A Semantic Occlusion Model for Human Pose Estimation from a Single Depth Image

Umer Rafi
RWTH Aachen University
Germany
rafi@vision.rwth-aachen.de

Juergen Gall
University of Bonn
Germany
gall@iai.uni-bonn.de

Bastian Leibe
RWTH Aachen University
Germany
leibe@vision.rwth-aachen.de

Abstract

Human pose estimation from depth data has made significant progress in recent years and commercial sensors estimate human poses in real-time. However, state-of-the-art methods fail in many situations when the humans are partially occluded by objects. In this work, we introduce a semantic occlusion model that is incorporated into a regression forest approach for human pose estimation from depth data. The approach exploits the context information of occluding objects like a table to predict the locations of occluded joints. In our experiments on synthetic and real data, we show that our occlusion model increases the joint estimation accuracy and outperforms the commercial Kinect 2 SDK for occluded joints.

1. Introduction

Human pose estimation from depth data has made significant progress in recent years. One success story is the commercially available Kinect system [16], which is based on [20] and provides high-quality body joints predictions in real time. However, the system works under the assumption that the humans can be well segmented. This assumption is valid for gaming application for which the device was developed. Many computer vision applications, however, required human pose estimation in more general environments where objects often occlude some body parts. In this case, the current SDK for Kinect 2 [14] fails to estimate the partially occluded body parts. An example is shown in Figure 1(a) where the joints of the left leg of the person are wrongly estimated.

In this work, we address the problem of estimating human pose in the context of occlusions. Since for most applications it is more practical to have always the entire pose and not only the visible joints, we aim to predict the locations of all joints even if they are occluded. To this end, we build on the work from [11], which estimates human pose from depth data using a regression forest. Similar to the SDK, [11] works well for visible body parts but it fails to estimate the joint locations of partially occluded parts since it does not model occlusions. We therefore extend the approach by an occlusion model. Objects, however, not only occlude body parts but they also provide some information about the expected pose. For instance, when a person is sitting at a table as in Figure 1, some joints are occluded but humans can estimate the locations of the occluded joints. The same is true if the hands are occluded by a laptop or monitor. In this case, humans can infer whether the person is using the occluded keyboard and they can predict the locations of the occluded hands.

We therefore introduce a semantic occlusion model that exploits the semantic context of occluding objects to improve human pose estimation from depth data. The model is trained on synthetic data where we use motion capture data and 3D models of furniture. For evaluation, we recorded a dataset of poses with occlusions[1]. The dataset was recorded from 7 subjects in 7 different rooms using the Kinect 2 sensor. In our experiments, we evaluate the impact of synthetic and real training data and show that our approach outperforms [11]. We also compare our approach with the commercial pose estimation system that is part of Kinect 2. Although the commercial system achieves a higher detection rate for visible joints since it uses much more training data and highly engineered post-processing, the detection accuracy for occluded joints of our approach is twice as high as the accuracy of the Kinect 2 SDK.

2. Related Work

3D Human Pose Estimation. Human Pose Estimation is a challenging problem in computer vision. It has applications in gaming, human computer interaction and security scenarios. It has generated a lot of research surveyed in [18]. There are variety of approaches that predict 3D human

[1] The dataset is available at http://www.vision.rwth-aachen.de/data.

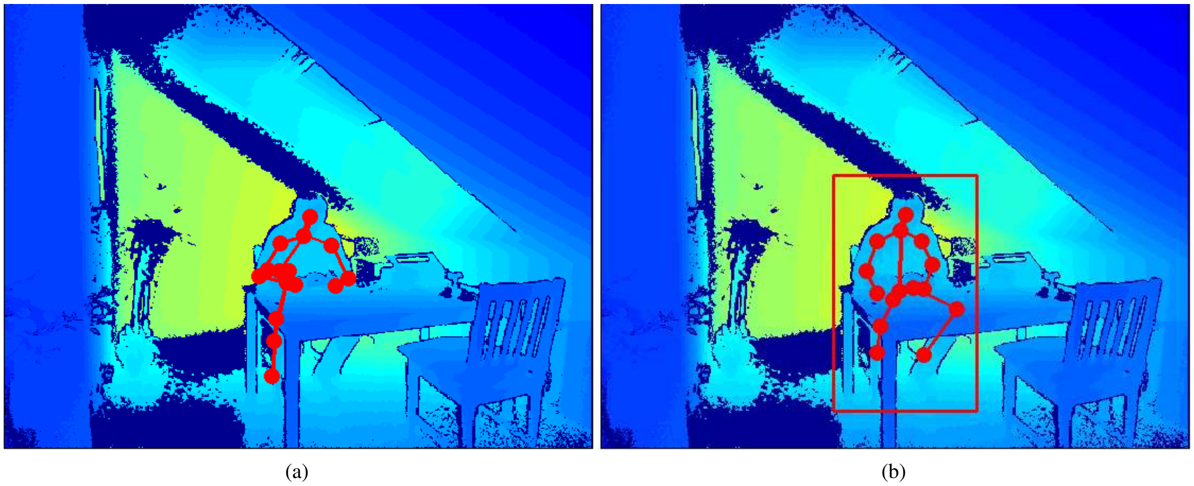

Figure 1: Comparison of Kinect 2 SDK and our approach. The SDK estimates the upper body correctly but the left leg is wrongly estimated due to the occluding table (a). Given the bounding box, our approach exploits the context of the table and predicts the left leg correctly (b).

pose from monocular RGB images [1, 3, 15]. However, a major caveat of using RGB data for inferring 3D pose is that the depth information is not available.

In recent years the availability of fast depth sensors has significantly reduced the depth information problem and further spurred the progress. Researchers have proposed tracking based approaches [13, 17, 8, 9]. These approaches work in real time but operate by tracking 3D pose from frame to frame. They cannot re-initialize quickly and are prone to tracking failures. Recently random forest based approaches [11, 20, 21, 23] have been proposed. They predict the 3D pose in super real time from a single depth image captured by Kinect. These approaches work on monocular depth images and therefore are more robust to tracking failures. Some model based approaches [2, 26] have also been proposed that fit a 3D mesh to image data to predict the 3D pose. However, all these approaches require a well-segmented person to predict the 3D pose and will fail for occluded joints when the person is partially occluded by objects. The closest to our method is approach from [11] that uses regression forest to vote for the 3D pose and can handle self-occlusions. Our approach on the other hand can also handle occlusion from other objects.

Occlusion Handling. Explicit occluder handling has been used in recent years to solve different problems in computer vision. Girshick *et al.* [12] use grammar models with explicit occluder templates to reason about occluded people. The occlusion patterns needs to be specially designed in the grammer. Ghiasi *et al.* [10] automatically learn the different part-level occlusion patterns from data to reason about occlusion in people-people interactions by using flexible mixture of parts [25]. Wang *et al.* [24] use patch based Hough forests to learn object-object occlusions patterns. In facial landmarks localization, recently some regression based approaches have been proposed that also incorporate occlusion handling for localizing occluded facial landmarks [4, 27]. However, these approaches only use the information from non-occluded landmarks in contrast to our approach that also uses the information from occluding objects. Similar to [10, 24] our approach also learns the occlusions from data, however our approach learns occlusions for people-objects interactions in contrast to [10] that learns occlusions for people-people interactions and the occluding objects in our approach are classified purely on appearance at test time and does not incorporate any knowledge about distance from the object they are occluding as in [24].

3. Semantic Occlusion Model

In this work, we propose to integrate additional knowledge about occluding objects into a 3D pose estimation framework from depth data to improve the pose estimation accuracy for occluded joints. To this end, we build on a regression framework for pose estimation that is based on regression forests [11]. We briefly discuss regression forests for pose estimation in Section 3.1. In Section 3.2 we extend the approach for explicitly handling occlusions. In particular, we predict the semantic label of an occluding object at test time and then use it as context to predict the position of an invisible joint.

3.1. Regression Forests for Human Pose Estimation

Random Forests have been used in recent years for various regression tasks, *e.g.*, for regressing the 3D human pose from a single depth image [11, 20, 21, 23], estimating the 2D human pose from a single image [7], or for predicting facial features [6]. In this section, we briefly describe the approach [11], which will be our baseline.

Regression forests belong to the family of random forests and are ensembles of T regression trees. In the context of human pose estimation from a depth image, they take an input pixel q in a depth image D and predict the probability distribution over the locations of all joints in the image. Each tree t in the forest consists of split and leaf nodes. At each split node a binary split function $\phi_\gamma(q, D) \mapsto \{0, 1\}$ is stored, which is parametrized by γ and evaluates a pixel location q in a depth image D. In this work, we use the depth comparison features from [20]:

$$\phi_\gamma(q, D) = \begin{cases} 1 & \text{if } D\left(q + \frac{u}{D(q)}\right) - D\left(q + \frac{v}{D(q)}\right) > \tau \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where the parameters $\gamma = (u, v, \tau)$ denote the offsets u, v from pixel q, which are scaled by the depth at pixel q to make the features robust to depth changes. The threshold converts the depth difference into a binary value. Depending on the value of $\phi_\gamma(q, D)$, (q, D) is send to the left or right child of the node.

During training each such split function is selected from a randomly generated pool of splitting functions. This set is evaluated on the set of training samples $Q = \{(q, D, c, \{V_j\})\}$, each consisting of a sampled pixel q in a sampled training image D, a class label c for the limb the pixel belongs to, and for each joint j the 3D offset vectors $V_j = q_j - q$ between pixel q and the joint position q_j in the image D.

Each sampled splitting function ϕ, partitions the training data at the current node into the two subsets $Q_0(\phi)$ and $Q_1(\phi)$. The quality of a splitting function is then measured by the information gain:

$$\phi^* = \arg\max_\phi g(\phi), \tag{2}$$

$$g(\phi) = H(Q) - \sum_{s\in\{0,1\}} \frac{|Q_s(\phi)|}{|Q|} H(Q_s(\phi)), \tag{3}$$

$$H(Q) = -\sum_c p(c|Q)\log(p(c|Q)), \tag{4}$$

where $H(Q)$ is the Shannon entropy and $p(c|Q)$ the empirical distribution of the class probabilities computed from the set Q. The training procedure continues recursively until the maximum allowed depth for a tree is reached.

At each leaf node l, the probabilities over 3D offset vectors V_j to each joint j, *i.e.*, $p_j(V|l)$ are computed from the training samples Q arriving at l. To this end, the vectors V_j are clustered by mean-shift with a Gaussian Kernel with bandwidth b and for each joint only the two largest clusters are kept for efficiency. If V_{ljk} is the mode of the k^{th} cluster for joint j at leaf node l, then the probability $p_j(V|l)$ is approximated by

$$p_j(V|l) \propto \sum_{k\in K} w_{ljk} \cdot \exp\left(-\left\|\frac{V - V_{ljk}}{b}\right\|_2^2\right) \tag{5}$$

where the weight of a cluster w_{ljk} is determined by the number of offset vectors that ended in the cluster. Cluster centers with $\|V_{ljk}\| > \lambda_j$ are removed since these correspond often to noise [11].

For pose estimation, pixels q from a depth image D are sampled and pushed through each tree in the forest until they reach a leaf node l. For each pixel q, votes for the absolute location of a joint j are computed by $x_j = q + V_{ljk}$. In addition a confidence value that takes the depth of the pixel q into account is computed by $w_j = w_{ljk} \cdot D^2(q)$. The weighted votes for a joint j are collected for all pixels q and form the set $\mathcal{X}_j = \{(x_j, w_j)\}$. The probability of the location of a joint j in image D is then approximated by

$$p_j(x|D) \propto \sum_{(x_j,w_j)\in\mathcal{X}_j} w_j \cdot \exp\left(-\left\|\frac{x - x_j}{b_j}\right\|_2^2\right) \tag{6}$$

where b_j is the bandwidth of the Gaussian Kernel learned separately for each joint j. As for training, the votes are clustered and only the clusters with the highest summed weights w_j are used to predict the joint location.

3.2. Occlusion Aware Regression Forests (OARF)

In order to handle occlusions, we propose Occlusion Aware Regression Forests (OARF) that build on the regression forest framework described in Section 3.1. They predict additionally the class label of an occluding object at test time and then use this semantic knowledge about the occluding object as context to improve the pose estimation of occluded joints. To this end, we use an extended set of training samples $Q_{ext} = Q \cup Q_{occ}$, where $Q_{occ} = \{(q_{occ}, D, c_{occ}, \{V_{jocc}\})\}$, is a set of occluding object pixels where each pixel q_{occ} is sampled from a training image D, has a class label c_{occ} and a set of of offset vectors $\{V_{jocc}\}$ to each joint of interest j.

During training we use the depth comparison features described in Section 3.1 for selecting a binary split function $\phi_\gamma(q, D)$ at each split node in each tree. To select binary split functions that can distinguish between occluding objects and body parts we minimize the Shannon entropy $H(Q)$ over extended set of class labels $c_{ext} = c \cup c_{occ}$:

$$H(Q) = -\sum_{c_{ext}} p(c_{ext}|Q)\log(p(c_{ext}|Q)), \tag{7}$$

To use the semantic knowledge about occluding object as an additional clue for prediction of occluded joints, we also store at a leaf node l the probabilities over 3D offset vectors V_{jocc} to each joint j, *i.e.*, $p_j(V_{occ}|l)$ that are computed from the training samples Q_{occ} arriving at l by using the mean shift procedure described in Section 3.1. The probability $p_j(V_{occ}|l)$ is approximated by

$$p_j(V_{occ}|l) \propto \sum_{k \in K} w_{ljk} \cdot \exp\left(- \left\| \frac{V_{occ} - V_{ljocck}}{b} \right\|_2^2 \right) \quad (8)$$

The inference procedure is similar to standard regression forest inference. At test time pixels q_{occ} from occluding objects in a test image D are also pushed through each tree in the forest until they reach a leaf node l and cast a vote $x_{jocc} = q_{occ} + V_{ljocck}$ with confidence w_{jocc} for each joint j. The weighted votes for each joint j form the set $\mathcal{X}_j = \{(x_j, w_j) \cup (x_{jocc}, w_{jocc})\}$. The final joint position is then predicted by using the mean shift procedure described in Section 3.1.

4. Training Data

Gathering real labeled training data for 3D human pose estimation from depth images is expensive. To overcome this, [20] generated a large synthetic database of depth images of people covering a wide variety of poses together with pixel annotations of body parts. For generating such a database, a large motion capture corpus of general human activities has been recorded. The body poses of the corpus were then retargeted to textured body meshes of varying sizes. Since the dataset is not publicly available, we captured our own dataset.

Synthetic Data. We follow the same procedure. To this end, we use the motion capture data for sitting and standing poses from the publicly available CMU motion capture database [5]. We retarget the poses to 6 textured body meshes using Poser [19], a commercially available animation package, and generate a synthetic dataset of 1'110 depth images of humans in different sitting and standing poses from different viewpoints. For each depth image, we also have a pixel-wise body part labeling and the 3D joint positions. The depth maps and body part labels are shown in Figure 2(a-b). For the occluding objects, we use the publicly available 3D models of tables and laptops from the Sweet Home 3D Furniture Library [22]. We render the object together with the humans as shown in Figure 2(c). The compositions are randomized under the constraints that the tables and feet are on the ground plane, the laptops on the tables and the distance between the humans and the objects is between 3-5 cm. For each composition, we compute the occluded body parts (Figure 2(d)) and the depth and class labels with occluding object classes Figure 2(e-f).

Real Data. We also recorded a dataset using the Kinect 2 sensor. The dataset contains depth images of humans in different sitting and standing poses without occlusions as shown in Figure 2(a). The 3D poses are obtained by the Kinect SDK and we discarded images were the SDK failed. This resulted in 2'552 images. The pixel-wise segmentation of the body parts is obtained by the closest geodesic distance of a surface point to the skeleton as shown in Figure 2(b). The composition with synthetic 3D objects is done as for the synthetic data.

5. Experiments and Results

In this section we describe in detail the experimental settings used to evaluate our method and report quantitative and qualitative results. For comparison, we consider three approaches:

1. The approach [11] described in Section 3.1 is our baseline. It is trained on the training data without occluding objects shown in Figure 2(a-b).

2. The occlusion aware regression forest (OARF W semantics) described in Section 3.2 is trained with the semantic labels of occluding objects shown in Figure 2(e-f).

3. To show the impact of the semantic information of the occluding objects, we also train an OARF by assigning all occluding objects a single label and not the labels of the object classes (OARF W/O semantics).

Training. We train all forests with the same parameters. For each training depth image, we sample 1000 pixel locations. The other parameters are used as in [11], *i.e.*, the regression forests consist of 3 trees with maximum depth 20. For each splitting node, we sample 2000 depth comparison features and use $b = 0.05m$.

Testing. For testing our approach, we use synthetic and real data. The real dataset is recorded in different indoor environments, offices and living rooms, and consists of seven sequences of different subjects in different sitting and standing poses. From each sequence, we select a set of unique poses thus providing us with a total of 1000 images. We split the 1000 images into test and validation set with 800 and 200 images, respectively. While b_j were set as proposed in [11], we observed that the values for λ_j proposed in [11] are not optimal for our baseline. We therefore estimate them on the validation set. The synthetic test set consists of 440 images of 2 subjects. The subjects, occluding objects and the poses in both test sets are different from the ones in the training set. We report quantitative results on real and synthetic test sets for the 15 body joint positions of our skeleton.

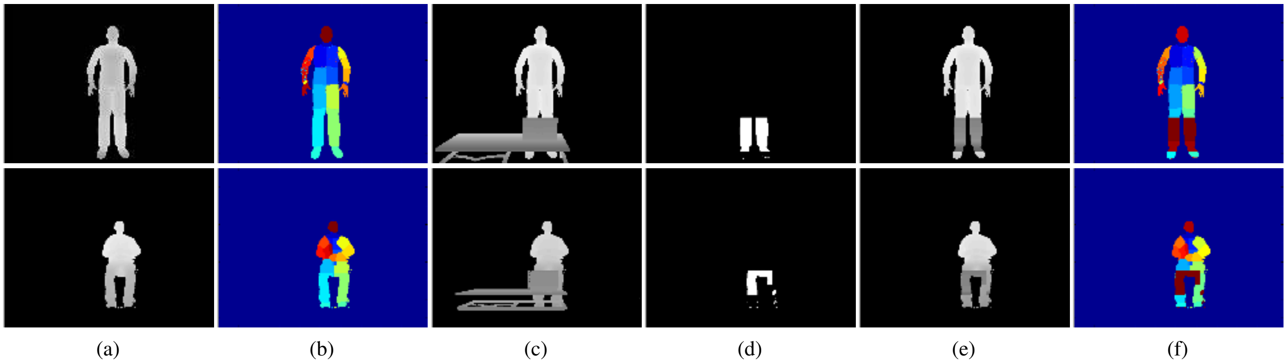

Figure 2: Procedure for generating depth images with pixel level ground truth body parts labels and occluding objects masks with class labels. Synthetic data (top row) and real data (bottom row): (a) Rendered or captured and segmented depth image. (b) Body part labels at pixel level. (c) Depth data with object. (d) Occluded body parts. (e) Segmented depth map with occluding objects. (f) Combined labels of body parts and occluding objects.

Method	Mean Average Precision
Baseline	44%
OARF W/O semantics	48%
OARF W semantics	54%

Table 1: Mean average precision of the 3D body joints predictions on the synthetic test set by using the evaluation measure from [20] with a distance threshold of 10 cm. The results show that integrating semantic knowledge about occluding objects provides a significant improvement over the baseline.

Synthetic Test Data. For quantitative evaluation on synthetic data, we use the evaluation measure from [20] with a distance threshold of 10 cm and report the mean average precision over the 3D body joints predictions in Table 1. Integrating the additional knowledge about occluding objects without object class labels alone provides 4% improvement over the baseline. When we also integrate semantic knowledge about occluding objects then this provides a significant improvement of 10% over the baseline. This shows that occlusion handling is beneficial for pose estimation, but also that the semantic context of occluding objects contains important information about joint locations.

Real Test Data. For real test data, it is difficult to get 3D ground truth body joint positions. For the quantitative evaluation, we therefore manually labeled the 2D body joint positions. Since manual annotations of the 2D positions of body joints, especially occluded joints, in depth images are sometimes noisy, we used the mean annotations of three different annotators as ground truth. For the quantitative evaluation on the test data, we use the evaluation measure from [7] with an error threshold of 0.1 of upper body size to report average detection accuracy over 2D body joint predictions.

Setting	Average Detection Accuracy(%)		
	Occluded Joints	Non Occluded Joints	All Joints
Synthetic Data			
Baseline	22.17	50.51	44.54
OARF W/O semantics	24.36	52.57	46.62
OARF W semantics	25.42	52.43	46.73
Real Data			
Baseline	25.42	46.21	41.82
OARF W/O semantics	28.26	49.72	45.19
OARF W semantics	31.12	51.69	47.94
Real + Synthetic Data			
Baseline	28.62	55.02	49.45
OARF W/O semantics	32.60	55.50	50.66
OARF W semantics	35.77	56.01	51.72
Kinect SDK	18.13	66.36	56.94

Table 2: Average detection accuracy of 2D body joint predictions on the real test set measured by the evaluation measure from [7] with an error threshold of 0.1 of upper body size.

In Table 2, the results for the baseline, OARF with and without semantics are reported. We also evaluate the impact of the synthetic and real training data. The results show that the synthetic training data is not as good as the real training data. However, if we combine real and synthetic data we get another boost of performance. For all three training settings, the numbers support the results of the synthetic test set. The baseline is improved by occlusion handling and semantic occlusion handling achieves the best result of the three methods. The semantic occlusion model mainly improves the accuracy of occluded joints, but there is also a slight improvement of non-occluded joints. Without occlusion handling, objects close to joints introduce some noisy votes that can result in wrong estimates. The occlusion handling reduces this effect. We also compared our results to the Kinect SDK. The Kinect SDK achieves a higher overall accuracy since it is trained on much more training data and the SDK includes additional post-processing, which is not part of our baseline. However, our approach achieves a much better accuracy for the occluded joints.

Table 3 presents detailed quantitative results for 11 body joints that are occluded in the real test set. The results are reported for OARF trained on real and synthetic data and for the Kinect SDK. The results show that for most joints our model achieves a better accuracy than the Kinect SDK and adding semantic knowledge provides further improvements. There are only three joints, namely the right elbow and the ankles, with a low accuracy. This can be explained by the training data that contains only few examples where these joints are occluded.

Qualitative Results. We show some qualitative results on our real test set in Figure 3 and a few failure cases in Figure 4. The top row shows the body joints predicted by OARF with the semantic occlusion model, the middle row shows the body joints predicted by OARF without the semantic occlusion model and the bottom row shows the body joints predicted by the Kinect SDK.

6. Conclusion

In this paper, we have proposed an approach that integrates additional knowledge about occluding objects into an existing 3D pose estimation framework from depth data. We have shown that occluding objects not only need to be detected to avoid noisy estimates, but also that the semantic information of occluding objects is a valuable source for predicting occluded joints. Although our experiments already indicate the potential of the approach and outperform a commercial SDK already for occluded joints, the overall performance can still be boosted by increasing the variety of objects and poses in the training data.

Acknowledgment: The work in this paper was funded by the EU projects STRANDS (ICT-2011-600623) and SPENCER (ICT-2011-600877). Juergen Gall was supported by the DFG Emmy Noether program (GA 1927/1-1).

Joint	Per Joint Detection Accuracy(%)		
	OARF W semantics	OARF W/O semantics	Kinect SDK
Spine	77.51	73.9	42.57
Left Elbow	76.47	73.53	47.06
Right Elbow	17.5	10.53	71.93
Left Hand	47.59	36.55	28.28
Right Hand	59.38	39.58	18.23
Left Hip	50.53	48.76	10.60
Right Hip	75.42	71.25	22.08
Left Knee	31.64	31.64	22.60
Right Knee	27.27	29.09	12.73
Left Ankle	3.87	4.9	5.15
Right Ankle	5.44	5.07	7.88
Average	35.77	32.60	18.13

Table 3: Per joint detection accuracy of 11 occluded body joints in our real test set by using the evaluation measure from [7] with an error threshold of 0.1 of upper body size. The results are reported for OARF with and without semantics and for the Kinect SDK.

References

[1] A. Agarwal and B. Triggs. Recovering 3d Human Pose from Monocular Image. *PAMI*, 28(1):44–58, 2006. 2

[2] A. Baak, M. Muller, G. Bharaj, H. P. Seidel, and C. Theobal. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *ICCV*, 2011. 2

[3] L. Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast Algorithms for Large Scale Conditional 3d Prediction. In *CVPR*, 2008. 2

[4] X. P. Burgos-Artizzu, P. Perona, and P. Dollar. Robust face landmark estimation under occlusion. In *ICCV*, 2013. 2

[5] CMU Mocap. http://mocap.cs.cmu.edu/. 4

[6] M. Dantone, J. Gall, G. Fanelli, and L. van Gool. Real-time Facial Feature Detection using Conditional Regression Forests. In *CVPR*, 2012. 3

[7] M. Dantone, J. Gall, C. Leistner, and L. van Gool. Human Pose Estimation using Body Parts Dependent Joint Regressors. In *CVPR*, 2013. 3, 5, 6

[8] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *CVPR*, 2010. 2

[9] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real Time Human Pose Tracking from Range Data. In *ECCV*, 2012. 2

[10] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes. Parsing Occluded People. In *CVPR*, 2014. 2

[11] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient Regression of General-Activity Human Poses from Depth Images. In *ICCV*, 2011. 1, 2, 3, 4

[12] R. B. Girshick, P. F. Felzenszwalb, and D. A. Mcallester. Object Detection with Grammar Models. In *NIPS*, 2011. 2

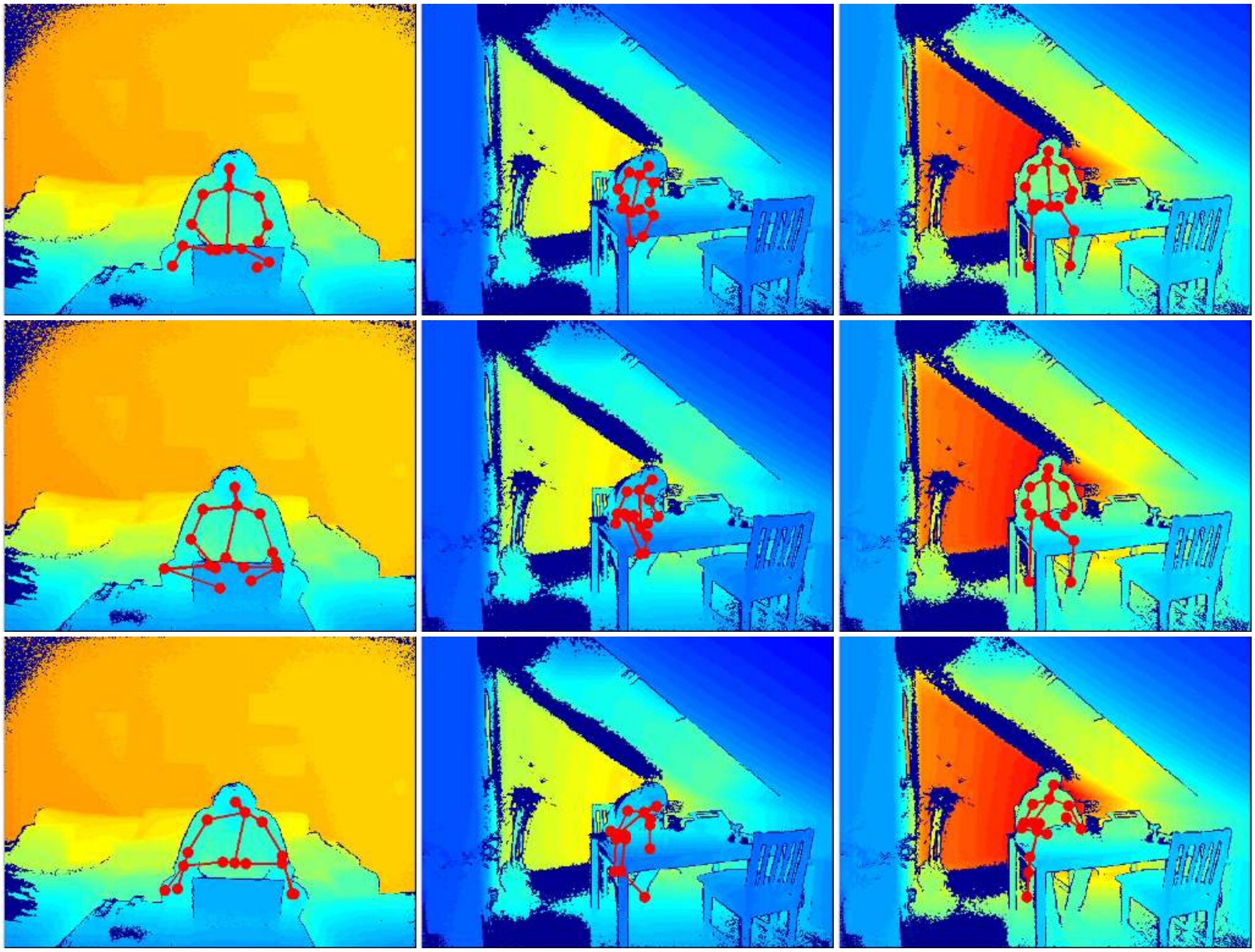

Figure 3: Qualitative results for some sample images taken from our real test set. Top row shows the body joints predicted by OARF with semantics. Middle row shows the body joint predictions from OARF without semantics. Bottom row shows the body joints predicted by the Kinect SDK. The results show that integrating knowledge about occluding objects helps in improving the body joint prediction performance for the occluded joints.

[13] D. Grest, J. Woetzel, and R. Koch. Nonlinear Body Pose Estimation from Depth Images. In *DAGM*, 2005. 2

[14] Kinect For Windows SDK 2.0. `http://www.microsoft.com/en-us/kinectforwindows/develop/`. 1

[15] I. Kostrikov and J. Gall. Depth Sweep Regression Forests for Estimating 3D Human Pose from Images. In *BMVC*, 2014. 2

[16] Microsoft Corp.Redmond WA. Kinect for Xbox 360. 1

[17] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun. Real-Time Identification and Localization of Body Parts from Depth Images. In *ICRA*, 2010. 2

[18] R. Poppe. Vision-based Human Motion Analysis. *CVIU*, 108(1-2):4–18, 2007. 1

[19] Poser. `http://my.smithmicro.com/`. 4

[20] 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 *CVPR*, 2011. 1, 2, 3, 4, 5

[21] M. Sun, P. Kohli, and J. Shotton. Conditional Regression Forests for Human Pose Estimation. In *CVPR*, 2012. 2, 3

[22] Sweet Home 3D. `www.sweethome3d.com`. 4

[23] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. In *CVPR*, 2012. 2, 3

[24] T. Wang, X. He, and N. Barnes. Learning Structured Hough Voting for Joint Object Detection and Occlusion Reasoning. In *CVPR*, 2013. 2

[25] Y. Yang and D. Ramanan. Articulated Pose Estimation using Flexible Mixtures of Parts. In *CVPR*, 2011. 2

[26] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefey. Accurate 3D pose estimation from a single depth image. In *ICCV*, 2011. 2

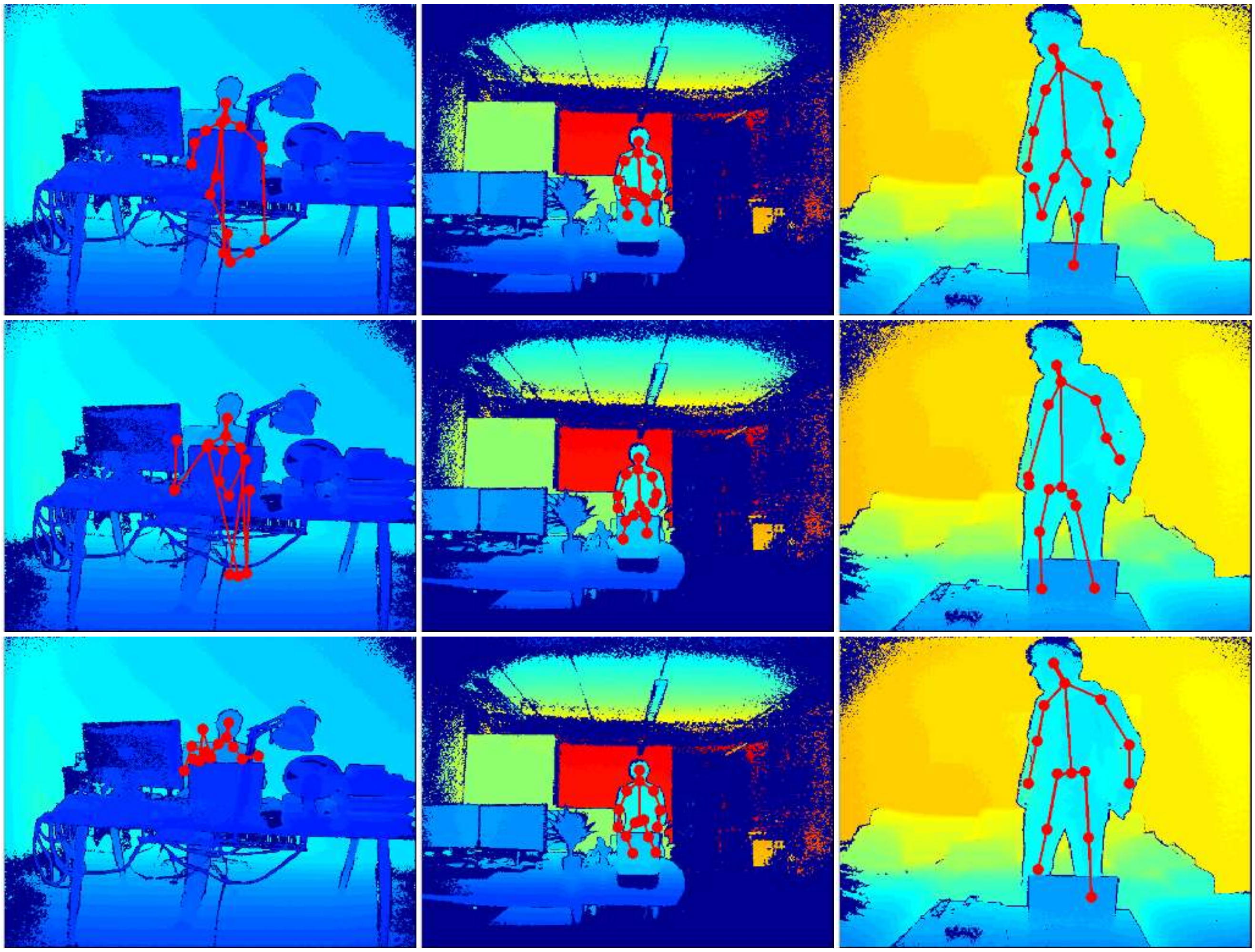

Figure 4: Example failure cases for occluded joints from our Real Test Set. OARF with semantics (top row), OARF without semantics (middle row) and Kinect SDK (bottom row).

[27] X. Yu, Z. Lin, J. Brandt, and D. N. Metaxas. Consensus of Regression for Occlusion-Robust Facial Feature Localization. In *ECCV*, 2014. 2