MANUS: Markerless Grasp Capture using Articulated 3D Gaussians

Supplementary Material

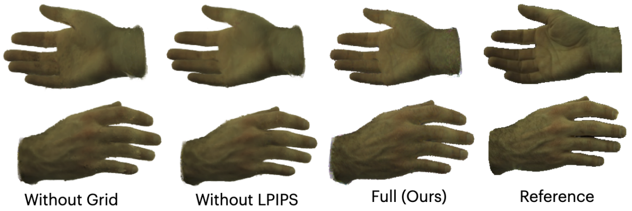

Figure 1. **Hand Ablation**: We perform ablation on the grid initialization of the skinning weights and the choice of LPIPS loss function. Clearly our approach is better in terms of visual appearance.

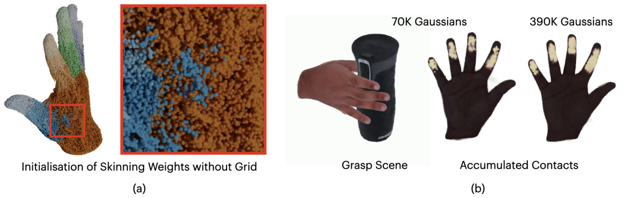

Figure 2. Here in **(a)** we show how initializing MANO weights without voxel grid allows the unstructured Gaussians to move erratically. In **(b)**, we show the affect on accumulated 2D contact renderings with change in the number of Gaussians.

1. Ablation Study

1.1. MANUS-Hand

Initialization of Skinning Weights: We observe that the choice of method used to initialize skinning weights significantly influences the performance of our hand model. As demonstrated in Figure 2 (a), initializing skinning weights directly onto Gaussians using a nearest neighbor approach, as opposed to grid initialization, leads Gaussians to move erratically and shift towards an unrelated bone. Consequently, this misalignment results in artifacts, where skinning weights are incorrectly allocated to the wrong bone, causing the position to be associated with the incorrect bone. The impact of this method of initialization is presented both quantitatively and qualitatively in Table 1 and Figure 1.

Ablation on LPIPS loss: We observed that LPIPS loss improves the quality of renderings and maintain consistency across views. We also demonstrate that LPIPS loss function improves the overall visual quality of our hand model qualitatively at Figure 1 and quantitatively at Table 1.

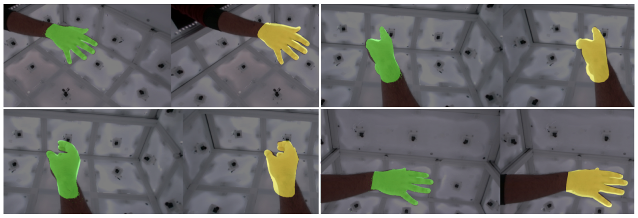

Figure 3. We display a comparison of the pixel misalignment between projected Gaussians and the MANO mesh against a reference image.

Alignment with image pixels: We now demonstrate the pixel-alignment results of MANUS-hand and MANO in Figure 3. Due to inherent design and photo-metric losses, our hand representation is pixel-aligned to reference image, resulting in reduced alignment as compared to that of MANO.

Benchmarking MANUS Grasp scenes: We also evaluate our MANUS Hand and Object method in Table 2 using the data included in the MANUS Grasp dataset. The well-lit scenes and the absence of harsh shadows in our dataset lead to improved evaluation metrics when compared with those of the InterHand2.6M dataset.

1.2. MANUS Grasp Capture

Affect of the number of Gaussians in contact map rendering: We show in Figure 2(b) that the quality of accumulated 2D contact maps deteriorates when the number of Gaussians is reduced. Therefore, in our experiments, we make sure to densely initialize Gaussians for both objects and hands.

Method	PSNR ↑	SSIM ↑	LPIPS ↓	Test time (s) ↓
w/o grid	26.108	0.987	0.0729	**0.0082**
w/o lpips	25.92	0.986	0.074	0.043
Ours	**26.328**	**0.9872**	**0.0688**	0.043

Table 1. Ablation on weight initialization approach and choice of LPIPS loss. Our design approach improve all visual quality metrics.

2. Implementation Details

Our method was implemented in Python using the PyTorch Lightning [4] framework. All experiments were conducted using a single Nvidia RTX3090 GPU with gradient accumulation for 4 iterations. The weights of the different loss function terms - α, β, γ and δ - were experimentally determined and set at values of 0.7, 0.1, 0.1, and 0.1, respectively.

Categories	PSNR ↑	SSIM ↑	LPIPS ↓
Mugs	43.08	0.999	0.002
Bottles	38.17	0.997	0.008
Fruits	39.57	0.998	0.005
Utensils	38.25	0.994	0.009
Misc	38.79	0.995	0.008
Colored	42.38	0.999	0.004
Bags	38.44	0.994	0.011
Jars	40.66	0.999	0.005
Books	36.17	0.998	0.015
Tech	38.81	0.995	0.007
Hand1	28.34	0.995	0.031
Hand2	29.94	0.998	0.029
Hand3	29.71	0.997	0.027

Table 2. Here we benchmark MANUS-hand and object method on MANUS Grasp scenes.

In all our experiments, we chose a grid size of 256x160x142 around the canonical hand skeleton for storing the skinning weights initialized from MANO [12]. MANUS-Hand is initialized with 30K Gaussians per bone, amounting to 900K Gaussians in total. After training, this number is pruned and filtered down to approximately 300K.

3. MANUS Dataset Details

Bone length estimation: We first use the [5] to acquire 2D keypoints for every frame and view. These keypoints are then triangulated into 3D keypoints using the [1]. With these triangulated keypoints, we determine the bone lengths for each subject. Specifically, we average the 3D keypoints across all grasp sequences and then adjust the length of the skeleton accordingly.

Inverse Kinematics: To obtain the joint angles of the hand and its global orientation we use an optimization-based approach inspired by [13]. Specifically, we treat the joint angles, global rotation and global translation as optimization parameters Θ. We then perform a forward kinematics ($Fk(\Theta)$) pass which takes the joint angles as input and outputs 3D joint locations. As the forward pass is differentiable, we apply gradient descent to obtain the optimal parameters that explain the given 3D joint positions. We minimize the L2 loss between predicted and target keypoints:

$$\mathcal{L}_{kyp} = ||Fk(\Theta) - x||^2 \tag{1}$$

where x are the 3D joint locations predicted by AlphaPose [5]. We also impose anatomical constraints (See Figure 6) and joint angle limits by applying a hinge loss as limit loss $\mathcal{L}_{lim}$ as follows:

$$\mathcal{L}_{lim} = \sum_{i=1}^{|\Theta|}((\max(0, ||\Theta^i - l_h^i||^2) + \max(0, ||l_l^i - \Theta^i||^2)) \tag{2}$$

where l_l and l_h are the lower and upper limits on joint angles, respectively. The final loss function is given by:

$$\mathcal{L} = \mathcal{L}_{kyp} + \lambda\mathcal{L}_{lim} \tag{3}$$

We use Adam [8] as our choice of optimizer with a learning of 0.001 and set the value of λ to be 1. We also initialize the current frame based upon previous frame, this helps in faster convergence and helps in maintaining temporal consistency. Once we get the joint angles, we apply one euro filter [2] to the joint angles to smoothen any high-frequency jitter in the sequence. We show illustration of this process in Figure 5.

Segmentation: For every segmentation task, we employ a combined approach utilizing InstantNGP [10] and SAM [9]. Initially, the scene is segmented using the text-based SAM technique. Following this, we obtain a segmentation mask that maintains consistency across multiple views using InstantNGP. If the segmentation masks are found to be inadequate due to inaccurate predictions from the text-based SAM, the process is repeated until satisfactory results are achieved.

Ground Truth Contacts: In Figure 4, we illustrate the methodology used to gather ground truth contact data for our evaluation sequences. Initially, the object is coated with a layer of bright, wet paint. Following this, the object is grasped, resulting in the transfer of paint residue to the hand. After the grasp is finalized, we document the pattern of contact residue left on the hand. To obtain the required viewpoints, we train [10] in the multi-view images and then select 10 distinct views for evaluation. We repeat this process for 15 different evaluation sequences for each subject.

Grip Aperture: The grip aperture [3] refers to the distance between the thumb and fingers when grasping or holding an object. It's an important concept in fields like ergonomics, rehabilitation, and robotics. Here in Figure 7, we plot the change of grip aperture with change in timestep for our dataset.

4. MANO and HARP evaluation

Pose and Shape Estimation: We begin by estimating the shape and scale parameters of the MANO model for each subject. First, we obtain the mesh for every time-step by training [10] on multi-view images. Next, we refine the mesh through the use of MeshLab and Blender software to achieve a cleaned version. We employ an optimization framework akin to that used in [6], focusing on optimizing all MANO parameters, including angle, translation, shape, and scale for the first timestep. This optimization incorporates both keypoint loss (1) and point-to-surface loss [11]

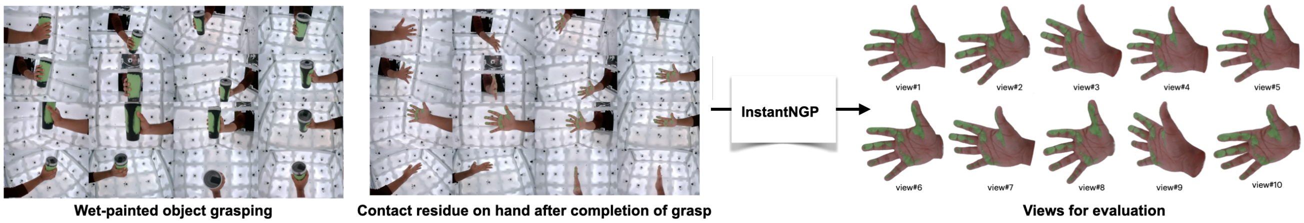

Figure 4. Here, we show the approach we used to obtain the ground truth contacts for the evaluation sequences. On the far right, we display all 10 views of one evaluation sequence for the quantitative assessment of grasp capture.

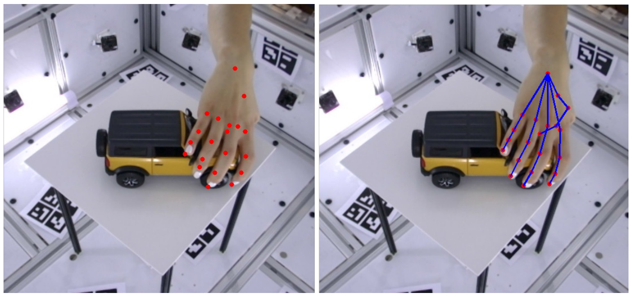

Figure 5. The left figure shows the backprojected 3D keypoints predicted by AlphaPose [5]. The right figure shows the fitted hand skeleton using inverse kinematics.

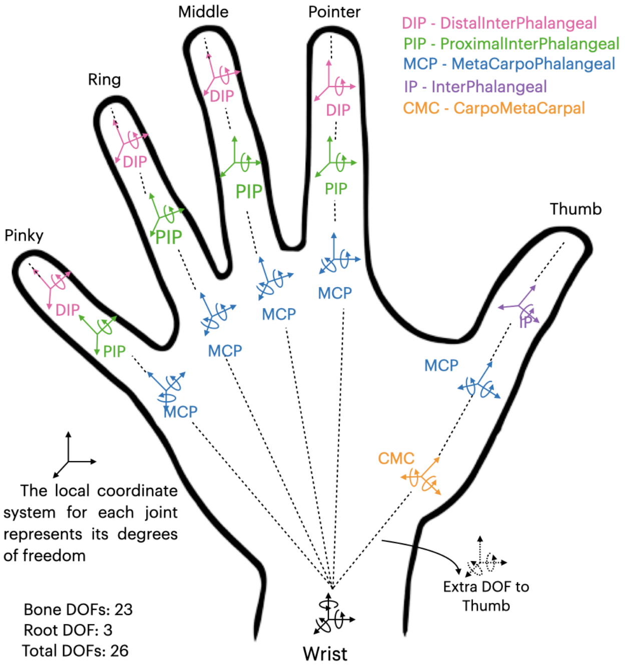
Middle
Pointer
Ring
Thumb
Pinky
DIP - DistalInterPhalangeal
PIP - ProximalInterPhalangeal
MCP - MetaCarpoPhalangeal
IP - InterPhalangeal
CMC - CarpoMetaCarpal
DIP
PIP
MCP
IP
CMC
The local coordinate system for each joint represents its degrees of freedom
Bone DOFs: 23
Root DOF: 3
Total DOFs: 26
Extra DOF to Thumb
Wrist

Figure 6. Figure showing the degrees of freedom of rotation for each of the joint.

with the clean mesh. For subsequent sequences , we keep the shape and scale parameters unchanged, focusing solely

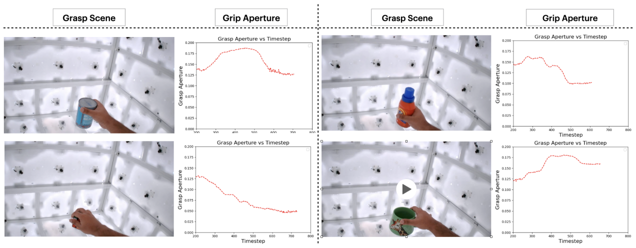

Figure 7. Variation of grip aperture with change in timestep while grasping.

on optimizing angles and translations through keypoint loss. To enhance the speed of convergence, we use the optimized parameters from the previous step as the starting point for new parameters.

To get better geometry than MANO we extend HARP [7] from monocular video setup to multi-view video setup. We start with already optimized MANO model (as mentioned above) and then optimize for the local displacement of the hand shape. We leverage the differentiable rasterizer, to optimize the HARP model based on the losses mentioned in [7].

Evaluation Setup: Please note that, we can't directly render contact maps for MANO and HARP in the same way as MANUS, which employs a Gaussian-based differentiable rasterizer. To obtain contact maps for MANO and HARP, we initially allocate contact values to each vertex, followed by utilizing Blender's emission renderer to render the contact mask. For fair comparison, we increase the resolution of MANO and HARP vertices from 778 to 49,000.

References

[1] Easymocap - make human motion capture easier. Github, 2021. 2

[2] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012. 2

[3] Umberto Castiello. The neuroscience of grasping. *Nature Reviews Neuroscience*, 6:818–818, 2005. 2

[4] William Falcon and The PyTorch Lightning team. PyTorch Lightning, 2019. 1

[5] Hao-Shu Fang, Jiefeng Li, Hongyang Tang, Chao Xu, Haoyi Zhu, Yuliang Xiu, Yong-Lu Li, and Cewu Lu. Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3

[6] Yana Hasson, Gül Varol, Dimitris Tzionas, Igor Kalevatykh, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 2

[7] Korrawe Karunratanakul, Sergey Prokudin, Otmar Hilliges, and Siyu Tang. Harp: Personalized hand reconstruction from a monocular rgb video. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12802–12813, 2022. 3

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 2

[9] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, 2022. 2

[11] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 2

[12] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 2

[13] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. Interactive markerless articulated hand motion tracking using rgb and depth data. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. 2