Hand Avatar: Free-Pose Hand Animation and Rendering from Monocular Video – Supplementary Material

Xingyu Chen Baoyuan Wang Heung-Yeung Shum
Xiaobing.AI

I. MANO-HD

MANO. MANO can be driven with parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ and $\boldsymbol{\theta} \in \mathbb{R}^{B\times 3}$ ($B = 16$ indicates the number of per-bone parts), where $\boldsymbol{\beta}$ is the coefficients of a shape PCA bases while $\boldsymbol{\theta}$ represents joint rotations in axis-angle form. Mean template mesh is deformed to match different shapes:

$$\begin{aligned}\tilde{\mathbf{V}} &= \bar{\mathbf{V}} + \mathcal{B}_s(\boldsymbol{\beta}) + \mathcal{B}_p(\boldsymbol{\theta})\\ \mathbf{J} &= \mathcal{J}(\bar{\mathbf{V}} + \mathcal{B}_s(\boldsymbol{\beta})),\end{aligned} \tag{I}$$

where $\bar{\mathbf{V}}, \mathcal{B}_s, \mathcal{B}_p$ are template vertices and shape/pose blendshapes. Canonical joint locations $\mathbf{J} \in \mathbb{R}^{B\times 3}$ are given with the regressor $\mathcal{J}$.

Then, bone transformation matrix $\mathbf{G}_b \in \mathbb{R}^{4\times 4}$ is computed along the kinematic chain $\mathcal{K}$ with the Rodriguez formula $\mathcal{R}$:

$$\mathbf{G}_b(\boldsymbol{\theta}, \mathbf{J}) = \prod_{j\in\mathcal{K}(b)} \left[\begin{array}{c|c} \mathcal{R}(\boldsymbol{\theta}_j) & \mathbf{J}_j \\ \hline \mathbf{0} & 1 \end{array}\right] \tag{II}$$

Finally, linear blend skinning is used to pose vertices with skinning weights $\mathbf{W} \in \mathbb{R}^{V\times B}$ (V denotes the number of vertices) as follows,

$$\mathbf{V}_i = \sum_{b=1}^{B} \mathbf{W}_{b,i}\mathbf{G}_b(\boldsymbol{\theta}, \mathbf{J}) G_k(\mathbf{0}, \mathbf{J})^{-1}\tilde{\mathbf{V}}_i. \tag{III}$$

Optimization of MANO-HD. Following [1], we subdivide the MANO template mesh to obtain a high-resolution version with 12,337 vertices and 24,608 faces. Nevertheless, articulated dynamic mesh subdivision is a non-trivial task, and mesh skinning operation is likely to introduce artifacts to deformed mesh. Thus, we optimize upsampled skinning weights $\mathbf{W}^{HD} \in \mathbb{R}^{V^{HD}\times B}$ to eliminate dynamic artifacts under various hand poses using energy functions as follows,

$$\begin{aligned}&\mathcal{L}_{l_0} = \sum_{i=1}^{B\cdot V^{HD}} (1 - e^{-\eta \mathbf{W}_i^{HD}})\\ &\mathcal{L}_{lap} = \frac{1}{V^{HD}} \sum_{i=1}^{V^{HD}} \sum_{j\in\mathbb{N}(i)} \frac{1}{\omega} ||\mathbf{V}_i^{HD} - \mathbf{V}_j^{HD}||_2\\ &\mathcal{L}_{surf} = \mathrm{Cham}(\mathbf{V}^{HD}, \mathbf{F}),\end{aligned} \tag{IV}$$

Method	*Lap.*	*Cham.*	l_0 norm (%)
MANO	23.31	-	16.29
MANO-HD w/o $\mathbf{W}^{HD}$ opt.	1.923	7.014	17.97
MANO-HD w/o $\mathcal{L}_{l_0}$	1.576	7.170	44.15
MANO-HD	1.753	7.039	16.89

Table I. Effects of the $\mathbf{W}^{HD}$ optimization (opt.).

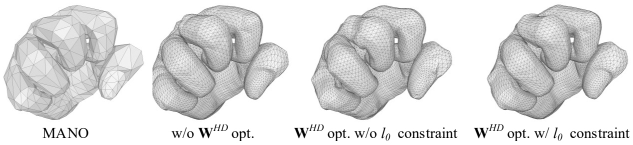
MANO
w/o $\mathbf{W}^{HD}$ opt.
$\mathbf{W}^{HD}$ opt. w/o l_0 constraint
$\mathbf{W}^{HD}$ opt. w/ l_0 constraint

Figure I. Hand mesh comparison under a large deformation.

where $\mathcal{L}_{l_0}$ is approximated l_0 norm constraint [2] to produce sparse skinning weights. $\mathcal{L}_{lap}$ is the Laplacian term for mesh smoothness, where $\mathbb{N}(\cdot)$ represents vertex neighborhood and ω is the normalization factor. The function $\mathrm{Cham}(\cdot,\cdot)$ computes the chamfer distance between MANO-HD mesh vertices $\mathbf{V}^{HD}$ and the MANO mesh faces $\mathbf{F}$. The overall energy function is given as $\mathcal{L}_{HD} = \lambda_{l_0}\mathcal{L}_{l_0} + \lambda_{lap}\mathcal{L}_{lap} + \lambda_{surf}\mathcal{L}_{surf}$ with balance term λ.

Implement Details of MANO-HD For $\mathbf{W}^{HD}$ optimization, we adopt the original MANO dataset [4] with 1,554 pose parameters for training and evaluation. We randomly compose and interpolate finger-level rotations for data augmentation. The training process lasts 3,000 steps with a batch size of 1,024. The learning rate begins at 10^{-5} and decreases with exponential decay. We use $\mathcal{L}_{HD}$ as the objective and adjust hyperparameters to balance multiple energy terms as $\eta = 100, \lambda_{l_0} = 0.01, \lambda_{lap} = 1, \lambda_{surf} = 10$.

Effects of MANO-HD We use Laplacian smoothing *Lap.* and chamfer distance *Cham.* to reflect the smoothness and accuracy of MANO-HD, whose definition is the same as $\mathcal{L}_{lap}$ and $\mathcal{L}_{surf}$ in Eq. IV. Both *Lap.* and *Cham.* are presented in 10^{-4}m. As the MANO surface is not smooth enough, *Lap.* and *Cham.* cannot be simultaneously improved. Besides, we introduce l_0 norm as the metric, which

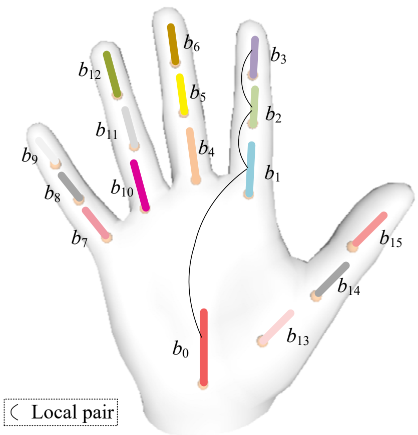

Figure II. Kinematic tree and part indices of the hand. We also show local pairs along the kinematic chain of forefinger.

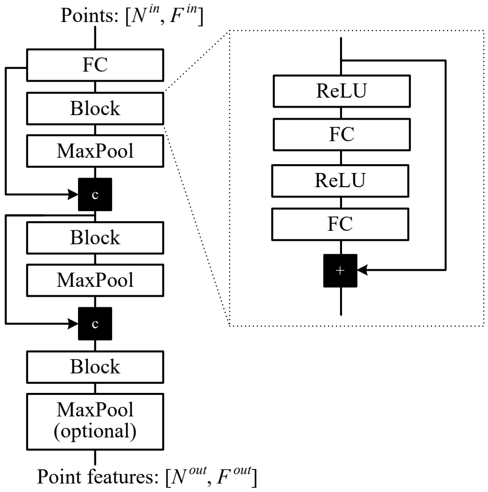

Figure III. PointNet structure. "c,+" indicate concatenation and element-wise sum; "FC" denotes fully connected layer.

is defined as the proportion of non-zero elements in the skinning weights.

Referring to Tab. I, mesh subdivision can improve *Lap.*, yet incurs artifacts during skinning as shown in the part-connection regions in Fig. I. Moreover, $\mathbf{W}^{HD}$ optimization without $\mathcal{L}_0$ cannot counteract the issue despite inducing lower *Lap.* value. The reason behind this is the poor sparsity of $\mathbf{W}^{HD}$. After $\mathcal{L}_{l_0}$-based optimization, the l_0 norm of $\mathbf{W}^{HD}$ is on par with that of MANO, and the skinning performance is improved (see Fig. I).

II. Network Structures

Kinematic Tree and Local Pair. Following the definition of MANO [4], Fig. II shows the bone indices and connections for the hand. In addition, we demonstrate our defined local pairs.

Name	Depth	Width	Input size	Output size
$\mathcal{M}_{shape}$	4	128	84	3
MLP in $\mathcal{Q}_{pair}$	4	128	64	1
$\mathcal{M}_{albedo}$	4	256	128	3
$\mathcal{M}_{illum}$	4	256	85	1

Table II. The details of MLPs.

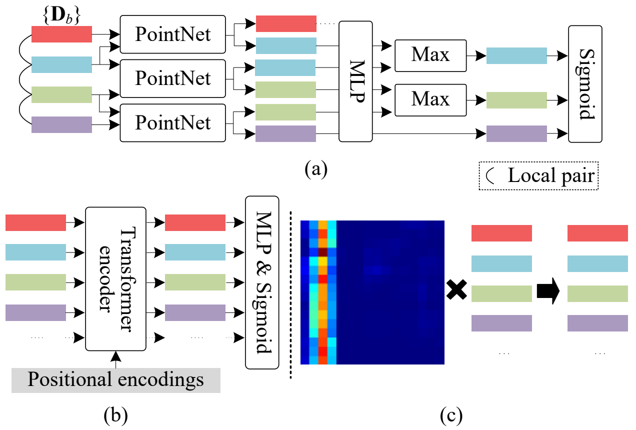

Figure IV. (a) Our proposed local-pair decoder. Multiple PointNets/MLPs share weights. (b) Transformer-based decoder for comparison. (c) Attention-based fusion of part-level encodings in Transformer for bone part b_2. Consistent with Fig.II, colors distinguish part-level geometry encodings. For visual conciseness, we only show the process of forefinger.

PointNet Structure. The PointNet used in part-space encoder $\mathcal{Q}_{part}$ and local pair decoder $\mathcal{Q}_{pair}$ is shown in Fig III, where N, F denote point amount and feature size. For $\mathcal{Q}_{part}$, $N^{in} = 256, F^{in} = 6, N^{out} = 1, F^{out} = 64$. For $\mathcal{Q}_{pair}$, we get rid of the last MaxPool, leading to $N^{in} = 2, F^{in} = 64, N^{out} = 2, F^{out} = 64$.

MLP Details. Referring to Table II, we list MLP details used in this paper.

Local-Pair Decoder. The local pair decoder $\mathcal{Q}_{pair}$ contains a PointNet and an MLP, whose detailed structures are introduced before. Furthermore, we illustrate how the $\mathcal{Q}_{pair}$ processes the geometry encodings along the kinematic chain of forefinger, as shown in Fig. IV(a). First, locally paired encodings are treated as two *points* and fused by the PointNet. Without across-point maxpooling, the shape of PointNet output remains the same as its input. Then, the MLP maps PointNet outputs to the occupancy domain. As a result, a bone part could have multiple occupancy values from multiple local-pair predictions, which are fused by a maximum operator. Because of the feature

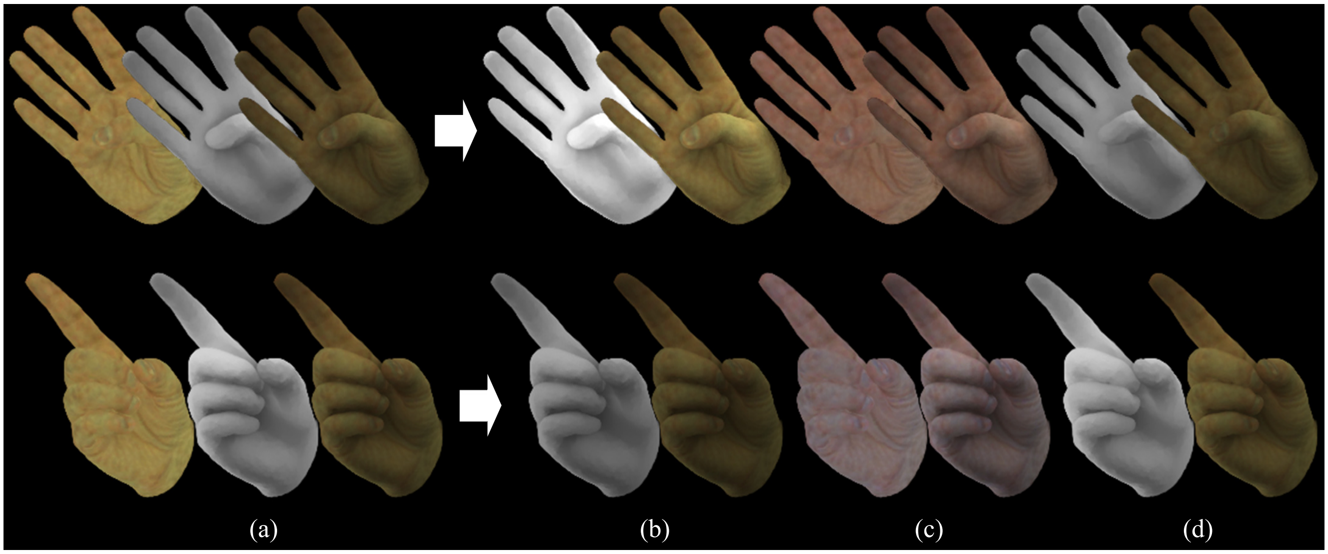

Figure V. Three groups of texture editing. (a) Reconstruction results. From left to right: albedo, illumination, and shaded image. (b) Lighting editing. (c) Albedo editing. (d) Shadow editing.

Capture	Training set	Validation set
test/Capture0	11,757	194
test/Capture1	18,474	232
val/Capture0	18,340	184

Table III. Data amount for training and validation sets.

fusion by the PointNet, part boundaries are blurred and extend to the connection direction. Hence, the maximum operator is used to produce a union of boundary-extended part geometries. Finally, the sigmoid function is employed for occupancy normalization.

Transformer-Based Decoder. To validate the local-pair design, we develop a learning-based method with Transformer. As shown in Fig. IV(b), all part-level encodings are fed to the Transformer encoder without any inductive prior. Through 4 self-attention blocks, the Transformer can perform adaptive feature fusion. Referring to Fig. IV(c), the attention map determines the way to select important features, where the effect is consistent with our local-pair decoder. That is, bone parts b_1, b_2, b_3 are connected in the attention map for evolving the encoding of part b_2.

III. Training Details

Video Data. For quantitative evaluation, we select three sequences from the InterHand2.6M dataset [3]. Data amount is shown in Table III, where training data are from the *ROM04_RT_Occlusion* sequence and validation data are from the *ROM03_RT_No_Occlusion* sequence. Because video frames are highly redundant, validation data are selected by fixed skip steps, and we adjust the steps to assure various hand poses and global rotations can be covered.

For each frame, we crop the hand region with annotated detection boxes as the ground truth. First, the box is regulated as a square box with 1.3 times expansion. Then, the hand region is cropped and resized to 256×256 resolution.

Training Settings. For PairOF pre-training, the learning rate begins at 5×10^{-4} and decreases with exponential decay. The training process has 270K steps with a batch size of 32.

For end-to-end training, the learning rate begins at 5×10^{-4} and decreases with exponential decay. Due to the pretraining, the learning rate of $\mathcal{Q}_{pair}$ is set to be 10 times smaller. Following [5], we use a patch strategy for training with a patch size of 32×32. The training process has 50K steps with a batch size of 16.

IV. Texture Editing

Referring to Fig V, our HandAvatar supports hand texture editing. Firstly, we change the illumination field by a 1.5- or 0.8-times multiplication, as shown in Fig V(b). Then, we shift the mean RGB value of the albedo field, leading to the results of Fig V(c). Besides, we edit shadow in Fig V(d). In the top row, the texture is induced by letting thumb-related directed soft occupancy values equal 0. As a result, the self-occluded shadow patterns on the palm become weakened. In the bottom row, we remove the shadow patterns between the middle and ring fingers by setting the directed soft occupancy values as 0 for bone parts b_5 and b_{11}. The shadow editing results also validate the effect of directed soft occupancy.

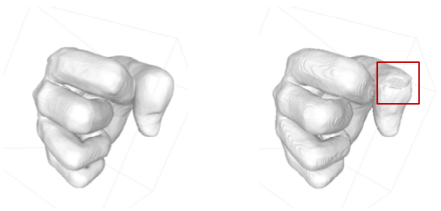

Figure VI. PairOF (left) *vs*. Transformer-based method (right)

V. Discussion

The Disentanglement of Albedo and Illumination. The disentanglement of albedo and illumination does not require extra regularization. Albedo is known to be independent of hand pose, while illumination depends on hand pose. In SelF, the input of the albedo field is unrelated to the hand pose, while pose-relevant elements are fed into the illumination field. Therefore, with various hand poses as training data, the optimization process would ensure that the illumination is free from the albedo field. In addition, the illumination field outputs a scalar, which cannot model RGB-based (3-channel) albedo.

PairOF *vs*. Transformer-Based Method The motivation of PairOF is to fuse part-level geometry encodings to eliminate the shape inconsistency in the area of part connections. Despite similar numerical results in Tab. 2 of the main text, our PairOF is more effective in feature fusion than the Transformer-based method. That is, hand bone connections are unchangeable, and PairOF can use this prior knowledge, leading to a more effective feature fusion than self-attention. As a result, self-attention cannot visually achieve our motivation as shown in Fig. VI. Also, we are more efficient in terms of learning, *i.e*., the convergence of the Transformer-based method is slower and more data-hungry.

VI. Limitations and Future Works

Besides the limitation discussed in the main text, HandAvatar can be further improved from the following perspectives in the future. First, full lighting editing is worthy of future research. For example, despite the lighting editing as shown in Fig V(b), it is hard to edit or add a point light in the illumination field. Second, the representation of specular effects in hand appearance is another interesting topic. To achieve this goal, hand surface properties with BRDF should be explored. Third, the hand geometry demonstrated in Fig. 5 in the main text is the result of PairOF pre-training. After end-to-end training with texture losses, the PairOF could produce a non-smooth surface with geometry wrinkles. This is caused by the hand-pose annotation error of InterHand2.6M. As a result, the PairOF could produce fragile hand geometry to compensate for this error.

References

[1] Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular RGB videos. In *CVPR*, 2022.

[2] Yuantao Gu, Jian Jin, and Shunliang Mei. l_0 norm constraint lms algorithm for sparse system identification. *IEEE Signal Processing Letters*, 16(9):774–777, 2009.

[3] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. Interhand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *ECCV*, 2020.

[4] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM TOG*, 2017.

[5] Chung-Yi Weng, Brian Curless, Pratul P Srinivasan, Jonathan T Barron, and Ira Kemelmacher-Shlizerman. HumanNeRF: Free-viewpoint rendering of moving people from monocular video. In *CVPR*, 2022.