Supplementary Material

Appearance Consensus Driven Self-Supervised Human Mesh Recovery

In this supplementary, we summarize the proposed differentiable colored-mesh recovery procedure followed by additional implementation details and qualitative results. Follow our project page[1] for more details.

The supplementary material is organized as follows:

- Section 1: Differentiable operations in the proposed framework
- Section 2: Sampling image pairs with diverse background
- Section 3: Reflectional symmetry groups
- Section 4: Qualitative evaluation

Table 1. A list of notations and their size as used in the main paper.

Notations	Description	Size
I	Input image	$224 \times 224 \times 3$
θ	View invariant SMPL pose	3J (J=23)
β	SMPL Shape parameter	10
ϕ	Pose embedding	32
V	3D vertex locations	6890×3
C	Vertex colors (RGB)	6890×3
v	Image-projected vertex locations	6890×2
N	Z-component of Camera-space normals	6890×1
Z	Camera-space depth of mesh vertices	6890×1
$I^z(u)$	Rendered depth image	$224 \times 224 \times 3$
W	Visibility-aware-weighing	6890×1
$\tilde{C}$	Intermediate Vertex colors (RGB)	6890×3
S	Vertex to symmetry group mapping	(1575+295)×6890
$\mathcal{C}$	Group color for symmetry groups	1870×3
Q	Vertex to part-segmentation mapping	-
H	*Conv2-1* output of pre-trained VGG-16	$112 \times 112 \times 128$
$\mathcal{H}^k$	Sampled feature from H at $v^{(k)}$	$\tilde{d} = 128$
$\mathcal{F}$	Part-prototype appearance feature	128
k	Index over mesh vertices	K=6890
g	Index over symmetry groups	G=1870
l	Index over body parts	L=14
a, b	Indicating association with inputs I_a, I_b	-

1 Differentiable operations in the proposed framework

We propose three completely differentiable modules in order to realize our self-supervised approach namely the color-recovery module, part-prototype module

[1] Project-page: https://sites.google.com/view/ss-human-mesh

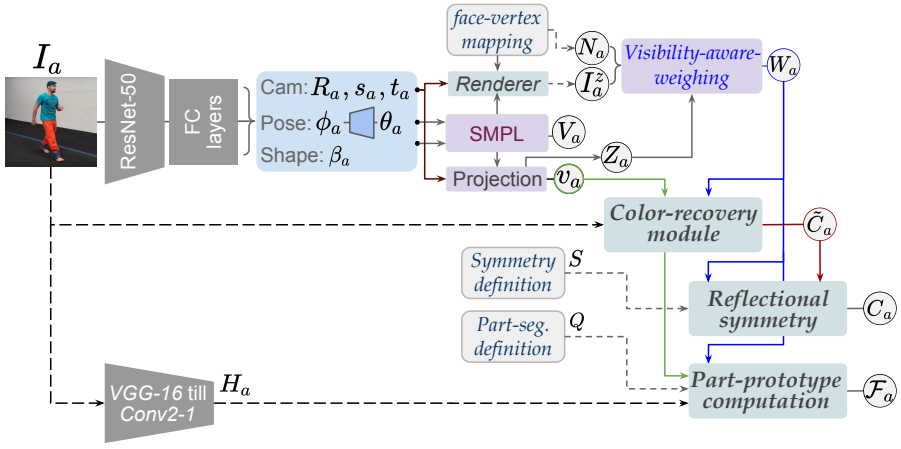

Fig. 1. The series of differentiable computations and their interdependence as employed in the proposed self-supervised mesh recovery framework (see Table 1 for the notations).

and the reflectional symmetry module. See Fig. 1 for an illustration of the differentiable computations and their interdependence.

a) Obtaining *visibility-aware-weighing*, W: All the modules use a differentiable *visibility-aware-weighing*, W to softly segregate the 3D vertices based on their visibility for a given (or predicted) camera view. The computation of W relies on the fact that visible vertices are influenced by two factors (i) camera and (ii) human skeleton self-occlusion. We identify camera facing vertices using the z-component of the Normal (N) while we handle human self-occlusion by soft selection based on a camera-centric depth image and a margin in z-buffering.

$$W^{(k)} \in [0,1] = \exp(-\alpha D^{(k)})\ \sigma(\gamma N^{(k)}), \text{ where } D^{(k)} = |I^z(v^{(k)}) - Z^{(k)}|$$

b) Recovering intermediate color, $\tilde{C}$: Next, we obtain the intermediate, visibility-aware colors, $\tilde{C}$ by weighting the raw picked colors (done by bilinear sampling of image I, given the vertex 2D projection $v^{(k)}$) as shown below.

$$\tilde{C}^{(k)} = I(v^{(k)})\ (2W^{(k)} - 1), \text{ where } I(v^{(k)}) \text{ denotes RGB color at the } v^{(k)}$$

c) Applying reflectional symmetry to obtain the full vertex color, C: Next, we focus on propagating the color intensities from the visible vertices (as stored in the intermediate $\tilde{C}$) to the invisible ones (*i.e.* the vertices having low $W^{(k)}$). To realize a fully-colored mesh C, we use a predefined, 4-way symmetry grouping knowledge (front-back and left-right) as stored in S. First the group colors $\mathcal{C}^{(g)}$ are computed as a normalized combination of the intermediate vertex colors weighted by their visibility weighing W. Then, the group colors are directly propagated to all the mesh vertices using S as shown in the following equation.

$$C = S^T * \mathcal{C}, \text{ where } \mathcal{C}^{(g)} = (S^{(g)} \circ \text{ReLU}(\tilde{C}))/(S^{(g)} \circ \text{ReLU}(2W - 1))$$

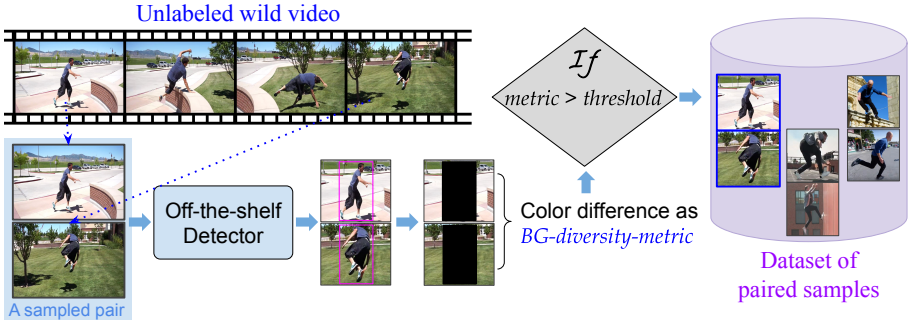

Fig. 2. An illustration of the adopted procedure to sample image pairs of diverse background. To built the dataset of image pairs as required by the proposed self-supervised framework, we chose image pairs depicting the same person in diverse pose (maintaining a considerable temporal gap) and background (via *BG-diversity-metric*).

d) Computation of part-prototype features, $\mathcal{F}$: Here, we reuse the color-recovery idea to realize part-prototype features. $\mathcal{F}^{(l)}$ is computed as the normalized weighted sum, of the recovered spatial features $\mathcal{H}^{(k)} = H(v^{(k)})$, over the vertices belonging to the part l (using vertex to part-segmentation mapping, Q).

$$\mathcal{F}^{(l)} = (\Sigma_{k \in Q^{(l)}} W^{(k)} \mathcal{H}^{(k)}) / (\Sigma_{k \in Q^{(l)}} W^{(k)}), \text{ where } \mathcal{H}^{(k)} = H(v^{(k)})$$

2 Sampling image pairs with diverse background

Given a video clip depicting actions of a single person, in consistent apparel, we aim to sample image pairs which would have diverge background (BG) appearance. To realize this, we first prune the video frames using an off-the-shelf person-detector [3] to obtain a reliable human-centric crop as required for the mesh estimation pipeline. Following this, we compute *L2* distance (mean squared error) between image pairs, only for the regions outside the detector box, to obtain a *BG-diversity-metric* (see Fig. 2). Among all possible frame pairs (beyond 1 sec temporal gap), we choose the pairs having *BG-diversity-metric* greater than a certain threshold value. In contrast to the in-studio datasets with hardly any camera movement implying static BG [1], our in-house collection of YouTube videos have diverse camera movements (*e.g.* Parkour and Free-running videos). The wild camera movement inherently results in huge diversity in the sampled image pairs. Note that, in static camera scenarios BG diversity occurs when the person moves from one location to another. This is because, instead of taking the full video feed, we consider a square region around the detector output as the effective input to the CNN regressor.

3 Reflectional symmetry groups

We define reflectional groups where each group constitutes a set of vertices which is assumed to have similar color property. Though this assumption does not hold

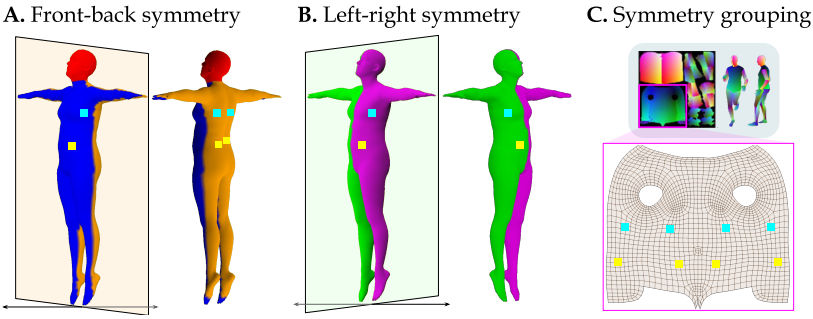

Fig. 3. An illustration of front-back and left-right symmetry. This is used to define the multi-hot encoding, $S^{(g)} \in \{0,1\}^K$ which constitutes of four ones indicating vertex members in the symmetry group g. Here, "yellow" and "cyan" color patches show rough location of the vertices for two such symmetry groups. Note that, for the head region only left-right symmetry is used (red colored region in panel A).

true in presence of illumination difference and non-symmetric apparel design, we find this to be helpful in general because of the following reasons. Firstly, it is rare to encounter non-symmetric apparel with diverse color difference between the left-right or front-back. Secondly, although the luminosity property (*i.e.* intensity) is influenced in presence of illumination difference, the color property (*i.e.* hue) remains comparable. However, the consistency loss on the part-prototypes and also on the intermediate vertex color $\tilde{C}$ helps us to effectively balance this shortcomings. Broadly, we define 2 types of symmetry groups; a) group sets of 2 members (vertex indices) only for the head region (295 groups), and b) group sets of 4 members (vertex indices) for rest of the body parts (1575 groups). See Fig. 3 for a rough illustration. 4-membered groups are obtained by applying both front-back and left-right symmetry. However, 2-membered groups represent only left-right symmetry. Note that all the group sets are mutually exclusive and exhaustive, *i.e.* 2*295 + 4*1575 = 6890, where 6890 is the total number of vertices. This symmetry knowledge is stored as a multi-hot encoding denoted as $S^{(g)} \in \{0,1\}^K$ which constitutes of four ones indicating vertex members in the symmetry group g. All the symmetry groups are combined in a symmetry-encoding matrix represented as $S \in \{0,1\}^{G \times K}$. This multi-hot symmetry group representation helps us to perform a fully-differentiable vertex color assignment for all the vertices including the occluded and non-camera facing ones.

4 Qualitative evaluation

In order to evaluate the generalizability of our model, we visualize our model's 3D pose and shape performance on a variety of images sampled from different datasets. Fig 4 shows the predicted colored mesh and the corresponding 3D pose in aligned grid plots. Fig. 5 shows a qualitative analysis on the standard 6-part mesh overlay. Here, mesh overlay can be considered as a proxy to evaluate both shape and pose in a collective fashion.

Fig. 4. Qualitative results on single image colored human mesh recovery.

Although, pose and shape are predicted correctly, background leaks could occur at misaligned locations that can be visualized using the coloured mesh reconstructions as shown in Fig 4. Also note that, SMPL [2] does not parameterize hand pose hence hands remain in a fixed mean pose (flat open hand). This tends to be a consistent location for background leakage. Background leakages are observed to generally occur at boundaries of hands and feet; *e.g.*, in row 2 last column of Fig 4 the green background leaks onto the appearance of the hand due to the limitation of the parametric human model in articulating the exact hand pose. Also, our model outputs sub-optimal results in cases with complex inter-limb occlusions as highlighted in magenta in Fig. 5.

Fig. 5. Qualitative results. In each panel, 1st column depicts the input image, 2nd column shows the model-based part segments on **A.** Human3.6M (in-studio), **B.** 3DPW (in-the-wild) **C.** LSP (in-the-wild) **D.** YouTube (in-the-wild). The model fails in presence of complex inter-limb occlusions (in magenta box).

References

1. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence (2013) 3
2. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM transactions on graphics (2015) 5
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NeurIPS. pp. 91–99 (2015) 3