic segmentations, our system is able to initialize the starting pose automatically and track human poses in occlusions with high accuracy. We show that our model may be trained using a limited number of examples with no manual annotation.

Many robotics applications such health care and personal assistance require very accurate 3D estimation of human pose. Our system furthers these goals by improving accuracy in challenging scenarios with clutter and occlusions. We demonstrate the performance of our system by comparing it with traditional geometric tracking, and the results show that our new method is able to significantly improve the tracking accuracy in the presence of human-object interactions.

REFERENCES

[1] T. Schmidt, R. A. Newcombe, and D. Fox, “Dart: Dense articulated real-time tracking,” in *Robotics: Science and Systems*, vol. 2, no. 1, 2014.

[2] N. Ye and R. Yang, “Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2345–2352.

[3] A. Walsman, W. Wan, T. Schmidt, and D. Fox, “Dynamic high resolution deformable articulated tracking,” in *3D Vision (3DV), 2017 International Conference on*. IEEE, 2017, pp. 38–47.

[4] D. Grest, J. Woetzel, and R. Koch, “Nonlinear body pose estimation from depth images,” in *Joint Pattern Recognition Symposium*. Springer, 2005, pp. 285–292.

[5] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, “Real-time human pose tracking from range data,” in *European conference on computer vision*. Springer, 2012, pp. 738–751.

[6] 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*. Springer, 2016, pp. 160–177.

[7] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, “Vnect: Real-time 3d human pose estimation with a single rgb camera,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 44, 2017.

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. Ieee, 2011, pp. 1297–1304.

[9] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Progressive search space reduction for human pose estimation,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[10] U. Iqbal, A. Milan, and J. Gall, “Posetrack: Joint multi-person pose estimation and tracking,” *arXiv preprint arXiv:1611.07727*, 2016.

[11] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4733–4742.

[12] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime multi-person 2d pose estimation using part affinity fields,” *arXiv preprint arXiv:1611.08052*, 2016.

[13] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Breger, and K. Murphy, “Towards accurate multi-person pose estimation in the wild,” in *CVPR*, vol. 3, no. 4, 2017, p. 6.

[14] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, “Single-shot multi-person 3d pose estimation from monocular rgb,” in *3D Vision (3DV), 2018 Sixth International Conference on*. IEEE, 2018.

[15] R. A. Güler, N. Neverova, and I. Kokkinos, “Densepose: Dense human pose estimation in the wild,” *arXiv preprint arXiv:1802.06434*, 2018.

[16] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, “Bodynet: Volumetric inference of 3d human body shapes,” *arXiv preprint arXiv:1804.04875*, 2018.

[17] M. Omran, C. Lassner, G. Pons-Moll, P. V. Gehler, and B. Schiele, “Neural body fitting: Unifying deep learning and model-based human pose and shape estimation,” *arXiv preprint arXiv:1808.05942*, 2018.

[18] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, “Real-time motion capture using a single time-of-flight camera,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 755–762.

[19] T. Helten, A. Baak, G. Bharaj, M. Muller, H.-P. Seidel, and C. Theobalt, “Personalization and evaluation of a real-time depth-based full body tracker,” in *2013 International Conference on 3D Vision (3DV)*. IEEE, 2013, pp. 279–286.

[20] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, “Real-time continuous pose recovery of human hands using convolutional networks,” *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, p. 169, 2014.

[21] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[22] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[23] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[24] F. Xia, P. Wang, X. Chen, and A. L. Yuille, “Joint multi-person pose estimation and semantic part segmentation,” in *CVPR*, vol. 2, no. 6, 2017, p. 7.

[25] G. L. Oliveira, A. Valada, C. Bollen, W. Burgard, and T. Brox, “Deep learning for human part discovery in images,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1634–1641.

[26] A. Bulat and G. Tzimiropoulos, “Human pose estimation via convolutional part heatmap regression,” in *European Conference on Computer Vision*. Springer, 2016, pp. 717–732.

[27] W. Shen, X. Wang, Y. Yang, X. Bai, and Z. Zhang, “Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3982–3991.

[28] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., “Tensorflow: a system for large-scale machine learning,” in *OSDI*, vol. 16, 2016, pp. 265–283.

[29] L. Kavan, S. Collins, J. Zára, and C. O’Sullivan, “Geometric skinning with approximate dual quaternions blending,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 4, p. 105, 2008.

[30] J. Sell and O. Patrick, “The xbox one system on a chip and kinec sensor,” *IEEE Micro*, no. 1, pp. 1–1, 2014.

[31] S. Song, S. P. Lichtenberg, and J. Xiao, “Sun rgb-d: A rgb-d scene understanding benchmark suite,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[32] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, “Detect what you can: Detecting and representing objects using holistic models and body parts,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1971–1978.

[33] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, “End-to-end recovery of human shape and pose,” in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] K. Nishi and J. Miura, “Generation of human depth images with body part labels for complex human pose recognition,” vol. 71. Elsevier, 2017, pp. 402–413.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.