Co-Evolution of Pose and Mesh for 3D Human Body Estimation from Video
–Supplemental Material–

Yingxuan You\textsuperscript{1} Hong Liu\textsuperscript{1}∥ Ti Wang\textsuperscript{1} Wenhao Li\textsuperscript{1} Runwei Ding\textsuperscript{1} Xia Li\textsuperscript{2}

\textsuperscript{1}Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University
\textsuperscript{2}Department of Computer Science, ETH Zürich

{youyx,tiwang}@stu.pku.edu.cn, {hongliu,wenhaoli,dingrunwei}@pku.edu.cn, xia.li@inf.ethz.ch

This supplemental material contains the following parts:
(A) The architecture of 3D pose estimation stream.
(B) Additional quantitative results.
(C) Additional ablation study.
(D) Details about loss functions.
(E) Additional visualization results.

A. Architecture of 3D Pose Estimation Stream

Figure A shows the detailed architecture of the 3D pose estimation stream. Firstly, the normalized 2D pose sequence is projected to high-dimensional joint features by a linear embedding layer. Secondly, we project and expand the static image features, which are added to their corresponding joint features in the same frame. Then we add the spatial and temporal embeddings to joint features and feed joint features to the spatial-temporal Transformer, which consists of cascaded spatial and temporal parts. In the spatial part, the spatial MSA calculates the similarities between joint tokens in the same frame. In the temporal part, the joint features are reshaped from $(T \times J \times C_1)$ to $(J \times T \times C_1)$, and thus the temporal MSA can calculate the similarities between frame tokens of the same joint. Finally, the joint features are regressed from $C_1$ to 3 and fused from $T$ frames to one frame to get the mid-frame 3D pose.

B. Additional Quantitative Results

Comparison with Single RGB-Based Methods. Table A compares our PMCE with single RGB-based methods on the 3DPW dataset. All methods use ResNet as the backbone. ‘†’ represents training w/o 3DPW training dataset. ‘∗’ represents training with 3DPW training set. The top two best results are highlighted in bold and underlined, respectively.

<table>
<thead>
<tr>
<th>Method</th>
<th>MPJPE ↓</th>
<th>PA-MPJPE ↓</th>
<th>PVE ↓</th>
<th>ACCEL ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMR [3]</td>
<td>130.1</td>
<td>76.7</td>
<td>-</td>
<td>37.4</td>
</tr>
<tr>
<td>GraphCMR [7]</td>
<td>-</td>
<td>70.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SPIN [6]</td>
<td>96.9</td>
<td>59.2</td>
<td>116.4</td>
<td>29.8</td>
</tr>
<tr>
<td>I2L-MeshNet [10]</td>
<td>93.2</td>
<td>57.7</td>
<td>110.1</td>
<td>30.9</td>
</tr>
<tr>
<td>PyMAF [13]</td>
<td>92.8</td>
<td>58.9</td>
<td>110.1</td>
<td>-</td>
</tr>
<tr>
<td>PARE [5]</td>
<td>82.9</td>
<td>52.3</td>
<td>99.7</td>
<td>-</td>
</tr>
<tr>
<td>ROMP+ [11]</td>
<td>79.7</td>
<td>49.7</td>
<td>94.7</td>
<td>-</td>
</tr>
<tr>
<td>METRO [1]</td>
<td>77.1</td>
<td>47.9</td>
<td>88.2</td>
<td>-</td>
</tr>
<tr>
<td>CLIFF+ [8]</td>
<td>72.0</td>
<td>45.7</td>
<td>85.3</td>
<td>24.7</td>
</tr>
<tr>
<td>PMCE (Ours)†</td>
<td>81.6</td>
<td>52.3</td>
<td>99.5</td>
<td>6.8</td>
</tr>
<tr>
<td>PMCE (Ours)+</td>
<td>69.5</td>
<td>46.7</td>
<td>84.8</td>
<td>6.5</td>
</tr>
</tbody>
</table>

Table B: Generalization evaluation in unseen views on Human3.6M. The test view is View 4.

<table>
<thead>
<tr>
<th>Training views</th>
<th>Only-pose model</th>
<th>PMCE</th>
<th>Improvements</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>161.7</td>
<td>82.9</td>
<td>78.8</td>
</tr>
<tr>
<td>1, 2</td>
<td>100.2</td>
<td>59.2</td>
<td>40.9</td>
</tr>
<tr>
<td>1, 2, 3</td>
<td>85.8</td>
<td>58.4</td>
<td>27.4</td>
</tr>
</tbody>
</table>

Table A: Comparison with single RGB-based methods. All methods use ResNet as the backbone. ‘†’ represents training w/o 3DPW training dataset. ‘∗’ represents training with 3DPW training set. The top two best results are highlighted in bold and underlined, respectively.

Parallelization in Unseen Views. Our method decouples 2D poses and image features from image sequences, which can not only provide complementary pose and shape information for better mesh estimation but also improve the generalization. To verify the latter, we compare our PMCE with

∥ Corresponding author.
Table C: Performance comparison between different initializations of mesh vertices on 3DPW.

<table>
<thead>
<tr>
<th>Mesh initialization</th>
<th>MPJPE ↓</th>
<th>PA-MPJPE ↓</th>
<th>PVE ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zeros</td>
<td>72.3</td>
<td>48.5</td>
<td>88.5</td>
</tr>
<tr>
<td>Template</td>
<td>71.4</td>
<td>47.6</td>
<td>86.2</td>
</tr>
<tr>
<td>Nearest joints (Ours)</td>
<td>69.5</td>
<td>46.7</td>
<td>84.8</td>
</tr>
</tbody>
</table>

the only-pose model (PMCE without using the image features) on the Human3.6M dataset. Specifically, based on the four camera views of Human3.6M, we train the networks on View 1, View 2, and View 3, then test them on the unseen View 4 to evaluate their generalization in unseen views. As shown in Table B, the only-pose model suffers from the domain gap between training and testing views, especially when training with few view data (top line). In contrast, our PMCE has better generalization ability and improves performance by a large margin. The results indicate that our method, complementing the pose information and image features, is effective for a robust mesh estimation in unseen views.

C. Additional Ablation Study

Impact of Mesh Initializations. Mesh initialization serves as a human body prior in our method. Table C examines the impact of different mesh initializations, including setting mesh vertices to zeros, using T-shape template mesh from SMPL [9] or setting the position of per mesh vertex as that of its nearest joint in estimated 3D pose $P_0$ (the distances between vertices and joints are pre-calculated from the template mesh and pose provided by SMPL). Compared with template mesh, setting vertices to their nearest joints makes the initialized mesh closer to the final mesh, which can provide a more precise human body prior and contribute to the final mesh performance.

D. Loss Functions

For the 3D pose estimation stream, we use the L1 joint loss to supervise the intermediate 3D pose $P_0$, which is defined as follows:

$$L_{int} = \frac{1}{J} \sum_{i=1}^{J} \| P_{gt} - P_0 \|_1.$$ (A)

After training the 3D pose estimation stream, we train the whole network using the following four loss functions.

Mesh Loss. We use the L1 loss between the ground truth 3D mesh vertices $M_{gt}\in\mathbb{R}^{V \times 3}$ and the predicted 3D mesh vertices $M\in\mathbb{R}^{V \times 3}$. The mesh vertex loss is calculated as:

$$L_{mesh} = \frac{1}{V} \sum_{i=1}^{V} \| M_{gt} - M \|_1.$$ (B)

Joint Loss. We multiply the predicted 3D mesh $M$ by a pre-defined matrix $\mathcal{J}\in\mathbb{R}^{J \times V}$ to obtain the regressed 3D joints and calculate the joint loss with ground truth 3D joints $P_{gt}$:

$$L_{joint} = \frac{1}{J} \sum_{i=1}^{J} \| P_{gt} - \mathcal{J} M \|_1.$$ (C)

Surface Normal Loss. This loss is used to improve surface smoothness and local details. It is calculated by the normal vectors of the ground truth mesh and the predicted mesh:

$$L_{normal} = \sum_{f} \sum_{\{i,j\} \in f} \left| \frac{m_i - m_j}{\| m_i - m_j \|_2}, n_{gt} \right|.$$ (D)

where $f$ denotes a triangle face in the mesh, $m_i$ and $m_j$ denote the $i_{th}$ and $j_{th}$ mesh vertices of the triangle face respectively. And $n_{gt}$ is the unit normal vector of the triangle face $f$ in the ground truth mesh.

Surface Edge Loss. This loss is used to improve the smoothness of the areas with dense vertices, e.g., hands and feet. The edge length consistency loss is calculated by the ground truth edges and the predicted edges as:

$$L_{edge} = \sum_{f} \sum_{\{i,j\} \in f} \| m_{gt_i} - m_{gt_j} \|_2 - \| m_i - m_j \|_2.$$ (E)

Given the four loss functions, the final loss is calculated as the weighted sum:

$$L = \lambda_m L_{mesh} + \lambda_j L_{joint} + \lambda_n L_{normal} + \lambda_e L_{edge},$$ (F)

where $\lambda_m=1$, $\lambda_j=1$, $\lambda_n=0.1$, $\lambda_e=20$ in the experiments.
Figure B: Qualitative comparison between MPS-Net [12] and our PMCE. For each video sequence, the top rows show the video frames, the middle rows show the predicted mesh results from our PMCE (blue), and the bottom rows show the mesh results from MPS-Net (pink). Our method can produce more accurate and smooth 3D human motion in fast motions (first sequence), occlusions (second sequence), and slight pose changes (last sequence).
E. Additional Visualization Results

Qualitative Comparison. Figure B shows the qualitative comparison between the previous state-of-the-art video-based method MPS-Net [12] and our PMCE on the challenging video sequences. It shows that our method can produce more accurate and temporally consistent mesh results, especially in fast motions, occlusions, and delicate body deformations.

Visualization of Attention Maps. We further study the interactions of pose and mesh in the proposed co-evolution decoder, including Mesh → Pose, Pose → Mesh, Mesh → Mesh, and Pose → Pose interactions. We obtain the above four kinds of attention maps from the last layer of the co-evolution decoder by averaging the attention values of all attention heads in their corresponding attention blocks. Figure C shows the visualization of attention maps taking different reference nodes. In ‘Mesh →
Pose’ interaction (Col. 3), each joint can obtain the global shape information from vertices which is not available in its original pose representations. In ‘Pose → Mesh’ interaction (Col. 4), each mesh vertex aggregates pose information that can guide the mesh deformation. And in ‘Mesh → Mesh’ (Col. 5) and ‘Pose → Pose’ (Col. 6) interactions, vertices and joints perform internal adjustments, respectively.

References


