Supplemental Material of Learning Visibility for Robust Dense Human Body Estimation

Chun-Han Yao[1] Jimei Yang[2] Duygu Ceylan[2] Yi Zhou[2]
Yang Zhou[2] Ming-Hsuan Yang[134]

[1]UC Merced [2]Adobe [3]Google [4]Yonsei University

In this supplemental material, we present the implementation details in Section 1, model analysis in Section 2, and additional results in Section 3.

1 Implementation Details

We implement our framework with PyTorch [19]. The VisDB model and SMPL regressor are parametrized by deep neural networks and trained with an Adam optimizer [9] . We apply Batch Normalization [7] after each convolutional layer and use ReLU [1] as the activation function of the middle layers. The initial learning rate is set to 10^{-4} and decayed by a factor of 10 after 8 epochs.

1.1 Pseudo ground-truth visibility

To obtain the dense UV correspondence map, we apply a DensePose [5] model pre-trained on the MSCOCO dataset [14] with ResNet101 [6] as backbone and DeepLabv3 as prediction head. The model predicts a bounding box of each human body as well as the part segmentation mask (I) and pixel-wise UV coordinates in the bounding box. We discover that directly calculating the pixel-to-vertex correspondence is too time-consuming and not feasible during training. To deal with this issue, we discretize the UV coordinates by a 30×30 grid for each of the 24 body parts, as the maximum number of vertices per part is smaller than 900. Given an estimated IUV image, we can efficiently obtain the dense correspondence by discretizing it and indexing a pre-defined map $M_{uv} \in \{1, 2, ..., N_V\}^{24\times30\times30}$, where $N_V = 6890$ in our experiments.

1.2 Network architecture

We show the detailed structure of VisDB network and SMPL regressor in Figure 1. The VisDB network predicts the dense heatmaps and visibility in x, y, and z dimensions. We apply element-wise multiplication to the visibility labels and concatenate them with the 3D coordinates obtained from heatmaps, which is then used as the input of SMPL regressor.

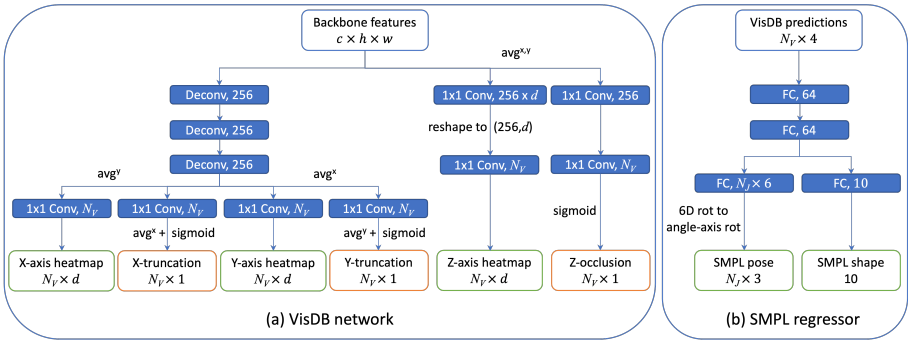


Fig. 1. Network architecture of the VisDB network (left) and SMPL regressor (right). We show the prediction modules of vertex coordinates and visibility labels. As in [17], we use a similar network for joint predictions and cascade the two networks. Given the VisDB predictions, we concatenate the 3D coordinates ($\mathbb{R}^{N_V \times 3}$) and overall visibility ($\mathbb{R}^{N_V \times 1}$) as the input of SMPL regressor.

1.3 Dataset details

In Table 1, we show the rough number of training and testing images for each dataset we use. For Human3.6M, we train our models using subjects S1, S5, S6, S7, S8, and we test the models using subjects S9 and S11. We first train our model on all the images from all datasets until convergence. Then, for the evaluation on the 3DPW dataset [15], we finetune the trained model on the whole 3DPW dataset and only 10% of the others since they are rather large-scale. We have also experimented with the UP-3D [11] (3D) and MPII [2] (2D) datasets for training, but we did not observe much performance gain.

Table 1. Dataset statistics. We show the number of training and testing images per dataset. For the testing datasets, we also report the number of partial-body (truncated) examples. The 3DPW dataset [15] contains more partial-body images.

	Human3.6M [8]	MuCo-3DHP [16]	3DPW [15]	MSCOCO [14]
Annotations	3D	3D	3D	2D
Training images	312.2K	200.0K	22.7K	149.8K
Testing images	2.2K	-	35.5K	-
Testing images (partial-body)	0.1K	-	1.9K	-

2 Model Analysis

2.1 Training and inference speed

Our entire training process takes about 24 hours to converge using 4 NVIDIA V100 GPUs. The inference speed of our VisDB network is around 23 fps on a single GPU, which is close to I2L-MeshNet [17] (25 fps) since both methods adopt a similar heatmap-based representation and network backbone On the other hand, METRO [12] and Mesh Graphormer [13] use a transformer [21]-based network which imposes higher computational costs and takes a longer training time to converge. METRO runs at 12 fps on a single NVIDIA P100 GPU, and Mesh Graphormer is slightly slower. Both transformer-based methods require 5 days for model training on 8 NVIDIA V100 GPUs.

2.2 Data augmentation and visibility evaluation

We evaluate the quality of our visibility predictions on the 3DPW dataset [15]. In Table 2, we report the mean accuracy of vertex truncation in the x and y-axis as well as occlusion in the z-axis. Note that the occlusion labels for both training and evaluation are pseudo ground-truths obtained from dense UV estimations. The results demonstrate that the VisDB network can effectively learn to predict accurate visibility labels with the proposed data augmentations with truncation and occlusion.

Table 2. Quantitative evaluations of visibility predictions. We report the accuracy of individual visibility prediction (x-axis truncation, y-axis truncation, and z-axis occlusion) on the 3DPW dataset [15]. The results demonstrate that our data augmentation strategies effectively facilitate the visibility learning.

Data augmentation	X-truncation	Y-truncation	Occlusion
✗	0.86	0.79	0.61
✓	**0.98**	**0.94**	**0.83**

2.3 Ablation study on visibility prediction

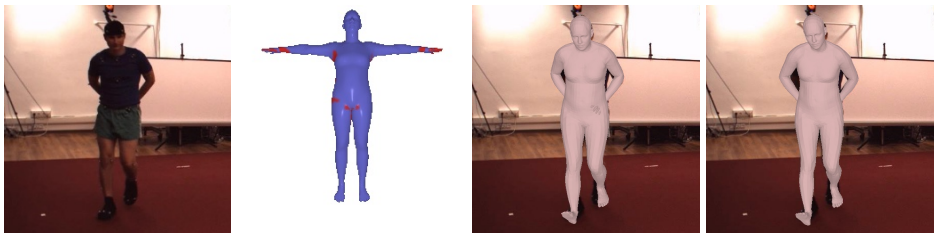
To justify our model design for visibility prediction, we perform ablation study with various prediction strategies in Table 3. A naive approach to model dense visibility is by predicting one binary label for each joint or vertex, which indicates whether it is overall visible in the image. The visibility labels can be predicted either from the image-space features (after deconvolution layers in Figure 1(a)) or depth-axis features (after $\text{avg}^{x,y}$ in Figure 1(a)). In contrast, we separate the joint/vertex visibility into 3 dimensions and train a network to predict 3 binary labels from the x, y, and z axis features, respectively. The results in Table 3 demonstrate that the separate visibility prediction from individual features is more effective with the heatmap-based framework.

Table 3. Ablation study on visibility prediction. In the first row, we predict one label of overall visibility per joint/vertex from the image-space features. The second model predicts the overall visibility from the depth-wise features. Finally, our VisDB network produces 3 separate labels for the 3 axes based on respective features. We report the accuracy of truncation and occlusion predictions as well as joint/vertex errors. The results show that separate visibility modeling leads to the best performance, which justifies our model design.

Prediction	Features	Truncation ↑	Occlusion ↑	MPJPE ↓	PA-MPJPE ↓	MPVE ↓
Overall visibility	Image-space (x,y)	0.81	0.69	78.4	46.3	89.8
Overall visibility	Depth-axis (z)	0.78	0.75	77.4	46.1	88.3
Separate visibility	All (x,y,z)	**0.93**	**0.83**	**73.5**	**44.9**	**85.5**

2.4 Depth ordering loss

In Figure 2, we show the outputs with and without depth ordering loss $\mathcal{L}_{depth}$ and visualize the self occluded regions (left and right hands). The results demonstrate that $\mathcal{L}_{depth}$ can effectively resolve depth ambiguity (left hand) based on visibility prediction.



Input | Depth error | w/o $\mathcal{L}_{depth}$ | w/ $\mathcal{L}_{depth}$

Fig. 2. Visualization of depth ordering loss. We visualize the depth ordering loss $\mathcal{L}_{depth}$ and show an example outputs with an without $\mathcal{L}_{depth}$ (blue:correct depth ordering, red:incorrect depth ordering).

3 Additional Results

3.1 Silhouette IoU evaluation

In addition to the joint and vertex error metrics, we evaluate the 2D silhouette IoU between the predicted mesh and ground-truth mesh. We show the comparison results in Table 4, which demonstrate that both the VisDB output mesh and optimized SMPL model capture the human body silhouettes more faithfully compared to I2L-MeshNet† [17]. Note that the heatmap-based mesh outputs produce higher silhouette IoUs than SMPL parameters. The proposed dense UV correspondence loss further improves the IoU since the UV map estimations are based on segmentation masks.

Table 4. Silhouette IoU evaluations on the 3DPW [15] dataset. The VisDB outputs and optimized SMPL parameters achieve higher silhouette IoUs than our baseline, I2L-MeshNet† [17]. Moreover, the dense UV correspondence loss $\mathcal{L}_{uv}$ is shown effective in improving the faithfulness of 2D silhouettes.

Method	MPJPE ↓	PA-MPJPE ↓	MPVE ↓	Silhouette IoU ↑
I2L-MeshNet† [17] (mesh)	84.5	51.1	98.2	86.3
I2L-MeshNet† [17] (param)	88.0	55.5	102.3	84.4
VisDB w/o $\mathcal{L}_{uv}$ (mesh)	74.9	45.6	87.1	87.7
VisDB w/ $\mathcal{L}_{uv}$ (mesh)	73.5	44.9	85.5	**90.3**
VisDB w/ $\mathcal{L}_{uv}$ (param)	**72.1**	**44.1**	**83.5**	89.9

3.2 Quantitative comparisons with dense UV-based losses

Several prior works take dense UV maps as additional input [22,4] and/or apply 2D losses via differentiable rendering [22,3], which either implies stricter inference conditions or imposes additional computations during training. Instead, our VisBD network only takes an image as input and produces image-space vertex position and visibility jointly, which exactly match the 2D supervisory signals we discovered from DensePose via Eq.(19) in the manuscript. We empirically found this direct vertex-level supervision more efficient and effective for training VisDB. In Table 5, we compare the performance of VisDB trained with the proposed vertex correspondence loss $\mathcal{L}_{uv}$, against a rendering-based loss $\mathcal{L}_{render}$ as baseline. To compute $\mathcal{L}_{render}$, we render the vertex part labels and UV coordinates of a VisDB output mesh and compare with DensePose estimates in terms of part mask IoU (as in [22]) and UV coordinate error (as in [23]). We also include other prior dense UV-based methods for reference.

Table 5. Human3.6M results against additional baselines.

Method	MPJPE	PA-MPJPE
NBF [18]	-	59.9
DenseRaC [22]	76.8	48.0
HoloPose [4]	60.3	46.5
OOH [24]	-	41.7
DSR [3]	60.9	40.3
DecoMR [23]	-	39.3
VisDB w/ $\mathcal{L}_{render}$ (mesh/param)	56.2 / 54.7	38.1 / 37.5
VisDB w/ $\mathcal{L}_{uv}$ (mesh/param)	**51.0 / 50.0**	**34.5 / 33.8**

3.3 Failure cases

In Figure 3, we show some failure cases of VisDB. Since VisDB only takes an image as input (without joint coordinates or segmentation mask), the model tends to be confused when multiple people are around the image center. In addition, the output meshes are sometimes tilted in the side view even though they look faithful from the front view. This is partly due to the erroneous pseudo ground-truths from SMPLify-X [20], which we often observe in the training datasets.

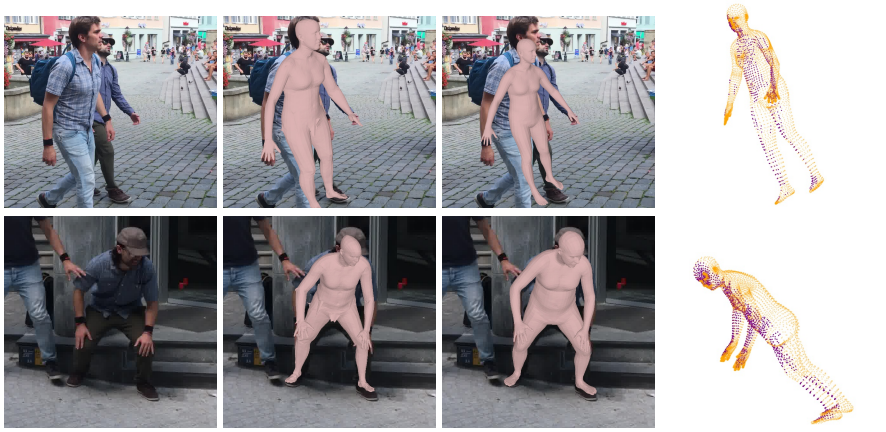


Fig. 3. Failure cases. For each example, we show the results of our VisDB mesh and optimized SMPL model, as well as visibility predictions in the side view (purple:visible, orange:invisible). The first example contains a crowded scene, where our model sees the two people in the middle as one. In the second example, we show that the output meshes are sometimes tilted in the side view despite being accurate in the front view.

3.4 Qualitative comparisons with occlusion-robust methods

In Figures 4, we show the qualitative comparisons against prior arts which can handle occlusions: PARE [10] and METRO [12]. We observe that PARE and METRO are robust to occlusions in general but VisDB aligns with the images better thanks to the accurate dense heatmap estimations.

3.5 Additional qualitative results

We show additional qualitative results on the Human3.6M [8] and 3DPW [15] datasets in Figure 5 and 6, respectively. For input images where the human bodies are truncated or occluded, the results of I2L-MeshNet [17] (param) are not accurate, especially around the face, hand, and foot regions. On the contrary, VisDB predicts faithful mesh and dense visibility, which lead to accurate SMPL parameters.

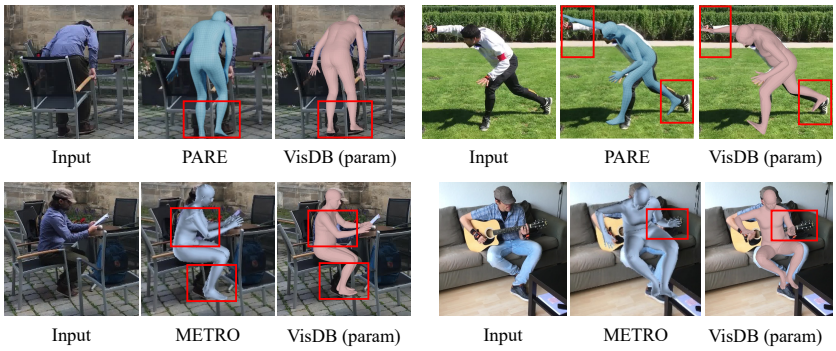


Fig. 4. Visual comparison with PARE [10] and METRO [12] (see red boxes).

Fig. 5. Qualitative results on the Human3.6M dataset [8]. For each example, we show the results of SMPLify-X [20] (pseudo ground-truth), I2L-MeshNet [17] (param), our VisDB mesh, our optimized SMPL parameters. This dataset contains more self-occlusion cases. We further apply random occlusions to demonstrate the performance gap. Despite that the pseudo ground-truth meshes for training are sometimes inaccurate, our VisDB results and optimized SMPL models are realistic, faithful to the input image, and robust to occlusions.

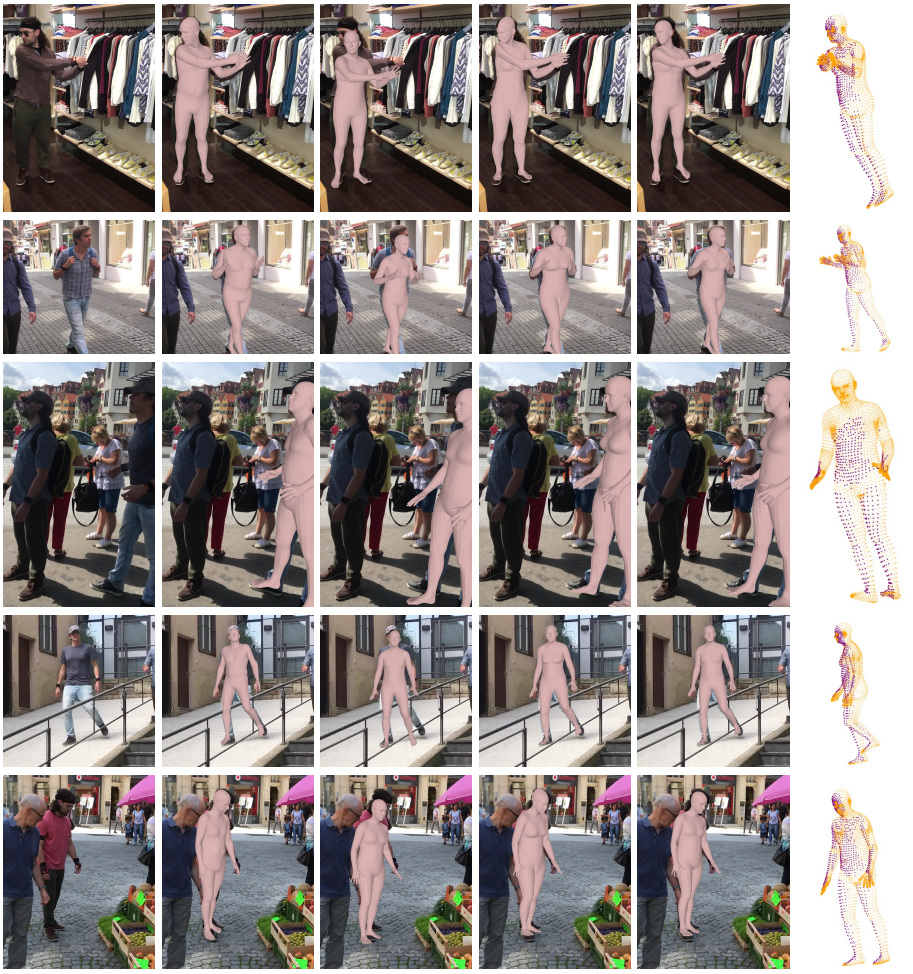


Fig. 6. Qualitative results on the 3DPW dataset [15]. For each example, we show the results of I2L-MeshNet [17] SMPL model, our VisDB mesh, our optimized SMPL model, as well as visibility predictions in the side views (purple:visible, orange:invisible).

References

1. Agarap, A.F.: Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375 (2018) 1
2. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: CVPR. pp. 3686–3693 (2014) 2
3. Dwivedi, S.K., Athanasiou, N., Kocabas, M., Black, M.J.: Learning to regress bodies from images using differentiable semantic rendering. In: ICCV. pp. 11250–11259 (2021) 5
4. Guler, R.A., Kokkinos, I.: Holopose: Holistic 3d human reconstruction in-the-wild. In: CVPR. pp. 10884–10894 (2019) 5
5. Güler, R.A., Neverova, N., Kokkinos, I.: Densepose: Dense human pose estimation in the wild. In: CVPR. pp. 7297–7306 (2018) 1
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 1
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML. pp. 448–456 (2015) 1
8. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. PAMI **36**(7), 1325–1339 (2013) 2, 6, 7
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 1
10. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: PARE: Part attention regressor for 3D human body estimation. In: ICCV. pp. 11127–11137 (2021) 6, 7
11. Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M.J., Gehler, P.V.: Unite the people: Closing the loop between 3d and 2d human representations. In: CVPR. pp. 6050–6059 (2017) 2
12. Lin, K., Wang, L., Liu, Z.: End-to-end human pose and mesh reconstruction with transformers. In: CVPR. pp. 1954–1963 (2021) 3, 6, 7
13. Lin, K., Wang, L., Liu, Z.: Mesh graphormer. In: ICCV (2021) 3
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014) 1, 2
15. von Marcard, T., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering accurate 3d human pose in the wild using imus and a moving camera. In: ECCV. pp. 601–617 (2018) 2, 3, 5, 6, 8
16. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. In: 3DV. pp. 120–130 (2018) 2
17. Moon, G., Lee, K.M.: I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In: ECCV. pp. 752–768 (2020) 2, 3, 4, 5, 6, 7, 8
18. Omran, M., Lassner, C., Pons-Moll, G., Gehler, P., Schiele, B.: Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In: 3DV. pp. 484–494 (2018) 5
19. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS **32**, 8026–8037 (2019) 1
20. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: CVPR. pp. 10975–10985 (2019) 6, 7

21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017) 3
22. Xu, Y., Zhu, S.C., Tung, T.: Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In: ICCV. pp. 7760–7770 (2019) 5
23. Zeng, W., Ouyang, W., Luo, P., Liu, W., Wang, X.: 3d human mesh regression with dense correspondence. In: CVPR. pp. 7054–7063 (2020) 5
24. Zhang, T., Huang, B., Wang, Y.: Object-occluded human shape and pose estimation from a single color image. In: CVPR. pp. 7376–7385 (2020) 5