Supplementary Material for
3D Human Mesh Estimation from Virtual Markers

Xiaoxuan Ma\textsuperscript{1} Jiajun Su\textsuperscript{1} Chunyu Wang\textsuperscript{3,*} Wentao Zhu\textsuperscript{1} Yizhou Wang\textsuperscript{1,2,4}

\textsuperscript{1} School of Computer Science, Center on Frontiers of Computing Studies, Peking University
\textsuperscript{2} Inst. for Artificial Intelligence, Peking University
\textsuperscript{3} Microsoft Research Asia
\textsuperscript{4} Nat’l Eng. Research Center of Visual Technology

\{maxiaoxuan, sujiajun, wtzhu, yizhou.wang\}@pku.edu.cn, chnuwa@microsoft.com

We elaborate on the post-processing implementation of the virtual markers and provide additional experimental details and results. At last, we discuss data from human subjects and the potential societal impact.

1. Post-processing on Virtual Markers

As described in Section 3.1, considering the left-right symmetric human body structure, we slightly adjust the learned virtual markers $Z$ to be symmetric. In fact, after the first step that updates each $z_i$ by its nearest vertex to get $e_Z \in \mathbb{R}^{3 \times K}$. $e_Z$ are almost symmetric with few exceptions.

To get the final symmetric virtual markers $e_Z^{sym} \in \mathbb{R}^{3 \times K}$, for each virtual marker located in the left body part, we take its symmetric vertex counterpart to be its symmetric virtual marker and finally get $e_Z^{sym}$.

2. Experiments

In this section, we first add detailed descriptions for datasets and then provide more experimental results of our approach.

\*Corresponding author

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure1.png}
\caption{Illustration of the human mesh template $X^t$ at the 3D coordinate system and a symmetric vertex pair $(X^t_i, X^t_j)$.}
\end{figure}

2.1. Datasets

\textbf{H3.6M [4].} Following previous works [6, 10, 11, 22], we use the SMPL parameters generated from MoSh [16], which are fitted to the 3D physical marker locations, to get the GT 3D mesh supervision. Following standard practice [6], we evaluate the quality of 3D pose of 14 joints derived from the estimated mesh, \textit{i.e.} $\hat{M}_{3D}$. We report Mean Per Joint Position Error (MPJPE) and PA-MPJPE in millimeters (mm). The latter uses Procrustes algorithm [3] to align the estimates to GT poses before computing MPJPE. To evaluate mesh estimation results, we also report Mean Per Vertex Error (MPVE) which can be interpreted as MPJPE computed over the whole mesh.

\textbf{3DPW [25].} The 3D GT SMPL parameters are obtained by using the data from IMUs when collected. Following the previous works [9, 13, 14, 26], we use the train set of 3DPW to learn the model and evaluate on the test set.

\textbf{MPI-INF-3DHP [19].} is a 3D pose dataset with 3D GT pose annotations. Since this dataset does not provide 3D
mesh annotations, following [6, 10], we only enforce supervision on the 3D skeletons (Eq. (9)) in mesh losses.

UP-3D [12] is a wild 2D pose dataset with natural images. The 3D poses and meshes are obtained by SMPLify [1]. Due to the lack of GT 3D poses, the fitted meshes are not accurate. Therefore we only use the 2D annotations to train the 3D virtual marker estimation network as in [23].

COCO [15] is a large wild 2D pose dataset with natural images. Previous work [20] used SMPLify-X [21] to obtain pseudo SMPL mesh annotations but they are not accurate. However, we find that if we project the 3D mesh to 2D image, the resulting 2D mesh vertices align well with the image. So we leverage the 2D annotations to train the virtual marker estimation network as in [23].

SURREAL [24] is a large-scale synthetic dataset containing 6 million frames of synthetic humans. The images are photo-realistic renderings of people under large variations in shape, texture, viewpoint, and body pose. To ensure realism, the synthetic bodies are created using the SMPL body model, whose parameters are fit by the MoSh [16] given raw 3D physical marker data. All the images have a resolution of 320 × 240. We use the same training split to train the model and evaluate the test split following [2].

2.2. Implementation Details and Computation Resource

Following common practice [2, 6, 8, 11, 13, 14, 20, 26], we conduct mix-training by using the above 2D and 3D datasets for experiments on the H3.6M and 3DPW datasets. To leverage the 3D pose estimation dataset, i.e. MPI-INF-3DHP [19], we extend the 64 virtual markers with the 17 landmark joints (i.e. skeleton) from the MPI-INF-3DHP dataset. For experiments on the SURREAL dataset, we use its training set alone as in [2, 18]. We implement the proposed method with PyTorch. All the experiments are conducted on a Linux machine with 4 NVIDIA 16GB V100 GPUs. The whole network is trained for 40 epochs with batch size 32 using Adam [7] optimizer.

We evaluate the model complexity in terms of FLOPs (G) and the number of model parameters in Table 1. Compared to the most recent state-of-the-art methods that directly regress all mesh vertices, such as I2L-MeshNet [20], METRO [13], and Mesh Graphormer [14], our approach with virtual marker representation reduces the computation overhead by a large margin while getting better estimation quality. The last column shows the MPVE errors on 3DPW test set for performance reference.

<table>
<thead>
<tr>
<th>Methods</th>
<th>FLOPs (G)</th>
<th>Params (M)</th>
<th>MPVE(^\downarrow)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2L-MeshNet [20] ECCV’20</td>
<td>28.7</td>
<td>141.2</td>
<td>110.1</td>
</tr>
<tr>
<td>METRO [13] CVPR’21</td>
<td>153.0</td>
<td>397.5</td>
<td>88.2</td>
</tr>
<tr>
<td>Mesh Graphormer [14] ICCV’21</td>
<td>48.8</td>
<td>180.6</td>
<td>87.7</td>
</tr>
<tr>
<td>Ours</td>
<td>22.1</td>
<td>109.6</td>
<td>77.9</td>
</tr>
</tbody>
</table>

Table 1. Computation overhead comparison with the recent state-of-the-art methods that directly regress all 3D vertices. The rightmost column shows the MPVE errors on the 3DPW test set for performance reference.

<table>
<thead>
<tr>
<th>MPVE(^\downarrow)</th>
<th>w/o (\mathcal{L}_{conf})</th>
<th>w/o (\mathcal{L}_{pose})</th>
<th>w/o (\mathcal{L}_{normal})</th>
<th>w/o (\mathcal{L}_{edge})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>58.0</td>
<td>59.2</td>
<td>50.3</td>
<td>60.6</td>
</tr>
</tbody>
</table>

Table 2. MPVE errors on H3.6M [4] test set when ablating different loss terms.

<table>
<thead>
<tr>
<th>Occ. VM Parts</th>
<th>MPVE(^\downarrow)</th>
<th>MPJPE(^\downarrow)</th>
<th>PA-MPJPE(^\downarrow)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None (Ours)</td>
<td>77.9</td>
<td>67.5</td>
<td>41.3</td>
</tr>
<tr>
<td>2 Arms</td>
<td>79.2 (\pm 1.3)</td>
<td>68.2 (\pm 0.7)</td>
<td>42.2 (\pm 0.9)</td>
</tr>
<tr>
<td>2 Legs</td>
<td>78.3 (\pm 0.4)</td>
<td>67.9 (\pm 0.4)</td>
<td>41.7 (\pm 0.4)</td>
</tr>
<tr>
<td>Body</td>
<td>78.6 (\pm 0.7)</td>
<td>68.0 (\pm 0.5)</td>
<td>41.8 (\pm 0.5)</td>
</tr>
<tr>
<td>Random</td>
<td>78.7 (\pm 0.8)</td>
<td>68.1 (\pm 0.6)</td>
<td>41.9 (\pm 0.6)</td>
</tr>
</tbody>
</table>

Table 3. Results on 3DPW [25] test set when different parts of virtual markers (VM) are occluded.

2.3. Additional Quantitative Results

Different loss terms. Table 2 reports the MPVE error on H3.6M [4] test set when ablating different loss terms. The confidence loss [5] is used to encourage the interpretability of the heatmaps to have a maxima response at the GT position. Without the confidence loss, the error increases slightly. If ablating the surface losses, MPVE increases a lot, which demonstrates the smoothing effect of these two terms.

Robustness to occlusion. We report results when different virtual markers are occluded by a synthetic mask in Table 3. The errors are slightly larger than the original image (None), which validates the effectiveness of the locality of the virtual marker representation. Occluding arm regions results in a larger error increase. This may be because the arm has larger variations in the dataset.

Comparison to fitting. In order to disentagle the ability of mesh regression from markers using \(\hat{A}\) with the ability to detect the virtual markers accurately from images, we first compute the estimation errors of the virtual markers. The MPJPE over all the virtual markers is 35.5mm, which demonstrates that these virtual markers can be accurately detected from the images. We then fit the mesh model parameters to these virtual markers. Table 4 shows the metrics of the fitted mesh on the SURREAL [24] test set. As we can
Table 4. Results on SURREAL [24] test set when the mesh is obtained by fitting to the estimated virtual markers.

<table>
<thead>
<tr>
<th>Method</th>
<th>MPVE ↓</th>
<th>MPJPE ↓</th>
<th>PA-MPJPE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fitting</td>
<td>44.6</td>
<td>34.8</td>
<td>29.5</td>
</tr>
<tr>
<td>Ours</td>
<td>44.7</td>
<td>36.9</td>
<td>28.9</td>
</tr>
</tbody>
</table>

see, the fitted mesh has a similar error as our regression ones which uses the interpolation matrix $\hat{A}$, which validates the accuracy of the estimated virtual markers.

2.4. Additional Qualitative Results

Figure 4 shows more qualitative comparisons with Pose2Mesh [2] on the SURREAL test set in which has diverse body shapes. The skeleton representation used in Pose2Mesh loses the body shape information so the method [2] can only recover mean shapes. For example, in Figure 4 (d) (e), the estimated meshes of Pose2Mesh tend to have the average body shape and fail to estimate the real body shape, regardless of whether the person is slim or stout. This is caused by the limited skeleton representation bottleneck so that the model learns a mean shape for the whole training dataset implicitly. In contrast, our approach with virtual marker representation generates more accurate mesh estimation results.

Figure 5 shows more qualitative comparisons with Pose2Mesh [2] and METRO [13] on the 3DPW test set. Pose2Mesh and METRO use the skeleton or all 3D vertices as intermediate representations, respectively. The estimated meshes are overlaid on the images according to the camera parameters. Pose2Mesh [2] has difficulty in estimating correct body pose and shapes when truncation occurs (a) or in complex postures (c). The results of METRO [13] have many artifacts where the estimated mesh is not smooth, and they also fail to align the image well. In contrast, our method estimates more accurate human poses and shapes and has smooth human mesh results. In addition, it is more robust to truncation and occlusion and aligns the image better.

Figure 6 shows more quality results of our approach on the 3DPW [25], H3.6M [4], MPI-INF-3DHP [19], and COCO [15] datasets. Figure 2 shows more qualitative results on Internet images with challenging cases, such as extreme body shapes or complex poses. Our method generalizes well on the natural scenes. Figure 3 shows typical failure cases, including inaccurate shape estimation due to the inaccurate virtual marker estimation around the arm when occluded, (b) Our method treats the lower arm of another person as its own due to occlusion. (c) Interpenetration around the right hand.

3. Human Subject Data

We use existing public datasets of human subjects in our experiments following their official licensing requirements. With proper usage, the proposed method could be beneficial to society (e.g. elderly fall detection).
Figure 4. Qualitative comparison between our method and Pose2Mesh [2] on SURREAL test set [24]. Our approach generates more accurate mesh estimation results on images of diverse body shapes.
Figure 5. Qualitative comparison between our method and Pose2Mesh [2], METRO [13] on 3DPW test set [25]. Our approach is more robust to occlusion and truncation and generates more accurate mesh estimation results that align images well.
Figure 6. Meshes estimated by our approach on images from the 3DPW [25] dataset (row 1-4), H3.6M [4] dataset (row 5), MPI-INF-3DHP [19] dataset (row 6), and COCO dataset (last 2 rows) [15].
References

[23] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In ECCV, pages 529–545, 2018. 2