MoCap-to-Visual Domain Adaptation for Efficient Human Mesh Estimation from 2D Keypoints

Supplementary Material

Figure 5. **Qualitative results on the H3.6M dataset [16].** For each sample, the first column displays the input image and 2D keypoint detections, the second column shows the SMPL mesh overlaid on the image, and the third column presents the SMPL mesh from the side view. Domain-adapted model is used to generate these qualitative results.

6. Qualitative Results

We showcase qualitative results on the H3.6M [16] dataset in Fig. 5. Moreover, we provide qualitative video results on the project website: https://key2mesh.github.io/. Our approach involves applying the Key2Mesh (adapted) model to each individual video frame, with predictions presented without any smoothing. As there are no 2D keypoints on hands and feet in the OpenPose detections, we occasionally observe jittery artifacts on the extremities. Incorporating additional 2D keypoints on the extremities could be addressed in future work.

7. Further Implementation Details

Model Architecture Details. Fig. 6 illustrates the architecture of our model, consisting of three key components: the Feature Extractor, the SMPL Head, and the Domain Critic. For each component, we adopted a standard multi-layer perceptron (MLP) architecture, incorporating skip connections as shown in the figure. The linear layers were equipped with 1024 neurons, and we introduced batch normalization after each linear layer, except in the case of the Domain Critic. In the early stages of our experiments, we observed that excluding the normalization layer inside the Domain Critic led to better performance. We utilize parametric Rectified Linear Unit (pReLU) and Dropout layers, setting the dropout probability to 0.2.

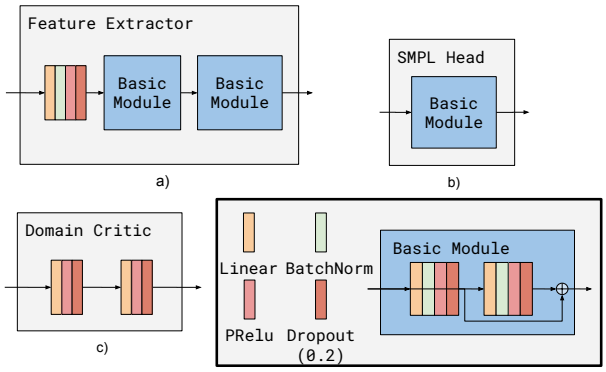

Figure 6. **Overview of Key2Mesh model architecture.**

Model Selection. While adapting to the test dataset, selecting a model becomes less straightforward due to the absence of 3D labels. As a result, benchmark metrics like PA-MPJPE, MPJPE, and PVE cannot be calculated during evaluation and model selection. To obtain the models adapted to the test sets of H3.6M and 3DPW, as depicted in the final rows of Tab. 1 and Tab. 2, we employ the training subjects from the H3.6M dataset and the validation split from 3DPW during the model selection process. We calculate PA-MPJPE every 500 training steps on these sets and checkpoint the model that achieves the best PA-MPJPE during this process.

8. Training Data Sampling from Unpaired 3D Human Body Data

In pre-training, we sample a 3D human body encoded with SMPL [30] from the MoCap domain and apply a range of augmentations. For a given SMPL sample, we follow LGD's [44] augmentation pipeline and apply random global rotations to simulate different views by using yaw angles drawn uniformly from the range of -180° to $+180^\circ$, and roll and pitch angles sampled uniformly within -20° to 20°. We also randomly occlude body joints with 20% probability. In addition to LGD's augmentation pipeline, we try to simulate the jitter introduced by the pose estimator on the visual data for each keypoint (x). To achieve this, we perturb keypoints randomly: $x = x + \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$.

9. Using Different 2D Pose Estimators

2D Pose Estimator	PA-MPJPE↓	MPJPE↓
OpenPose [2]	51.4	108.1
ViTPose [51]	**49.3**	**104.2**

Table 6. **Comparing the use of OpenPose [2] and ViTPose [51] as 2D pose estimators in the pipeline.** We report PA-MPJPE and MPJPE (both in mm) based on the H3.6M evaluation subjects after applying our domain adaptation stage using H3.6M training subjects.

In table Tab. 6, we compare the performance of using OpenPose [2] and ViTPose [51] as 2D pose estimators in the pipeline. We observe that utilizing ViTPose results in lower PA and MPJPE scores compared to OpenPose. This underscores the potential for incorporating different pose estimators into Key2Mesh's pipeline, with a better-performing pose estimator leading to improved SMPL estimation accuracy.

10. Impact of the Domain Adaptation on the Features

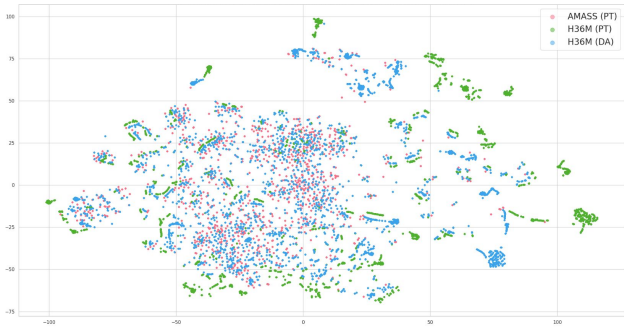

Figure 7. **t-SNE visualization of features extracted by pre-trained and domain-adapted feature extractors on H3.6M.** Best viewed when zoomed in.

The domain-adapted feature extractor is constrained to produce features that mimic those of the source domain when the domain of the input 2D poses changes. As illustrated in Fig. 7, the t-SNE plot shows that features generated by the domain-adapted feature extractor on the H3.6M dataset (blue) exhibit improved alignment with the source domain, as represented by features generated by the pre-trained feature extractor on the AMASS dataset (pink), compared to features produced solely by the pre-trained feature extractor (green). This aligns with our intuition regarding domain adaptation, as it improves the performance of SMPL-Head when transitioning between domains by operating on features that more closely resemble those from its training set.

11. Failure Scenarios

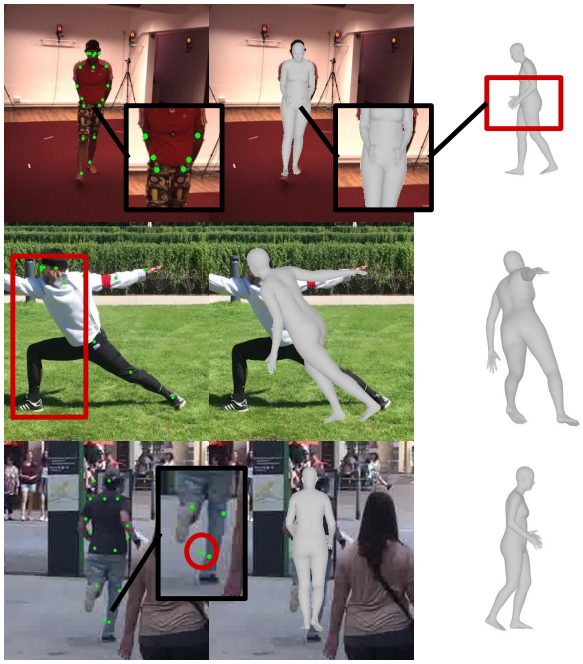

Figure 8. **Some failure cases.** In the first row, the algorithm encounters challenges related to depth ambiguity, leading to inaccurate hand position detection. In the second and third rows, Key2Mesh exhibits issues with missing or incorrect keypoint detection. The model's exclusive reliance on 2D keypoints renders it susceptible to errors in cases where keypoints are either absent or detected erroneously.

Fig. 8 illustrates various cases that highlight the limitations of Key2Mesh.

12. Limitations and Future work

While our pipeline shows promising performance in predicting 3D human pose and shape from 2D keypoints, inherent ambiguity in representing 3D pose and shape through 2D keypoints poses fundamental challenges. Future work might involve integrating temporal information to enhance performance.