

Supplementary Materials: 3D Gaussian Editing With A Single Image

Anonymous Author(s)

CCS CONCEPTS

• Computing methodologies → Point-based models; Rendering.

1 IMPLEMENTATION DETAILS

All experiments are performed on a PC with an NVIDIA RTX 3090 GPU with 24GB memory. We leverage Adam optimizer [8] using default parameters of $\alpha = 0.02$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and use the same learning rate as the original 3DGS [7] for the Gaussian parameters, with a cosine scheduler to interpolate the learning rate of means. We set the same learning rate and scheduler for the anchor point as the means of 3D Gaussians and a constant learning rate of 0.001 for all learnable masks.

1.1 Anchor Initialization

The number of anchor points has a great impact on the optimization process. Too few anchor points limit the freedom of deformation, thus hindering the alignment with the reference image. Too many anchor points restrain the gradient from effective propagation through the occluded object parts, thus slowing down the convergence. Therefore, we choose to select a different number of anchor points based on the geometric complexity of the scene. For all characters in the 3DBiCar [11], we sample 800 anchor points. For the NS dataset [12] and Mip-NeRF 360 dataset[2], we sample different numbers according to editing operations, such as 3000 anchor points for Lego deformation and 800 anchor points for Chair stretching. To explore the influence of the number of anchor points, we compare the optimization results of the coarse stage under different numbers of anchor points in Fig. 1. Noticeably, as the number of anchor points increases, the rendered image aligns better with the reference image, while too many anchors lead to structural instability.

1.2 Linear Blend Skinning

We employ an anchor-based hierarchical structure to model long-range object motions. To be more specific, we derive the deformation field of Gaussians using linear blend skinning (LBS) [14] by locally interpolating the transformations of their neighboring anchor points, expressed as

$$\bar{\mu}_i = \sum_{j \in \mathcal{N}_i} w_{ij} (R_j^a (\mu_i - a_j) + \bar{a}_j) \quad (1)$$

$$\bar{q}_i = \left(\sum_{j \in \mathcal{N}_i} w_{ij} r_j^a \right) \otimes q_i \quad (2)$$

Here, a_j is the initial position of anchor point j , \bar{a}_j denotes the current position, and \otimes is the production of quaternions. For each Gaussian i , we use KNN search to obtain its K nearest anchor points, denoted by \mathcal{N}_i . We compute the interpolation weight w_{ij} between a Gaussian G_i and an anchor point A_j with RBF [3, 13]. Consequently,

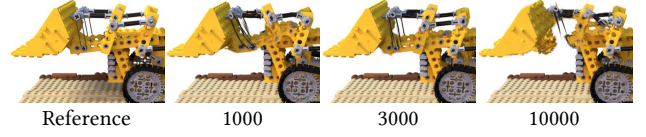


Figure 1: Non-rigid deformation results of the coarse stage under different numbers of anchor points.

we can compute the derivatives with respect to the positions and quaternions of anchor points.

1.3 Rigidity Mask

We regularize the overall structural stability with adaptive rigidity constraints. All the regularization terms are defined using the Radial Basis Function(RBF) [3, 13], where the hyper-parameter γ for RBF is set to be 5. We periodically reset the weight mask by taking the maximum value between the weight and a hyper-parameter η

$$m_{ij} = \sigma^{-1}(\max(\sigma(m_{ij}), \eta)) \quad (3)$$

where η is set to be 0.99. We initialize all the rigid masks to be $\sigma^{-1}(0.99)$ and reset them every 3000 iterations.

1.4 Loss Function

In our experiments, we set $\lambda_1 = 3.2$, $\lambda_{SSIM} = 0.8$, $\lambda_{ARAP} = 600$, $\lambda_{Rotation} = 600$, $\lambda_{Distance} = 30$ and $\lambda_{Mask} = 0.005$ in the coarse stage and adopt $\lambda_1 = 0.8$, $\lambda_{SSIM} = 0.2$, $\lambda_{ARAP} = 300$, $\lambda_{Rotation} = 30$, and $\lambda_{Distance} = 30$ in the fine stage. Additionally, we add $\lambda_{scale} = 300$ and $\lambda_{color} = 1.0$ in the fine stage for geometry editing and texture editing, respectively.

2 EXPERIMENT DETAILS

2.1 Data Preparation

For the NeRF Synthetic dataset [12] and Mip-NeRF 360 dataset [1], we use the original train split to fit the 3D Gaussian model. We select an image from the test split to edit through the 2D imaging tool PhotoShop, including changing the shape of the garden table, bending the microphone and drum stand, stretching the material ball and hot dog, etc. For the 3DBiCar dataset [11], we use the same perspective as in NS to render 50 images of the T-pose mesh. We render 8 images of the posed mesh, including one front view, one back view, two side views, and four surround views diagonally above, as shown in Fig. 2. For the Panoptic Studio dataset [6], we initialize the scene from the pre-trained model from [10] and then select one video to track the subsequent frames.

2.2 Baseline

We use vanilla 3DGS [7], DROT [16], and Deforming-NeRF [17] as baselines. DROT utilizes mesh as 3D representation and optimizes the mesh vertices. For the NS dataset, we obtain the mesh and the corresponding UV map through NeRF2Mesh[15]. For the

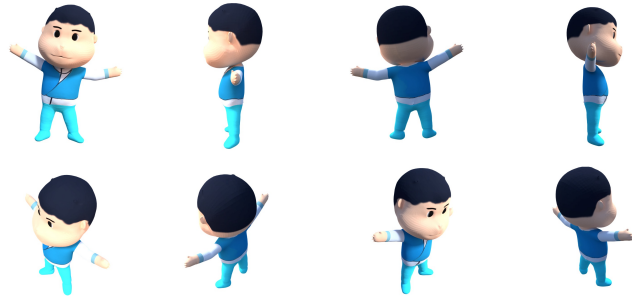


Figure 2: Test views of the posed meshes in the 3DBiCar dataset

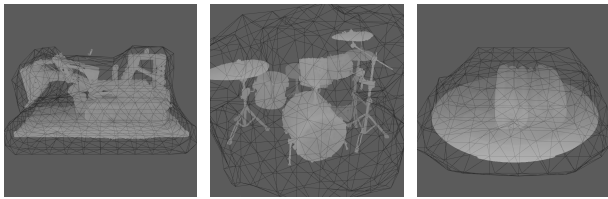


Figure 3: Illustration of the extracted cages of Deforming-NeRF on the NS dataset.

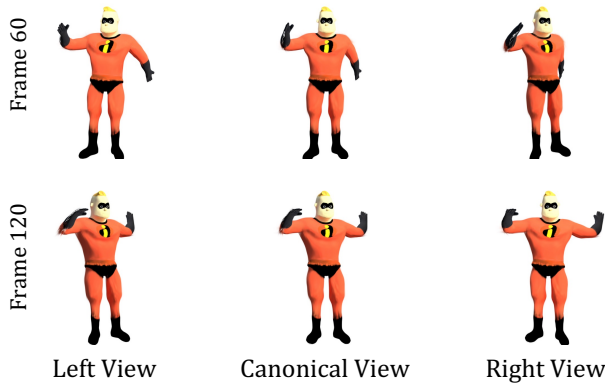


Figure 4: The rendered results of the 3D scene from different frames and different views.

3DBiCar dataset, we directly use the ground truth of the T-pose mesh. Deforming-NeRF fits a NeRF(Plenoxels [4]) with multi-view images, extracts the mesh through the Marching Cubes[9] algorithm, and simplifies it into a cage. Then the user can deform the NeRF by manually adjusting the deformable cage through editing software like Blender. We illustrate the cages extracted from Deforming-NeRF in Fig. 3.

3 APPLICATION

We can use image-to-video models, AnimateAnyone [5] to generate a video of cartoon characters from a fixed perspective. We then use the video to drive the 3D model frame by frame, thereby generating a dynamic 3D scene. We show results from different frames and different perspectives in Fig. 4.

REFERENCES

- [1] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. 2021. Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields. In *2021 IEEE/CVF International*

- Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*. IEEE, 5835–5844. <https://doi.org/10.1109/ICCV48922.2021.00580>
- [2] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 5460–5469. <https://doi.org/10.1109/CVPR52688.2022.00539>
- [3] Mingsong Dou, Sameh Khamis, Yuri Degtyarev, Philip L. Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts-Escobedo, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. 2016. Fusion4D: real-time performance capture of challenging scenes. *ACM Trans. Graph.* 35, 4 (2016), 114:1–114:13. <https://doi.org/10.1145/2897824.2925969>
- [4] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. IEEE, 5491–5500. <https://doi.org/10.1109/CVPR52688.2022.00542>
- [5] Li Hu, Xin Gao, Peng Zhang, Ke Sun, Bang Zhang, and Liefeng Bo. 2023. Animate Anyone: Consistent and Controllable Image-to-Video Synthesis for Character Animation. *CoRR abs/2311.17117* (2023). <https://doi.org/10.48550/ARXIV.2311.17117>
- [6] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart C. Nabbe, Iain A. Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. 2015. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 3334–3342. <https://doi.org/10.1109/ICCV.2015.381>
- [7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.* 42, 4 (2023), 139:1–139:14. <https://doi.org/10.1145/3592433>
- [8] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). <http://arxiv.org/abs/1412.6980>
- [9] William E. Lorensen and Harvey E. Cline. 1987. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987, Anaheim, California, USA, July 27-31, 1987*, Maureen C. Stone (Ed.). ACM, 163–169. <https://doi.org/10.1145/37401.37422>
- [10] Jonathan Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. 2023. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713* (2023).
- [11] Zhongjin Luo, Shengcai Cai, Jinguo Dong, Ruibo Ming, Liangdong Qiu, Xiaohang Zhan, and Xiaoguang Han. 2023. RaBit: Parametric Modeling of 3D Biped Cartoon Characters with a Topological-Consistent Dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 12825–12835. <https://doi.org/10.1109/CVPR52729.2023.01233>
- [12] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 12346)*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer, 405–421. https://doi.org/10.1007/978-3-030-58452-8_24
- [13] Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 343–352. <https://doi.org/10.1109/CVPR.2015.7298631>
- [14] Robert W. Sumner, Johannes Schmid, and Mark Pauly. 2007. Embedded deformation for shape manipulation. *ACM Trans. Graph.* 26, 3 (2007), 80. <https://doi.org/10.1145/1276377.1276478>
- [15] Jiaxiang Tang, Hang Zhou, Xiaokang Chen, Tianshu Hu, Errui Ding, Jingdong Wang, and Gang Zeng. 2023. Delicate Textured Mesh Recovery from NeRF via Adaptive Surface Refinement. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*. IEEE, 17693–17703. <https://doi.org/10.1109/ICCV51070.2023.01626>
- [16] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. 2022. Differentiable Rendering Using RGBX Derivatives and Optimal Transport. *ACM Trans. Graph.* 41, 6 (2022), 189:1–189:13. <https://doi.org/10.1145/3550454.3555479>
- [17] Tianhan Xu and Tatsuya Harada. 2022. Deforming Radiance Fields with Cages. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXXIII (Lecture Notes in Computer Science, Vol. 13693)*, Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer, 159–175. https://doi.org/10.1007/978-3-031-19827-4_10